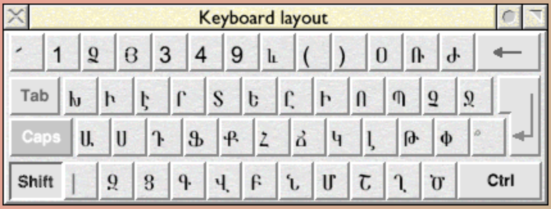


!KeyMap

A front-end to Richard Spencers
Desktop Keymap Module

Comments, questions, complaints?
paul@sprie.nl

Check for new versions at:
www.riscos.sprie.nl

This PDF file is created with !PrintPDF, by Stephen Fryatt, from an original OvationPro document, on an Iyonix PC equipped with RISC OS 5·16

General

!KeyMap is a front-end to Richard Spencers clever *Desktop Keymap Module*, which in fact is a programmable keyboard driver.

The module is able to assign any character, either Latin or UTF-8 encoded, to any key of the keyboard, by means of a keymap table that can be created easily in !Edit, or any other text editor.

Unicode characters, however, can only be retrieved from unicode fonts. This implies that only RISC OS 5 users, or users that have the RISC OS 5 FontManager and ROMFonts modules installed, can fully benefit from !KeyMap.

Others may still use KeyMap for creating keyboard drivers for special Latin fonts, such as Greek, East or Cyrillic, or for simply modifying the keyboard according to their own wishes.

A special feature of !KeyMap is the *Example keyboard*, that displays the defined characters of every keymap in the intended keyboard lay-out.

It can even act as a virtual keyboard, by clicking on the appropriate keys.

Warning

Although KeyMap is able to output UTF-8 encoded text, it doesn’t pretend to overcome the limitations on unicode usage on RISC OS 5.

Currently, RISC OS 5 doesn’t provide a way to assign encoding information to a character, and although it can be configured to use UTF-8 as the system encoding, this will unfortunately break up most of the applications. To put it simply, RISC OS is not designed with encoding awareness in mind.

Therefore, a program can only accept unicode character input, if it meets both of the following two conditions:

- It can be configured to use a font with the *EUTF8* extension.
- It can be set up to *expect* unicode input, by choosing that font.

Although neither of these two conditions is particularly demanding, so far the only program that benefits from the unicode approach of KeyMap is *!Dict*, also available from www.riscos.sprie.nl.

Why this front-end?

Richard Spencers Desktop Keyboard Module is able to read so-called *keymap tables* that can be created in any text editor.

Activating (or de-activating) such keymaps can be done by means of special command line calls.

The front-end !KeyMap essentially keeps track of every keymap table that has been created. Switching between individual keymaps, or between a keymap and the default system keyboard driver is a matter of simple mouse clicks.

KeyMap also provides a virtual keyboard for examining the lay-out of every keymap table.

The iconbar icon, as well as the iconbar text, gives information about which driver is currently active.

RISC OS 5 users

KeyMap is able to create unicode keyboard drivers. Therefore, you will need one or more unicode fonts. The Homerton and Trinity fonts that come with RISC OS 5 can be considered as unicode fonts indeed, but they only support a limited set of characters. Fortunately, more extensive fonts, such as Cyberbit, can easily be downloaded from the internet. However, before they can be used, they'll have to be converted to the RISC OS font format first. You can use John-Mark Bell's *!TTF2f* to achieve that. It is available from *jmb.drobe.co.uk*.

Non-RISC OS 5 users

Non-RISC OS 5 users may try to download both the FontManager and the ROMFonts modules from the ROOL site *(www.riscosopen.org)* and place them in *Boot.Choices.Predesk*.

But even if unicode is not applicable, KeyMap can still be used for creating keyboard drivers for special Latin fonts, such as the Cyrillic fonts of the Electronic Font Foundry *(EFF_Times.Cyrillic)*, or just for transposing keys, for example in case of a not entirely compatible Windows keyboard.

Running !KeyMap

- Save the !KeyMap application directly from the archive to your hard disk.

- Double-click on the !KeyMap icon in the filer window.
  The same icon will pop up on the iconbar.

- Choose a keymap from Keymaps in the iconbar menu (17 keymaps are already provided).

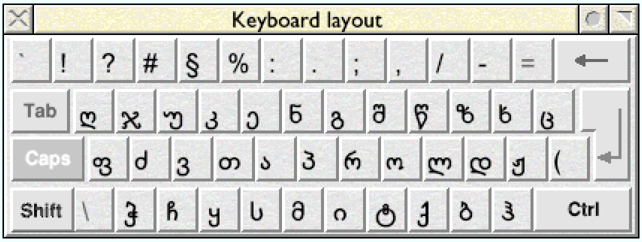
- Choose *Keyboard* from the iconbar menu (or click with ADJUST on the iconbar icon), to pop up the *example keyboard* that will show you the lay-out of the chosen keymap.

- Start typing in whatever application that accepts text. Bear in mind, however, that currently the only program that accepts unicode input, is !Dict.

- You can always switch to the system keyboard by clicking on the icon-bar icon.
  If you want to go back to the previous keyboard, click on the iconbar icon again.

- If you need to switch regularly between two different keymaps, you can define these in the Configure window, accessible from the iconbar menu. Toggling between both keymaps is a matter of clicking ADJUST on the iconbar icon. This supersedes the default behaviour of Adjust-clicking (i.e. short-cut to pop up the example keyboard) when no Adjust toggle is defined in the Configure window.

Included keymaps



Currently, 17 keymaps are included, 12 of which are related to unicode encoded fonts, that can only be used in conjunction with both the RISC OS 5 FontManager and ROMfonts modules.

Most of the keymaps are modelled after the keyboards that are used in Google Translate. By no means they intend to be comprehensive.

- There are 6 Russian keyboard drivers.
  - *Russ Uni* is the driver that should be used in conjunction with a unicode font, mimicking a Russian typewriter lay-out, while
  - *Russ Uni-f* is modelled after a more phonetic approach.
  - *Russ Lat-f* has the same lay-out as *Russ Uni-f,* but rather than producing unicode Cyrillic characters, it drives the Russian Latin1 fonts, such as provided by the Electronic Font Foundry (typically *EFF_Times.Cyrillic*).
  - *Russ Lat-acc* gives access to accented vowels in the same Cyrillic Latin1 fonts.
  - *Russ 1251* is the driver that should be used in conjunction with a Windows 1251 encoded font, such as the one that is included in this archive.
  - *Russ 1251-f* drives the same 1251 encoded font, but its keyboard layout is similar to the Russ Uni-f and Russ Lat-f drivers.

- The *UTF/Latin* keymap mimics the entire ASCII set of the common RISC OS Latin fonts. This means that top bit characters, such as accents, that are accessible in Latin1 fonts by familiar key presses, will now also be available in UTF8 encoded fonts.

- For example, *á* is achieved by pressing *Alt-[* followed by *a*.
- The Dutch *ij* is assigned to the `-key, as well as to the *Ctrl-y* combination.

➻ The *VRPC* keymap attempts to address the deviating keys that are used by VirtualRPC on the parent Windows machine. Since individual keyboards tend to differ, this keymap is probably subject to personal modifications.

Creating keymaps

General

A keymap table is a nothing more than a text file, that should be saved within the !KeyMap program directory.

Double click on the !KeyMap icon in the filer window, whilst holding down Shift. Search for the directory Keymaps and open it. Here you will find the keymaps that are already defined.

Save your own keymaps to this directory.

You will need to restart !KeyMap in order to let the program be aware of new keymaps.

Syntax

The syntax of a keymap table is simple.

- First, type the key to which you want to assign a character.
- Then type a space, followed by the intended character.

For example, if you wish to replace the Q and the W by A and Z (for turning your standard QWERTY-keyboard into an AZERTY one), you should type the following:

```
Q A
W Z
q a
w z
```

ASCII

Rather than typing the key, you can also type its ASCII value.

For example:

```
65 81
97 113
```

is similar to

```
A Q
a q
```

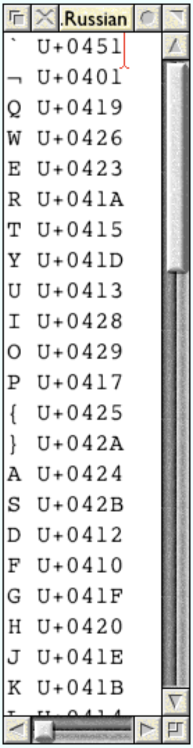
For some keys the ASCII value is obligatory, typically the number keys 0-9 (that otherwise would be interpreted as the wrong ascii values!), as well as the #-key, which is used as the comment key (see below).

For example, if you want to want to transpose the #-key and the ?-key, you should type:

```
? 35
35 ?
```

Unicode

If you want to assign unicode characters to a key, you will need to type the so-called *Unicode code points* – in the form of: `U+` followed by a *hex* number, as shown in the illustration. These code points (there are tenths of thousands of them!) can be found on the internet. Here's a rather comprehensive table: *www.utf8-chartable.de*



```
`  U+0451
¬  U+0401
Q  U+0419
W  U+0426
E  U+0423
R  U+041A
T  U+0415
Y  U+041D
U  U+0413
I  U+0428
O  U+0429
P  U+0417
{  U+0425
}  U+042A
A  U+0424
S  U+042B
D  U+0412
F  U+0410
G  U+041F
H  U+0420
J  U+041E
K  U+041B
```

Control keys

Apart from the upper and lower case keys, also the so-called Control keys can be defined by typing their ASCII values. They are in the range from 0 (`Ctrl-@`) to 31 (`Ctrl-_`).

However, these keys should be used with care, since many programs as well as the OS itself use quite a few of these combinations for special operations (think of `Ctrl-C`, for copying a selection).

The *Russ Lat-f* keymap table has the accented vowels defined as Control keys. However, since a clash between these mappings and default operations in, say, OvationPro are likely to occur, a separate keymap table, which assigns these accented vowels to normal key presses, is provided as well.

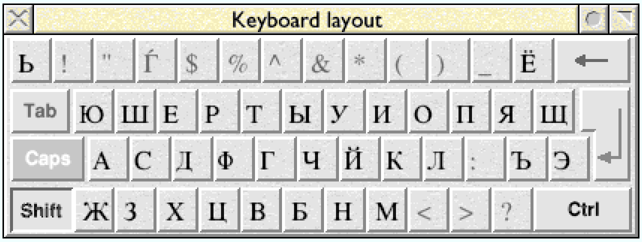
Comments

If a line in the keymap table starts with a #, a comment is assumed. Such lines will be ignored by the module. However, if the first line of a keymap table starts with `#Font:` then !KeyMap will use the font that is typed after it for rendering the example keyboard. The keymaps *Russ Lat-acc* and *Russ Lat-f* make use of this little feature.

This can even be used for turning !KeyMap into a simple font viewer. For example, if you want to see which *Dingbats* characters can be typed directly from the keyboard, you can create a keymap file, called *Dingbats*, that contains just one line:

```
#Font:Dingbats
```

Example keyboard

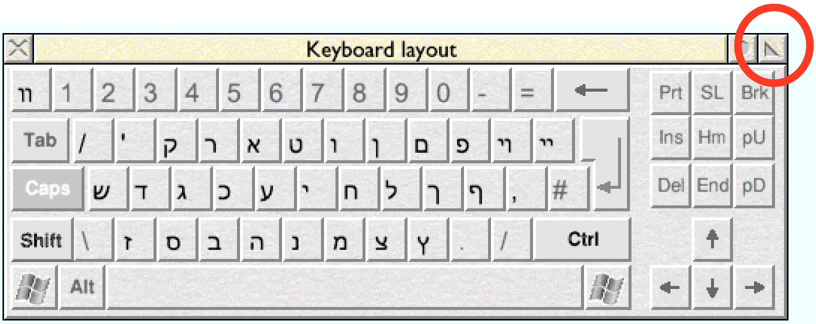

Keyboard layout
Ь ! " Ґ $ % ^ & * ( ) _ Ё ←
Tab Ю Ш Е Р Т Ы У И О П Я Щ
Caps А С Д Ф Г Ч Й К Л : Ъ Э
Shift Ж З Х Ц В Б Н М < > ? Ctrl

A valuable feature of !KeyMap is the Example Keyboard, which can be accessed from the iconbar menu, or by clicking ADJUST on the iconbar icon (as long as the *Toggle keymaps* in the Configure window are not defined).

The Keyboard is not only valuable as a reminder of where you've put every character, but also for checking newly created keymap tables for the inevitable typos.

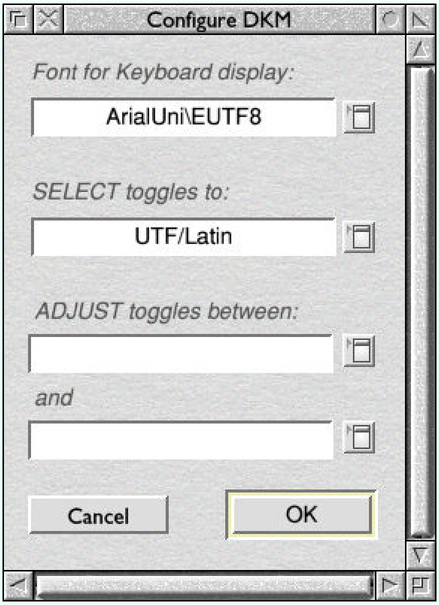
Moreover, the Keyboard can also act as a virtual keyboard, just by clicking on the appropriate keys. For capital characters, click *Shift* first, or click with *Adjust*. Clicking on *Ctrl* will reveal the characters that are assigned to Control keys (if there are any).

If you want a more comprehensive virtual keyboard, complete with cursor keys, click on the *resize icon* at the top right side of the window.


Keyboard layout
װ 1 2 3 4 5 6 7 8 9 0 - = ←
Tab / ' ק ר א ט ו ן ם פ ױ ײ
Caps ש ד ג כ ע י ח ל ך ף , #
Shift \ ז ס ב ה נ מ צ ץ . / Ctrl
Alt
Prt SL Brk
Ins Hm pU
Del End pD

Customising !Keymap

Choosing *Configure* from the iconbar menu will reveal the following options:



Font for Keyboard display: ArialUni\EUTF8

SELECT toggles to: UTF/Latin

ADJUST toggles between:

and

Font for Keyboard display:

This is the font that will be used for rendering the defined keys in the example keyboard. RISC OS 5 users will likely choose a unicode font here. The more extensive such font is, the better. *Cyberbit* and *ArialUni* (from Windows) are very large and therefore very suitable fonts. Other users should choose a non-unicode font. If no font is chosen, *Homerton.Medium* will be used, either with or without the *\EUTF8* extension, depending on the unicode capabilities of the system.

SELECT toggles to:

By default, clicking with SELECT on the iconbar icon toggles between the last chosen keymap table and the system keyboard driver (usually Latin1).

However, this is made configurable for those who wish to toggle between the last chosen keymap table and another keymap table. If you want to toggle to the system keyboard driver again, you should enter <*none*> from the accompanying menu.

ADJUST toggles between:

If you need to toggle between two special keymaps on a regular base, you can specify both maps in the last two icons of the Configure window, by choosing them from the accompanying menu.

ADJUST will then toggle between those two – contrary to the initial behaviour of popping up the example keyboard. If you want to revert to that, enter <*none*> in both icons.

Press OK in order to save and activate the changes.

Notes

Limitations

- Currently, a keymap table is limited to 224 entries, comment lines included.

- Key *combinations*, such as used for entering accented characters, are not yet implemented.

- The ultimate limitation is the practical use of !KeyMap when it comes to unicode output. Virtually none of the RISC OS applications, !NetSurf included, is able to expect UTF-8 encoded text as an input and process it properly, while the near future doesn’t promise radical changes in this respect. The only program that benefits from the !KeyMap unicode feature will be !Dict for the time being.

Having said that, using !KeyMap for entering unicode Russian into !Dict, reveals a wealth of Russian dictionaries and texts on *dict.dvo.ru*, that would otherwise remain hidden permanently.

License

Both Desktop KeyMap Module and !KeyMap are freeware. Use at own risk (no real danger, though). May be freely copied without modifications and with these copyright notices intact.

Thanks to

- Richard Spencer, for developing the priceless Desktop KeyMap Module.
- John-Mark Bell for providing the True Type to RISC OS font convertor and for refuting my fantasies about a universal solution concerning unicode usage on RISC OS 5.
- Jon Robinson, for composing the Russian 1251 drivers and for providing the appropriate converted Windows 1251 fonts.

Version history

27-04-2011

First alpha version.

04-05-2011

The keyboard sprites are kept in separate directories.
They will only be loaded when necessary, rather than keep them in an ever growing sprite pool. This will also make copying new versions over the old one easier, although the key sprites are still kept in the !Sprites area.

06-05-2011

The idea of keyboard lay-outs as a sprite is abandoned. Instead, the lay-out is now painted on the fly, while scanning the keymap in question.
Clicking with Adjust on the iconbar icon toggles between a keyboard lay-out or not.

08-05-2011

- Various bug fixes.
- In a keymap file, a special font can be specified. The first line should be something as: #Font:EFF_Cyrillic.medium Keymap will then use this font for rendering the example keyboard.
- The default unicode font is now configurable. Choose Config from the iconbar menu.

13-05-2011

- Clicking on an icon (key) in the example Keyboard passes the corresponding character to the keyboard buffer.
- Clicking Adjust on the iconbar icon toggles between two keymaps, as defined in the Configure window, otherwise it will toggle between an example keyboard or not.

20-05-2011

- Clicking Select has been made configurable. It used to toggle between the last chosen keymap and the System set, but now it will toggle between the last chosen keymap and a predefined keymap (see choices).
- Keymap UTF/Latin added, which mimics a Latin1 encoded font. This means that top bit characters that are accessible in Latin1 by key presses, such as accents, will now also be available in UTF8 encoded fonts. (Very useful for !Dict, for example.)

30-05-2011

- When two keys are transposed, clicking on the corresponding key passed the original character, rather than the transposed one.

This is now fixed, although some other unexpected key passes still remain.

- Shift and Ctrl-keys show their status in the example keyboard.

15–06–2001

- The default font can now be chosen from a font menu, rather than entering it manually, at last.

30-06-2011

- Rather than checking for the Os-version, !KeyMap now reads the Fontmanger version and the ROMFonts version in order to decide whether unicode is applicable.
- The !Run file also checks for the presence of ABClib.

02-07-2011

- Reading the Fontmanager version appears to be not fool proof, due to overlapping version numbers. !KeyMap now uses a routine, kindly provided by Martin Würthner, that checks for the correct installation of a possible UCS FontManager.

03-07-2011

- Better check for fonts, whether unicode capable or not.

23-08-2011

- Enter-key added to virtual keyboard.
- Slightly better position of characters on a slightly smaller keyboard.

25-08-2011

- Appearance of the virtual keyboard improved. Click on the resize icon to reveal the other buttons.

18–09–2011

- Trailing spaces in keymap files are now ignored.