

HOW TO MAKE A NEW KEY MAP FOR KEY CURRY

Edward H. Trager

April 20, 2010

Key Curry makes use of a simple key map file format which is almost identical to the format used by the Open Source editor, **Yudit** (<http://yudit.org>).

However, Key Curry's map parser is not as sophisticated as the parser used in Yudit and therefore several simple modifications to key maps originally created for Yudit may be required in order for pre-existing key maps to work in Key Curry.

If you want to modify an existing key map to make it work in Key Curry, this document will point out to you specifics that may require your attention. For new key maps, just follow the instructions presented here and you should be able to quickly create new key maps that work under Key Curry.

1. ASCII and UTF-8 File Format. The key map file should be saved as a UTF-8 file. The actual map data itself uses only the ASCII subset of UTF-8. However, the file should be saved in UTF-8 format so that file comments can be written using the full repertoire of Unicode characters.

2. Comments. Comments in a map file are introduced by two forward slash characters at the beginning of the line, just as is common in C++ and other code:

```
//  
// PanAfricanLatin.kmap  
// © 2009 by Edward H. Trager.  
//  
// Includes extended Latin letters needed for African languages  
// as well as letters and punctuation for French  
//
```

3. Map entries. Map entries consist of *key-value pairs*. The entire key-value pair is presented as a single string surrounded by ASCII double quotation marks. The entry is always followed by a comma - this is true even for the last line of the file:

```
"A;=0x0041 0x0327",
```

4. Keys. Keys consist of one or more ASCII characters available on the physical keyboard. In the example shown here, the key consist of the sequence of two characters, LATIN CAPITAL LETTER A and SEMICOLON:

```
"A;=0x0041 0x0327",
```

5. Separator. The key is separated from the value by the presence of EQUAL SIGN as a separator character:

```
"A;=0x0041 0x0327",
```

6. Values. Values consist of Unicode Hex values separated by a space character. The hex values are introduced with the “0x” notation that is standard for JavaScript (and other computer languages). In this example, two Unicode values – representing the capital letter A and the combining diacritical mark – are shown:

```
"A;=0x0041 0x0327",
```

7. Don't forget the comma! A comma is always required to terminate an entry. This is true even for the last line of the file:

```
"A;=0x0041 0x0327",
```

8. Comments after entries. You should always include a comment after an entry. This makes the file much easier for humans to read:

```
"A;=0x0041 0x0327", // A
```

9. Symbolic Key Values. In order to increase file readability and to reduce problems with file parsing, some ASCII characters **must** be described symbolically within curly braces when used as part of the keys. For example, here the the key sequence CAPITAL LETTER E followed by EQUAL SIGN is used to obtain CAPITAL LETTER E WITH TILDE:

```
"E{equalsign}=0x1EBC", // Ë
```

Because the EQUAL SIGN itself is used a separator character between keys and values, it cannot be directly entered into the file as part of a key. For this reason, the EQUAL SIGN should always be entered using its symbolic description as shown.

Below is the full table of characters that should always be entered using the symbolic descriptions shown within curly braces when used as part of a key:

{equalsign}	"="	Equal sign
{space}	" "	Space key (preferred)
{s}	" "	Space key (not recommended)
{tab}	"\t"	Tab key
{comma}	","	Comma
{singlequote}	"'"	Single quotation mark
{doublequote}	"'"	Double quotation mark

10. Suggested Documentation, Contact Information, and Licensing

Information. Please document the sources used to research or construct your key map. Also provide your own contact information - an email address is sufficient. Be sure to also include licensing information. We believe that a *Creative Commons Attribution-Share Alike License* is a good choice for these key maps. Here are the entire header comments from an example file:

```
//  
// Dzongkha Keyboard Layout  
//  
// Based on the charts and keyboard layout Copyright (c) 2000  
// by the Royal Government of Bhutan, Dzongkha Development Commission,  
// P.O.B. 122, Thimpu, Bhutan. The original document says,  
// "All Rights Reserved. May be freely distributed and implemented."  
//  
// NOTA BENE: In this implementation, we use the "%" key  
// instead of "AltGr" to access the less-commonly used  
// characters on the 3rd and 4th charts.  
//  
// Data compiled into Key Curry / Yudit keymap format
```

```
// Copyright (c) 2009 by Edward H. Trager
// <ed dot trager at gmail dot com>
//
// http://www.unifont.org
//
// Released by Ed Trager under the Creative Commons
// Attribution-Share Alike License which seems to encapsulate
// the distribution and usage intent of the original source
// document.
//
// http://creativecommons.org/licenses/by-sa/3.0/us/
//
// 2009.02.24.ET
//
```

*** End of Document ***