What is MacTranslit? 1
Do I need it? 1
Quick Start 1
What is transliteration? 2
What is included in this package 4
Features of MacTranslit 4
Using MacTranslit 5
Caveats 6
Future plans 6
Shareware Notice 7
Legal Stuff 7
Credits 7
Contact info 7
Index 8

What is MacTranslit?

MacTranslit is System 7-only text utility that allows you to transliterate text written with the use of different character sets.

Do I need it?

If you need to communicate with people in different countries or if you study or process foreign languages, then MacTranslit is probably for you.

Quick Start

For those people who know what's the difference between font, glyph, code page, style and plain-text, just follow these steps:

If you want to transfer a document written by you from one font encoding to another:

1. Save document as plain ASCII text file.

2. Launch MacTranslit.

3. Select type of font used in the original document in ***From*** pop-up menu.

4. Select type of font you want to use in ***To*** pop-up menu.

5. Click the ***Do Transfer*** button.

6. Select input file, ener name for output file and click the ***Save*** button.

7. Open output document with your word processor and set it into new font.

If you want to read document created by someone else:

1. Find out from its author which font was used to create this document.

2. Open the document with your favorite word processor. Word, for instance, should be able to convert almost any type of document into Word format. **You do not need to have font used to create this document installed in your System.**

3. Save the document as a plain text file.

4. Launch MacTranslit.

5. Select the type of font used in the original document in ***From*** pop-up menu.

6. Select the type of font you want to use in ***To*** pop-up menu.

7. Click the ***Do Transfer*** button.

8. Select the input file, enter name for the output file and click on the ***Save*** button.

9. Open the output file with your word processor and set it into the new font.

What is transliteration?

Text is stored in memory or on a disk as a set of numbers. Every number represents its own **character** on the screen. The numbers used are 8-bit long binary codes, which means that they can only represent 256 different characters. That is more than enough for most languages. The English alphabet for instance has only 26 letters in it, so 256 codes is enough to uniquely identify every letter of English language plus all the numbers plus special characters and have plenty more to spare.

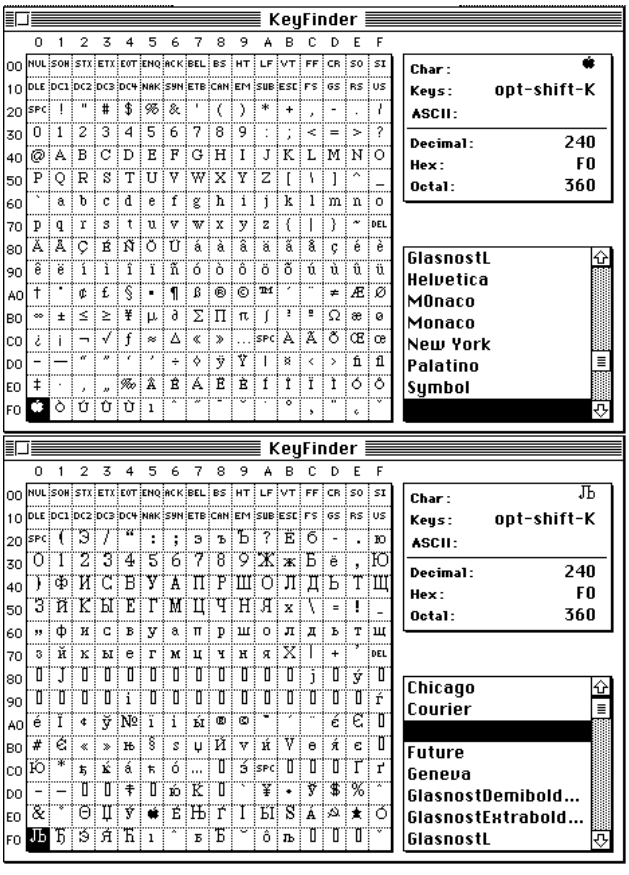
In fact, standard **ASCII character set** also known as **low ASCII** uses only 128 characters that can be described by 7-bit long number (that's why it called **7-bit ASCII** as well). The purpose of **font** is to provide means for the system to associate these **character codes** with their visual representation on the screen or on paper. That visual representation of a character is called a **glyph**.

A **font** is a collection of all the glyphs used to represent the characters of the language. Different fonts of the same language usually have slightly different glyphs but same character codes for them will represent similar looking characters. No matter which font you use, Chicago or Times, bold or italic, size 10 or 24, letter "N" would have same corresponding character code ($4E hex or 78 decimal), although it will have slightly different shapes. For those who use only English, different fonts are distinguished only by the shape and look of the letters.

Unfortunately things get a lot messier when you need to deal with foreign languages. The 8-bit character set is not enough to represent all the languages in the world or even just few of them. The most you can fit in a standard 8-bit font is 2 alphabets, if they are not very big (English and Russian fit nicely for instance). To be able to display text with languages other than English, you would have to use a font that uses same character codes to represent different characters on the screen. The way Macintosh knows what character it needs to draw on the screen, English "**T**" or Cyrillic "**Т**" is determined by information that your word processor has, which is kept in the document that tells the system which font to use.

It is OK if you write your own document on your own system and only give printed copies of it to others. The problem arises when you need to exchange the actual electronic bits, which make up this text file. English character set is well defined and standard in virtually every country; everyone knows what ASCII is and if your document is written in English it usually means it uses an ASCII based font. Even if you don't have the exact font used by someone else to create the document you are trying to read, you can substitute it with another **Roman character set** based font. It will break formatting information, but document will still be readable. Unfortunately it is not so for many other languages.

Cyrillic fonts, or to be more precise, Russian fonts are a very good example. Russia had its own standard character set, somewhat equivalent to the low ASCII, called KOI-7 (actually, there are 3 variants of KOI-7). This set has all the letters of the Russian alphabet, numbers and special symbols. There is another standard character set in Russia, called KOI-8, which is bilingual. Its first 128 characters are the same as in low ASCII and upper half is for Russian letters. There is yet another character set, the KOI-8 Alternative Variant, which is similar to the original KOI-8 but differs in the Russian upper half. There are a few more different encodings but their description is outside the scope of this limited manual. Contact me if you are interested.



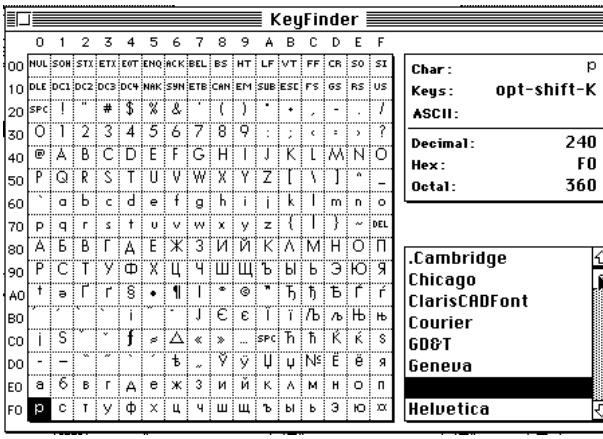


Fig. 1

Things are even worse in the USA. There is an old Cassady & Greene Cyrillic font, Glasnost, which is a unilingual Russian character set, the Apple standard Russian set which is bilingual, the Russian set for Windows and probably many others. All these sets have different codes used for same characters. If you don't have fonts based on these sets you cannot view them. In Fig. 1 I show the mapping of 3 different fonts – Roman Character Set (Times), Russian Apple Standard (Glasnost from Cassady & Greene) and Cassady & Greene old unilingual Russian font. The first 127 codes represent same glyphs for first 2 sets, third set is totally different.

This is where MacTranslit comes to the rescue. MacTranslit can remap different sets, provided coding tables for both of them available. There are a few catches though. Since I don't know how a word processor like Word or Nisus stores its documents, MacTranslit can only work on plain text files. It means that you must save your document as a text file before MacTranslit can do anything useful. It also means that if you had any formatting in your document it will be lost. It also means that if you used more than one character set based font in your document, one of them will be turned into garbage. For example, if you had paper written with unilingual Glasnost font for Russian and any of the Roman fonts for English in the same document and then transliterate it into Apple Standard Russian, the English part will be transferred as if it were Russian. In the future I hope to implement the **Rich Text Format** parser that will allow you to keep some of the formatting and do transliteration only on the fonts that you need to.

What is included in this package

This package consists of a shareware version of MacTranslit v1.2 application and of this manual. The shareware version of MacTranslit includes 3 conversion tables to transliterate among:

Russian Apple Standard Character Set

KOI-8 Alternative Variant (Russian Standard for 8-bit bilingual character set)

Unilingual Russian Character Set by Cassady & Greene (the Glasnost family)

Features of MacTranslit

MacTranslit is a System 7 only utility. It 32-bit clean, supports balloon help and drag and drop. It is extremely fast (500 kb in 10 sec on Mac IIsi or 20 sec on a Mac Plus). It requires 100 kb RAM to run, although it can survive with 60 kb if balloons are not used. It supports AppleEvents, with 3 events – OAPP, ODOC and QUIT. ODOC event will do transliteration the same way as in drag and drop (see **Using MacTranslit**).

MacTranslit can work with almost any other languages, not just Russian. It is just a matter of using a different table.

It is possible for users to create new tables, although it is much easier (and cheaper) to pay me and leave me have all the headaches associated with it. If you want to do that, contact me for information.

Registered user of MacTranslit will receive:

Floppy with MacTranslit v1.2

Printed manual.

My full technical and moral support.

Additional tables, which at the moment include:

KOI-7 Russian Character Set.

WP-866 Russian Character Set for WordPerfect.

Russian Character Set for Windows.

And more conversion tables when I write them.

Also, registered users will be able to order custom-made tables for a nominal fee of $10.00. (Contact me on how to order)

Using MacTranslit

When you start MacTranslit, it will automatically open the Transfer window (Fig. 2)

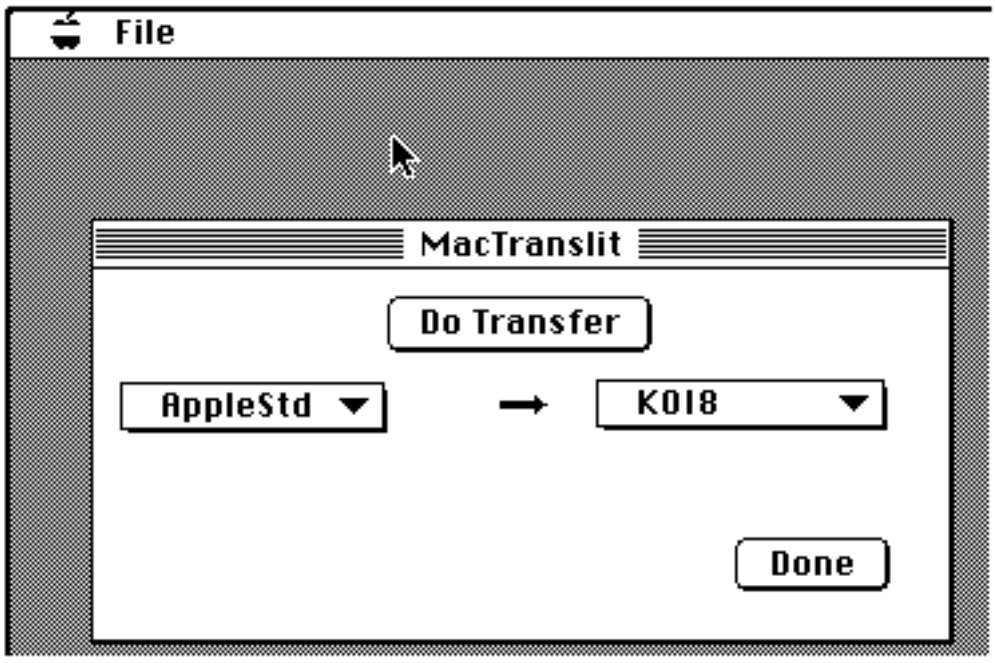


Fig. 2

Two pop-up menus are provided to select input and output character sets (left menu is input, right is output). The **Do Transfer** button will initiate the selected transfer, bringing up a standard GetFile Dialog box to select an input file, and the PutFile dialog box to name the output file. **Done** button will hide the window. **Transfer...** item in **File** menu will bring it back.

Drag and drop operation will be performed according to the transfer you have selected if MacTranslit is running, or according to your last choice if it is not. The default output filenames for drag and drop operation will be same as ithe nput name with **.out** appended. If the file with same name as the output file already exists in same directory, MacTranslit will terminate the transfer with a somewhat cryptic error -48 (duplicate file name error). Output files will be placed in the same directory as input files.

MacTranslit will recognize only **TEXT** files , so if you don't see the file you want to transliterate in GetFile dialog box, or if cannot drag your document onto the MacTranslit icon, make sure it is saved as TEXT in your favorite word processor.

Caveats

As was mentioned in **What is transliteration?** MacTranslit cannot handle multistyled text, so you have to restore all formatting by hands. Also, if more than one character set is used, the secondary sets (the ones you did not want to transliterate) will be transliterated as well. It will most likely turn the other sets into some unreadable garbage. If you are in such situation, separate all parts of the text that were created with the other set and reinsert them in your document after transliteration. One last problem arises from the usage of special characters that exists in the input set but not in the output set. For instance, the Apple symbol might exist and be*used* in ithe nput set, but not in the output set. In cases like that MacTranslit will try to transliterate that character into the closest one in the output set. Unfortunately, it is not always possible to find such a character. In such cases MacTranslit will not convert that particular character at all.

Future plans

For immediate future I plan to :

Provide Unicode support to ease transliteration between all existing character sets.

Provide clipboard support (it means it will be possible to transliterate text in the clipboard)

Provide styled text support through Rich Text Format (RTF) and clipboard.

Additional tables for transliterating between other writing system, such as Ukrainian, Belorussian, Polish, German, French, etc.

For more distant future I plan to :

Add some additional text processing capabilities, such as add/strip LF and CR, maybe grep-like.

Add support for 2-byte text (like Japanese)

AppleScript support.

Shareware Notice

The MacTranslit is shareware. It means that you can evaluate it before you decide to pay for it. If you like it and continue using it you have to pay for it, if you don't like it or don't want to pay for it you have to stop using it. It can be distributed freely provided it is not modified and this manual is distributed together with it. Organizations that charge anything except normal download charges for distribution have to get my express written permission before distributing MacTranslit. It can be distributed on Info-Mac CD. Since I understand that it is practically impossible to force someone to pay for shareware, I'll provide incentive to buy it by sending additional tables to all registered users(see **Features of MacTranslit** for a list.) In addition I will make custom tables for registered users for a nominal fee of $10.00 per table. And yes, registered users will be eligible for free upgrades.

I accept cash, checks and money orders in US$.

Legal Stuff

This software is provided "as is" and you, the user, are assuming full responsibility for its use and the entire risk as to its quality and performance. In no event will I be responsible for any damages, direct or consequential to your data, computer, hard drive, home, life, marriage, divorce, children, death, etc. I also cannot be held responsible for any change in atmospheric conditions, earthquakes, meteorite strikes, fires, wars, etc.

The software is believed to be bug-free. It has been tested on Mac SE, IIsi, LC, Centris 610, Plus, 512 and accelerated SE with SYS 7.0, 7.0.1 and 7.1. It is 32-bit clean and should be able to work with virtual memory (including Virtual from Connectix). Bug reports and suggestions are welcome and solicited. I promise to give my full and immediate attention to any important bug discovered.

Credits

Big thanks to Felix Kreisel for being alpha, beta, just tester and the first user as well for his suggestions and, most importantly, for making me to write this program.

Contact info

Send you suggestions, bug reports, and questions:

E-mail (preferred):

fridberg@pfc.mit.edu

US-mail

Mikhail Fridberg
7 Bennett St. #3
Brighton, MA 02135
USA

Index

7-bit ASCII 2
ASCII character set 2
character 2
character codes 2
font 2
glyph 2
low ASCII 2
Rich Text Format 4
Roman character set 3