VIRTUAL KEYBOARDS

VirtualKeyboards lets you change the behavior of your Lisp workstation keyboard to mimic another keyboard, hence making yours a "virtual" version of that other keyboard. It also lets you display pictures of keyboards on your screen and use them as menus for typing occasional special characters. Several keyboards may be displayed on the screen at once, letting you switch easily among keyboards for several languages and making hundreds of characters available for typing.

The virtual keyboards supplied with the module are Dvorak, German, Greek, Italian, logic, math, Spanish, European accents, and standard Russian. You can also define new keyboards with the associated Keyboard Editor module, which lets you edit a keyboard while seeing the actual look of the characters.

The virtual keyboards can be used with TEdit, DEdit, in the Lisp Executive window, for any application with which you use your keyboard.

Requirements

You need one of the following files, as appropriate for the machine you are using:

DANDELIONKEYBOARDS
DORADOKEYBOARDS
DOVEKEYBOARDS

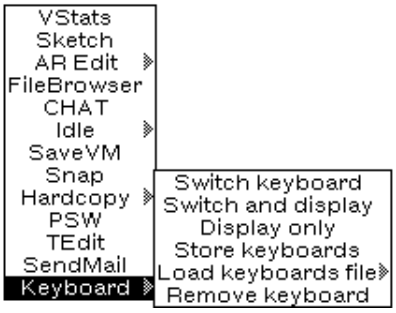
Installation

Load `VIRTUALKEYBOARDS.LCOM` from the library. VirtualKeyboards loads the xxxKEYBOARDS files.

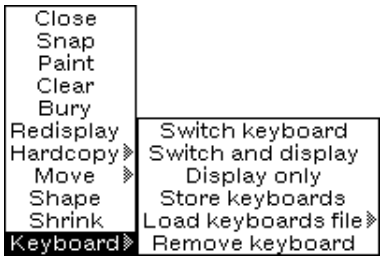
User Interface

Loading VirtualKeyboards adds the item KEYBOARD to the background menu and the default window menu. Using the mouse in this module is the same as in the KeyboardEditor.

Selecting this item from the background menu changes your keyboard for all windows.



Selecting this item from the default window menu allows you to specify a keyboard for an individual window.



The main keyboard module commands are described in detail below.

Switch Keyboards

Use the SWITCH KEYBOARD command to change the behavior of your keyboard to that of a selected virtual keyboard. This brings up a menu of the keyboards currently known to the program.

Select the keyboard you want to substitute for your workstation keyboard. Once you have changed your keyboard's behavior, pressing a key will send the character newly assigned to that key to the current input stream.

Switch & Display a Keyboard

Use the SWITCH AND DISPLAY command to change the behavior of your keyboard to that of a selected keyboard and, in addition, display that keyboard's layout on the screen. You will be offered a menu of the keyboards known to the program; select the one you want to substitute for your workstation's keyboard. Displaying the keyboard layout helps if you're typing on an unfamiliar keyboard. SWITCH AND DISPLAY lets you type characters by using the keyboard displayed on the screen as a menu.

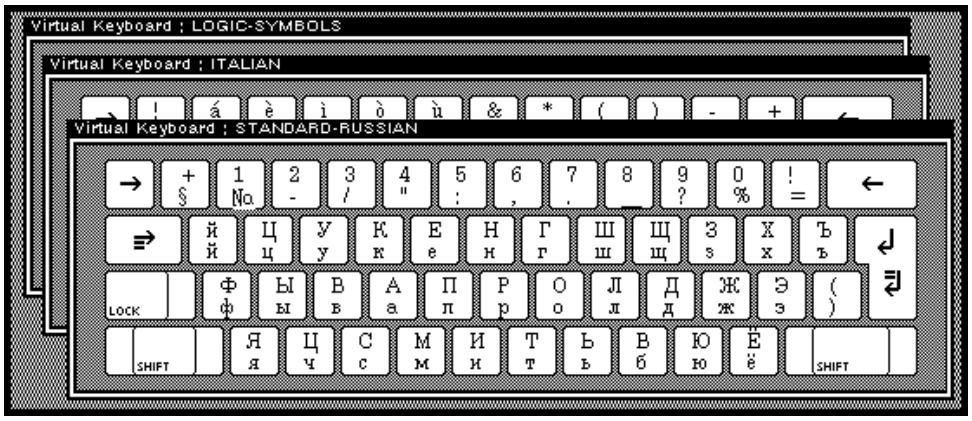


Figure 9. Keyboard layout display

Display-Only a Keyboard

You can display the layout for any given virtual keyboard using the DISPLAY ONLY command. You will be offered a menu of the keyboards known to the program (such as above); select the one you want to display. This is useful if you are primarily using the standard English keyboard but need to type some characters in other languages, or some special characters such as mathematical symbols.

You can use the displayed image as a menu: Selecting a key from the image with the left mouse button will send the character assigned to that key, and pressing the shift key while you click on a key will send the shifted character. Middle-clicking also sends the shifted character.

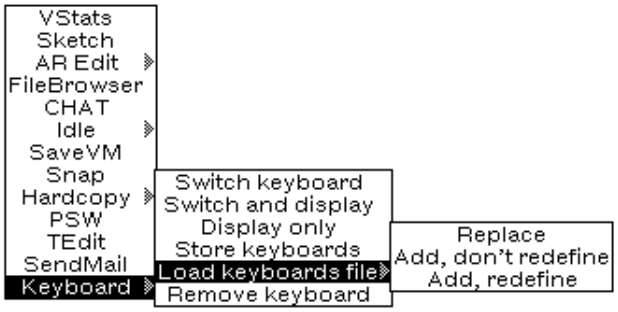
The effect is exactly as if you had pressed a key with that character assigned to it (except that interrupt characters are treated as ordinary characters; i.e., they do not cause an interrupt). The character is sent to the process that has the TTY (usually where the caret is flashing).

Store Keyboards

After you edit a keyboard (using the KeyboardEditor module), you can store it using the `STORE KEYBOARDS` command in the top-level menu. When you select `STORE KEYBOARDS`, the system will prompt you for a file name. After you type the file name into the prompt window, the system will store all the keyboards known to it (both new and old) in that file in a form that will enable it to load them.

Loading Keyboards File

To load a keyboards file, choose the `LOAD KEYBOARDS FILE` command, then slide the mouse cursor to the right and choose one of the three items in its submenu.



Replace All Known Keyboards

Choosing REPLACE will load a set of keyboards that you stored using the StORE KEYBOARDS command, replacing all the keyboards currently known to the system.

The currently known keyboards will be lost.

Add New Keyboards to the List of Known Keyboards

To add new keyboards without replacing any of the currently known keyboards, select ADD--DON'T REDEFINE. This will load a set of keyboard definitions.

If a keyboard in the file has the same name as one that is already known to the system, that keyboard will not be loaded and the current definition will stay in effect.

Load New Keyboards and Redefine Existing Keyboards

The ADD--REDEFINE command is similar to ADD--DON'T REDEFINE, except that it redefines existing keyboards that have the same name as keyboards on the file.

Currently known keyboards that do not have the same name as newly loaded keyboards will remain in the list of known keyboards.

Removing Keyboards From the Menu

To remove a keyboard from the set of currently known keyboards, select the REMOVE KEYBOARD command. This will pop up a menu of the known keyboards (such as above), from which you can select a keyboard to be deleted.

Defining a Virtual Keyboard

A virtual keyboard is a list whose CAR is the name of the keyboard and whose CDR is a list of key actions. Creating a new virtual keyboard can be done directly in Lisp or interactively, using the KeyboardEditor module.

Using the Functional Interface

The list of keyboards that are known to the program appears in the menu of keyboard names that pops up when you select SWITCH KEYBOARD from the background menu. This list is stored in the global variable `VKBD.KNOWN-KEYBOARDS` (see below). To add a keyboard to the list, you have to define that keyboard. To define a keyboard you can

either call the function DEFINEKEYBOARD or manipulate the variable VKBD.KNOWN-KEYBOARDS directly as explained herein.

You may also use the KeyboardEditor module, which provides a menu-based user interface for creating and changing keyboard layouts.

A virtual keyboard is a list of the form

(KEYBOARD-NAME *KEY-ASSIGNMENT1 KEY-ASSIGNMENT2* . . .)

A *KEY-ASSIGNMENT* is a list of the form

(*KEY* (*UNSHIFTED-CHAR SHIFTED-CHAR LOCK/UNLOCK*))

KEY is a key name (the character that appears on the actual keyboard).

UNSHIFTED-CHAR and *SHIFTED-CHAR* are character codes. Each can be either an integer representing the actual code or a list of two elements: the number of the character set and the number of the character in the set.

LOCK/UNLOCK is either the atom LOCKSHIFT, in which case *SHIFTED-CHAR* will be transmitted when the shift-lock key is down, or NOLOCKSHIFT, in which case the shift-lock key has no effect on that key. *LOCK/UNLOCK* is LOCKSHIFT by default.

(DEFINEKEYBOARD *KEYBOARD-NAME LIST-OF-KEY-ASSIGNMENTS KEYS-ARE-NUMBERS?*) [Function]

Creates a new virtual keyboard after parsing the list of key assignments and adds the keyboard to the list of known keyboards.

If *KEYS-ARE-NUMBERS?* is T, the function expects to find key numbers instead of key names.

(SWITCHKEYBOARDS *NEW-KEYBOARD SWITCH-FLG DISPLAY-FLG MENU-POSITION*) [Function]

Switches the current keyboard to *NEW-KEYBOARD*, where *NEW-KEYBOARD* is either a virtual keyboard or the name of a known keyboard.

If *SWITCH-FLG* is non-NIL, the actual key actions of the keyboard will be modified.

If *DISPLAY-FLG* is non-NIL, a window with a menu will be displayed. This displayed keyboard will act as a menu and will send characters to the current input stream when a character is selected.

VKBD.KNOWN-KEYBOARDS [Variable]

Contains the list of all currently known virtual keyboards.

Limitations

After loading the Dvorak keyboard, and then restoring defaults, you lose the shift- lock key.

[This page intentionally left blank]