KEYBOARDEDITOR

KeyboardEditor is intended for use with the VirtualKeyboards module. You should read that module's documentation before reading this. The KeyboardEditor module lets you create new virtual keyboards and change existing ones to suit your needs.

Requirements

VIRTUALKEYBOARDS

Installation

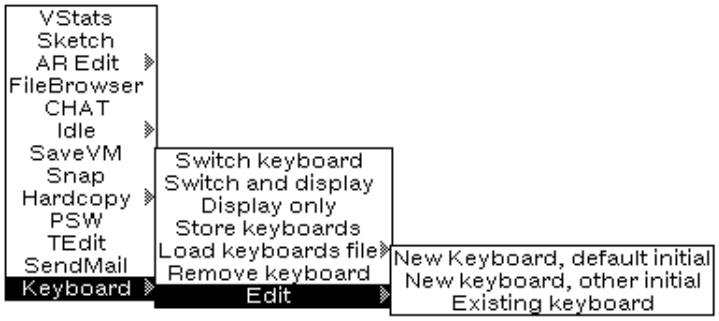
Load KEYBOARDEDITOR.LCOM and VIRTUALKEYBOARDS.LCOM from the library.

User Interface

Loading KeyboardEditor adds EDIT to the Virtual Keyboard submenu on the background menu.

Background Menu

The keyboard editor is used to modify and create virtual keyboards. You can call it by selecting EDIT from the main KeyboardEditor/VirtualKeyboards menu and sliding the cursor to the right to bring up the editor menu. You can also simply select EDIT, which gives you the same options as NEW KEYBOARD, DEFAULT INITIAL.



Creating a New Keyboard From a Copy of the Default Keyboard

Choose NEW KEYBOARD, DEFAULT INITIAL to create a keyboard from a copy of the default keyboard (which initially has the same key assignments as the 1108 keyboard). The system prompts you for a name for the new keyboard, then call the editor with a copy of the default keyboard as the initial keyboard. The key assignments that are not changed during the editing session remain as they are in the default keyboard.

Creating a New Keyboard From a Copy of Any Known Keyboard

To create a new keyboard from a copy of a known keyboard other than the default keyboard, select NEW KEYBOARD, OTHER INITIAL from the Edit submenu. You are

prompted for a name for the new keyboard. The system then displays a menu of the known keyboards from which to choose the initial keyboard.

Changing an Existing Keyboard

You can change an existing keyboard by selecting `EXISTING KEYBOARD` from the Edit submenu. Like the `NEW KEYBOARD`, `OTHER INITIAL` command, this brings up a menu of known keyboards from which you can choose a keyboard for editing. However, you are not prompted for a keyboard name first, because you are editing the actual keyboard rather than using it as a base for a new keyboard.

Calling the Keyboard Editor From Lisp

The editor can also be called using the function

`(EDITKEYBOARD` *KEYBOARD INITIALKEYBOARD*`)` [Function]

where *KEYBOARD* is either a virtual keyboard (i.e., a list) or the name of a virtual keyboard. If *KEYBOARD* is a virtual keyboard or the name of a known keyboard (a keyboard that was defined before), the editing is done on that keyboard and the second argument is ignored.

If *KEYBOARD* is a new name, the editing is done on a copy of *INITIALKEYBOARD*, with *KEYBOARD* as its new name. If *INITIALKEYBOARD* is NIL, the default keyboard is used as a base keyboard.

Examples:

To create a totally new virtual keyboard, call `(EDITKEYBOARD` *NEWNAME*`)`.

To create a new keyboard that is similar to a keyboard with the name K1, call `(EDITKEYBOARD` *NEWNAME* `'K1)`

To modify a keyboard with the name GREEK, call `(EDITKEYBOARD 'GREEK)`.

Using the Keyboard Editor

There are four different keyboard editor menus, three of them displayed at any given time. After you call the editor, the command menu is at the top, the character menu in the middle, and the keys menu at the bottom.

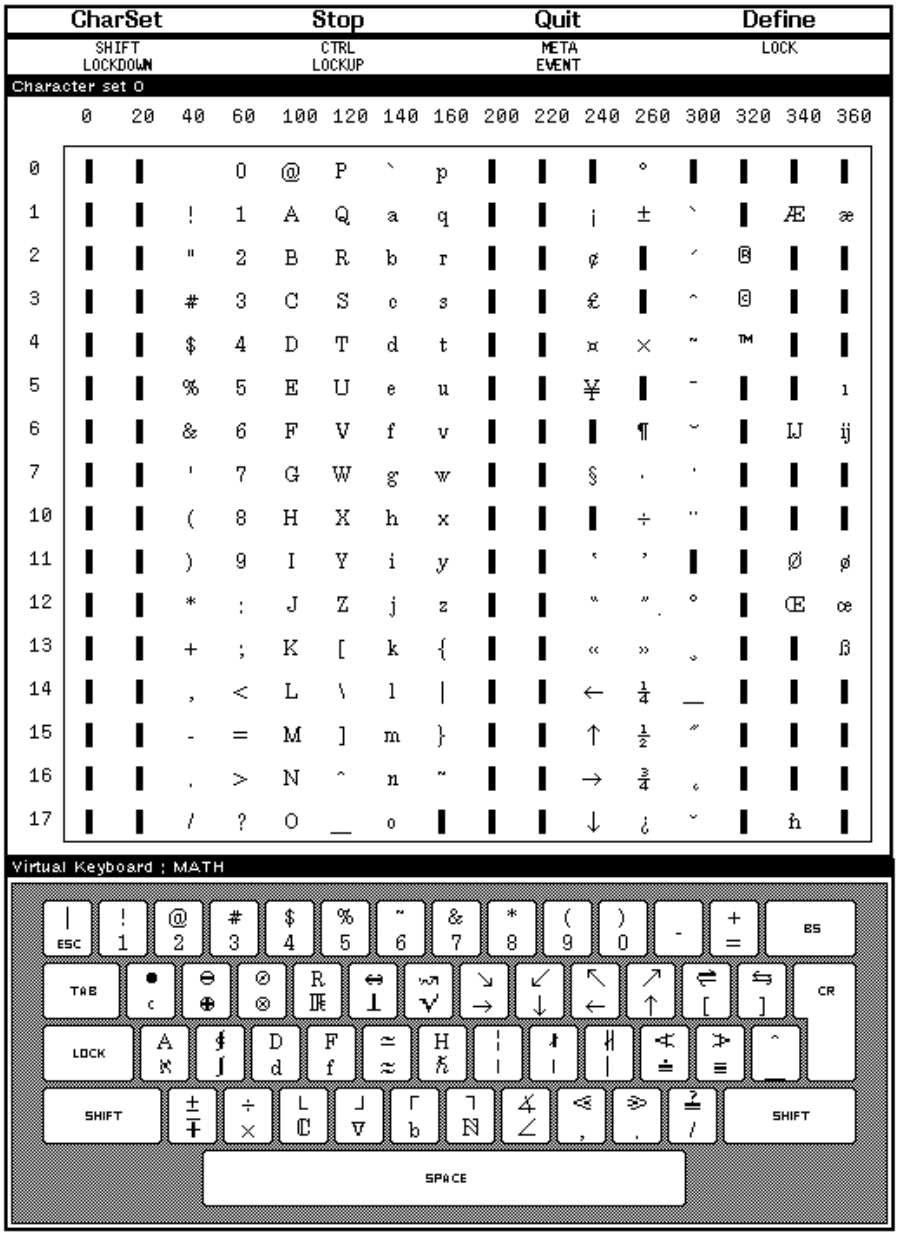


Figure 6. Character Display

The character menu is a 16-by-16-character display of the 256 characters available in the current character set. The set that is displayed when you enter the editor is character set 0, which includes all of the ASCII characters plus many other symbols. See Figure 6. If you need characters from other character sets, you have to select Char Set from the command menu. A new menu pops up that contains numbers from 0 to 377 octal. This is the character set menu, and it lets you switch the character menu to display characters from other sets. Most of the character set numbers are not currently implemented. The most useful ones are shown in Figure 7.

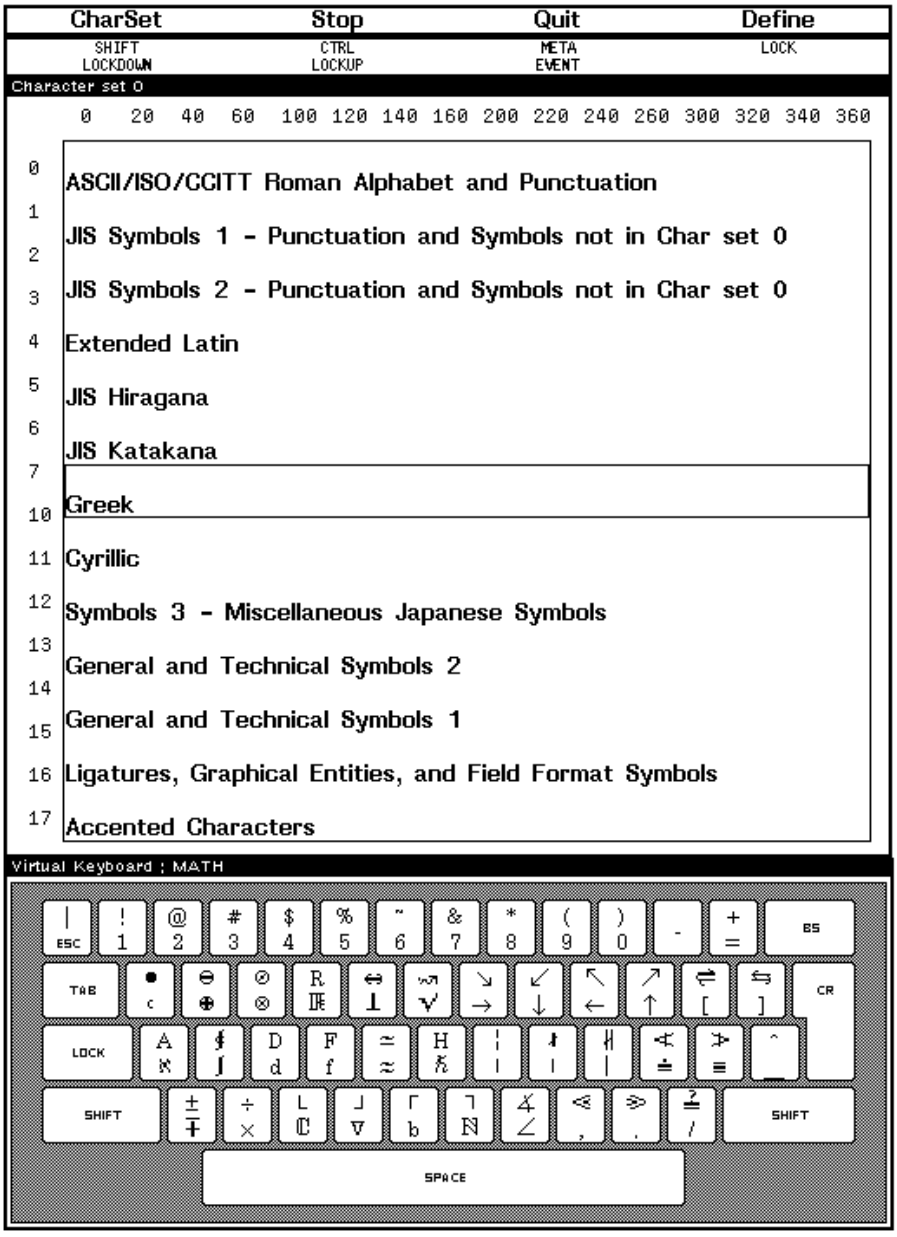


Figure 7. Character Sets

The keys menu lets you make a key the current key by selecting it. A selected key is marked by a black frame. To make a shifted key the current key, Shift-select the key (hold the Shift key down and click on the icon with the left button); it ise marked by inverted Shift keys in addition to the black frame.

The basic operation of editing is assigning a character to a key. You can only assign character keys; keys other than character keys retain their current definitions. You assign a character to a key by selecting the key from the keys menu, then selecting the character from the character menu. If the character is to be assigned to the shifted key, select the shifted key as the current key.

A second type of editing operation is to change the LOCKSHIFT state of a key. Each key either has or does not have a LOCKSHIFT property. If a key has a LOCKSHIFT property and the shift lock key of the keyboard is down, typing the key on your

workstation keyboard sends the shifted character of the key, regardless of the state of the shift keys. The same rule applies to a virtual displayed keyboard; if the LOCK item is inverted and the key has a LOCKSHIFT property, selecting a key sends the shifted character to the current input stream.

If a key has the LOCKSHIFT property, the lock key is inverted in the keys menu. To change the LOCKSHIFT property of a key, first make the shifted key the current key. Then set or unset the LOCKSHIFT property by selecting the lock key from the keys menu.

If you are creating a new keyboard and you are satisfied with the key assignments, select Define from the command menu. This adds the newly created keyboard to the list of known keyboards (it will thus appear on future menus). `Quit` exits after modifying the virtual keyboard; `Stop` exits without modifying the keyboard. In both cases the new keyboard is returned to the caller of `EDITKEYBOARD` function (above).

Creating New Keyboard Configurations

`KEYBOARDCONFIGURATION` [Record]

Describes a physical keyboard: its layout, the key numbers that are used with `KEYACTION`. It also describes each key: its default meaning, its default label, whether you can change the key's meaning with the keyboard editor.

A configuration consists of a number of parts:

`CONFIGURATIONNAME` [Record field]

The name of this configuration.

For example, KeyboardEditor comes with configurations named `DANDELION (1108)`, `DORADO (1132)`, `DOVE (1186)`, and `FULL-IBMPC`.

`KEYSIDLIST` [Record field]

An Alist of the IDs you use for the keys in the rest of the configuration; i.e., your names for the keys. For simplicity, these are usualy numbers starting beyond 100 (to avoid overlapping the true range of key numbers).

`KEYREGIONS` [Record field]

An Alist of key IDs and the regions they occupy in the keyboard's image when it is displayed. For example, the alphabetic keys in the DANDELION keyboard are 29 screen points wide and 33 high.

`DEFAULTASSIGNMENT` [Record field]

An Alist of key IDs and their default `KEYACTION`s (see *IRM*).

`KEYNAMESMAPPING` [Record field]

An Alist of key names to key IDs. The key names should be mnemonic, and should distinguish relevant differences; e.g., the 7 on the 1186's numeric keypad is named NUMERIC7, while the 7 key in the main keyboard cluster is named 7.

MACHINETYPE [Record field]

The kind of machine for which this configuration is intended.

For example, the FULL-IBMPC configuration is meant to be used with a DAYBREAK keyboard, so its MACHINETYPE is DAYBREAK.

KEYLABELS [Record field]

An Alist of key numbers to special labels. This is used to label keys such as the "Next" key, where the key assignment may not be a printable character.

KEYLABELSFONT [Record field]

The font you want to use for the key labels. The default value is Helvetica 5.

BACKGROUNDSHADE [Record field]

The shading for the non-key parts of the virtual keyboard's image. This defaults to a reasonable gray value.

KEYBOARDDISPLAYFONT [Record field]

The font used to display actual character assignments. This should probably be Classic 12, since it is the most complete font.

CHARLABELS [Record field]

An Alist from character codes to names. Used to give symbolic names to characters such as ESCAPE, which do not otherwise print.

ACTUALKEYSMAPPING [Record field]

A function that takes one of your key IDs and returns a true key number, for use by KEYACTION.

Note: To create a new configuration, create an instance of the KEYBOARDCONFIGURATION record, using the field names shown above. Then add it to the list VKBD.CONFIGURATIONS. You may then edit it using the configuration editor described below.

Note: You must save your own configurations. There is no user interface for saving them, nor any automatic scheme.

Editing a Keyboard Configuration

Once you have created a KEYBOARDCONFIGURATION, you can make modest changes to it using the function:

(EDITCONFIGURATION *CONFIGNAME*) [Function]

where C*ONFIGNAME* is the CONFIGURATIONNAME you have assigned to your new configuration. This creates a virtual keyboard editing window with a menu on top of it as shown in Figure 8.

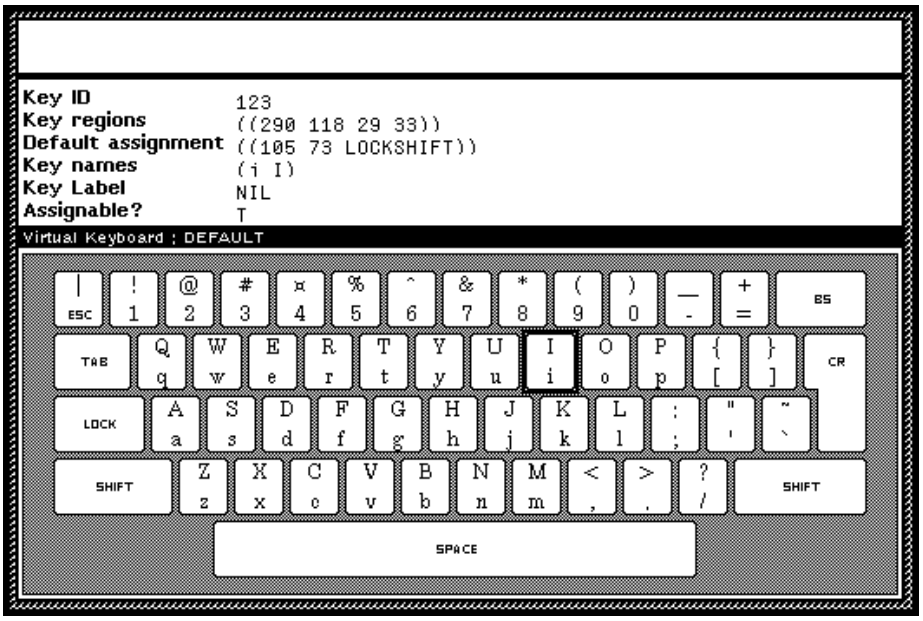


Figure 8. Virtual Keyboard Editing Window

Selecting a key with the mouse fills in the fields in the menu. The figure shows the 1108's configuration being edited, with the I key selected. To change one of the values, select the label at the left edge of the menu (e.g., ASSIGNABLE?). You are prompted to edit the existing value using TTYIN.

The keyboard image is not automatically updated. To refresh it, select REDISPLAY in the right-button window menu.

When you have finished editing, simply close the keyboard window.

[This page intentionally left blank]