$[1] #[2] +[3] Glossary

Argument
Binding
Buffer
Clipboard
Command
DOS Box
File Locking
Function
Group
Keyboard Macro
Keystroke
Kill Buffer
Kill Ring
Macro
Mark
Message Line
Meta Key
Mode Line
Mouse Action
Page
Path
Point
Popup Buffer
Region
Screen
Selection
Variable
Window

[1] $ Glossary

[2] # Glossary

[3] + Index:6010

$[4] #[5] +[6] K[7] A **DOS Box** is a Microsoft Windows feature within which DOS programs are executed. In Windows "386-enhanced" mode, a DOS box can appear as an icon, a window or it can occupy the whole screen. In Windows "standard" mode, DOS programs can execute only when their DOS box occupies the whole screen.
Under Windows NT, the equivalent of DOS Boxes are named "shell boxes"

[4] $ DOS box (glossary)
[5] # DosBox
[6] + Glossary:dosbox
[7] K DOS box

$[8] #[9] +[10] K[11] **Commands** are built in functions that represent basic things that MicroEMACS does. For example, the up arrow key activates the "previous-line" command which moves the cursor up to the line of text immediately before the current line.

[8] $ Commands (glossary)

[9] # Command

[10] + Glossary:command

[11] K commands;macro

$[12] #[13] +[14] K[15] A **binding** is a link between a sequence of keys and a command or macro. For instance, the command "previous-line" is bound to the up-arrow key, and to the ^P key. Pressing a key sequence causes the command to which it is bound to execute.

Under Microsoft Windows, commands and macros can also be bound to menu items.

[12] $ Binding (glossary)
[13] # Binding
[14] + Glossary:binding
[15] K binding

$[16] #[17] +[18] K[19] The **meta key** is the key used to start many commands. On most keyboards this is the Escape key, but many times it is rebound/changed to the key in the upper left corner of the keyboard. This is often the grave accent symbol.

16[$] Meta key (glossary)
17[#] MetaKey
18[+] Glossary:metakey
19[K] meta key

$[20] #[21] +[22] K[23] Interactively, a **numeric argument** is supplied by typing the meta key (usually the Escape key), followed by a decimal number, before invoking a command.

Within the macro language, a numeric argument is placed before the name of the associated command.

[20] $ Numeric argument (glossary)
[21] # Argument
[22] + Glossary:argument
[23] K numeric argument

$[24] #[25] +[26] K[27] **Buffers** are areas of memory set aside to hold text. Each buffer has a buffer name which is used to refer to it, and a file name from which it has been read or where it will be written.

[24] $ Buffers (glossary)
[25] # Buffer
[26] + Glossary:buffer
[27] K buffer

$[28] #[29] +[30] K[31] **Popup Buffers** are a way to display a buffer temporarily, without using a window. When a popup buffer is displayed, it occupies the whole screen. If more than one screenfull is needed, the text "-- more --" appears on the message line. The next screenfull can be viewed by pressing the space bar. Pressing any other key cancels the popup buffer and the keystroke is then processed by MicroEMACS.

[28] $ Popup buffers (glossary)

[29] # PopupBuffer

[30] + Glossary:popupbuffer

[31] K popup buffer

$[32] #[33] +[34] K[35] **Windows** are sections of the current screen which display a portion of a buffer. More than one window may be visible at a time. Multiple windows split the screen horizontally.

Notice that the MicroEMACS usage of the word window is different from the meaning used in window-based systems:

MicroEMACS	Operating System
Window	Pane
Screen	Window

[32] $ Windows (glossary)

[33] # Window

[34] + Glossary:window

[35] K window

$[36] #[37] +[38] K[39] **Screens** are collections of windows. On a older text style system, one screen is displayed at a time. On a newer window based system, like OS/2, the Macintosh or Microsoft Windows, each operating system window can display a different MicroEMACS screen.

Notice that the MicroEMACS usage of the word window is different from the meaning used in window-based systems:

MicroEMACS	Operating System
Window	Pane
Screen	Window

[36] $ Screens (glossary)

[37] # Screen

[38] + Glossary:screen

[39] K screen

$[40] #[41] +[42] K[43] The **mode line** is the line at the bottom of each window naming the buffer being displayed, along with its file name. Also the active modes of the window are shown.

40[$] Mode line (glossary)
41[#] ModeLine
42[+] Glossary:modeline
43[K] mode line

$[44] #[45] +[46] K[47] The **command line** or **message line** is the line at the bottom of the screen where you give more information to some commands and also receive information or error messages.

[44] $ Message line (glossary)

[45] # MessageLine

[46] + Glossary:messageline

[47] K message line;command line

$[48] #[49] +[50] K[51] **Macros** (also called **procedures**) are programs written in the MicroEMACS language which let you customize the editor and, in particular, automate repetitive editing tasks.

[48] $ Macro (glossary)
[49] # Macro
[50] + Glossary:macro
[51] K macro

$[52] #[53] +[54] K[55] A **keyboard macro** is a remembered sequence of keystrokes which can be used to greatly speed quick and dirty repetitive editing.

[52] $ Keyboard macro (glossary)

[53] # KeyboardMacro

[54] + Glossary:keyboardmacro

[55] K keyboard macro;macro

$[56] #[57] +[58] K[59] **Pages** are groups of macros which have been written to handle a particular editing task, and which have been packaged to be available from the MicroEMACS startup file. These files usually have a filename extension of ".CMD".

The MS-Windows version of MicroEMACS is bundled with sample macro pages called CUA.CMD, DEV.CMD and MDI.CMD.

[56] $ Page (glossary)
[57] # Page
[58] + Glossary:page
[59] K page;macro

$[60] #[61] +[62] K[63] The **path** is a list of directories that MicroEMACS searches for the following files:
EMACS.RC (the startup file)
The argument of the execute-file command
The argument of the &find function
The default DOSEXEC.PIF and DOSBOX.PIF files
EMACS.HLP (for the help command)

The following items compose the **path** (in order of decreasing priority):

1. The directory specified by the HOME system variable (or, under MS-Windows, the directory where the MicroEMACS executable resides).
2. The current working directory.
3. The directories specified in the PATH system variable.
4. The following directories (MS-DOS-based or Windows NT systems only. Other implementations use different lists):

\sys\public
\usr\bin
\bin
\

[60] $ Path (glossary)
[61] # Path
[62] + Glossary:path
[63] K path

$[64] #[65] +[66] K[67] The **point** is the position of the cursor in the text of the current window. The point can be considered to lie between the character the cursor rests on and the one immediately after it.

[64] $ Point (glossary)

[65] # Point

[66] + Glossary:point

[67] K point

$[68] #[69] +[70] K[71] The **mark** is the position in the current buffer which delimits the beginning or the end of a region. Various commands operate on text from the mark to the point, or move the current point to the mark. The mark can be set by the set-mark command.

Each buffer contains 16 independent marks, numbered 0 to 15. Most region-related commands, however, only refer to mark 0.
Marks 10 to 15 are reserved for use by macros. Marks 0 to 9 are reserved for interactive use. Marks 11 and 12 are used internally by MicroEMACS by the mouse-region-down and mouse-region-up commands.

[68] $ Mark (glossary)
[69] # Mark
[70] + Glossary:mark
[71] K mark;region

$[72] #[73] +[74] K[75] A **region** is the text located between the point (i.e. the position of the cursor) and the mark number 0. The mark can be set by the set-mark command.

[72] $ Region (glossary)
[73] # Region
[74] + Glossary:region
[75] K region;mark

$[76] #[77] +[78] K[79] The **selection** is available only if the macros from the CUA.CMD page have been loaded. It is the piece of text that has been selected by dragging the mouse (with the left button held down) over it, or by moving (with the arrow or the page keys) through the text with the Shift key held down.

The CUA.CMD file is distributed as part of the MicroEMACS for Windows package.

In the current version of MicroEMACS, the selection is not highlighted.

[76] $ Selection (glossary)

[77] # Selection

[78] + Glossary:selection

[79] K selection;clipboard

$[80] #[81] +[82] K[83] The **clipboard** is a temporary storage area. Text can be cut or copied to the clipboard from a Windows application and be pasted into another application.

[80] $ Clipboard (glossary)
[81] # Clipboard
[82] + Glossary:clipboard
[83] K clipboard;selection

$[84] #[85] +[86] K[87] **Variables** are elements of the MicroEMACS macro language. They carry numeric, boolean or string values.

Variables that begin with a dollar sign "**$**" are called environmental variables. They control various aspects of the editor.

[84] $ Variables (glossary)
[85] # Variable
[86] + Glossary:variable
[87] K variable;macro

$[88] #[89] +[90] K[91] **Functions** are elements of the MicroEMACS macro language. Functions have arguments and return numeric, boolean or string values.

Function names begin by an ampersand "**&**". Only the first 3 characters of a function name are significant.

[88] $ Functions (glossary)
[89] # Function
[90] + Glossary:function
[91] K function;macro

$[92] #[93] +[94] K[95] **Groups** can be used with text substitution commands or macros in MAGIC mode, to duplicate parts of the target into the result.

In the search string, a group is defined as a portion beginning by the characters backlash and opening parenthesis "**\(**" and ended by the characters backlash and closing parenthesis "**\)**". There can be up to nine such groups.

In the replace string, groups appear as a backlash followed by a decimal digit ("**\1**" to "**\9**"). The portion of the target string matched by the *n*th group is substituted to each occurrence of \n to form the replacement string.

The function &group *n* can be used in macros to obtain the text matched by the *n*th group in a search.

[92] $ Groups (glossary)
[93] # Group
[94] + Glossary:group
[95] K group;search;replace

$[96] #[97] +[98] K[99] MicroEMACS may implement **file locking** to prevent simultaneous access of the same file by different MicroEMACS instances. The method used for this is dependant on the base operating system.

File locking is active only if MicroEMACS was compiled with a specific "FILOCK" option. Standard release versions usually do not implement file locking.

[96] $ File Locking (glossary)

[97] # FileLocking

[98] + Glossary:filelocking

[99] K file locking;lock

$[100] #[101] +[102] K[103] The **kill buffer** accumulates any text which is "killed" by a number of delete commands. If more than one delete command is used in a row, all the text from all the commands will be in the kill buffer. Using any command between deletes causes the kill buffer to just hold the most recent deletions.

Using this feature and the yank command, you can switch between windows, screens and files and copy text from one file to another.There is no limit to the amount of text that can be stored in the kill buffer except that of the memory of the computer running MicroEMACS. Extremely large kills may take a few seconds.

The last 16 kill buffers are kept in the kill ring. You can retrieve their contents through the cycle-ring or the yank-pop commands.

[100] $ Kill Buffer (glossary)
[101] # KillBuffer
[102] + Glossary:killbuffer
[103] K kill;copy;move;delete

$[104] #[105] +[106] K[107] The **kill ring** is a circular list of the last 16 kill buffers. The position of the current kill buffer can be changed by the cycle-ring and the yank-pop commands. The kill ring can be emptied (and thus the used memory reclaimed) by using the delete-kill-ring command.

[104] $ Kill Ring (glossary)
[105] # KillRing
[106] + Glossary:killring
[107] K kill;copy;move;delete

$[108] #[109] +[110] K[111] Mouse Syntax

Key bindings can include mouse actions which are represented as follows:

	Press	Release
Left button:	**MSa**	**MSb**
Center button:	**MSc**	**MSd**
Right button:	**MSe**	**MSf**
Shift+Left button:	**S-MSa**	**S-MSb**
Shift+Center button:	**S-MSc**	**S-MSd**
Shift+Right button:	**S-MSe**	**S-MSf**
Ctrl+Left button:	**MS^a**	**MS^b**
Ctrl+Center button:	**MS^c**	**MS^d**
Ctrl+Right button:	**MS^e**	**MS^f**

Mouse movement: **MSm**
Dropping files dragged from the MS-Windows File Manager: **MS!**

[108] $ Mouse Syntax (glossary)

[109] # MouseAction

[110] + Glossary:mouseaction

[111] K mouse;binding

$[112] #[113] +[114] K[115] **Keystroke Syntax:**

In key bindings, regular characters are represented by the corresponding uppercase, preceded by a hat "**^**" sign if the Ctrl key is depressed. For instance, for Ctrl+G: **^G**.

Ctrl+SPACE is represented by **^@**.

Function keys are represented as:

F1 to F9, F10:	**FN1** to **FN9**, **FN0**
Arrows:	up **FNP**, down **FNN**, left **FNB**, right **FNF**
Page keys:	up **FNZ**, down **FNV**
Other keys:	Home: **FN<**, End: **FN>**, Insert: **FNC**, Del: **FND** (or **^?**)

If the Ctrl key is depressed for a function key, the hat "**^**" is located before the last char. For instance, for Ctrl+F1: **FN^1**.

The prefix, if any, appears before the keystroke:

M-	the meta key (usually the Escape key) is depressed and released.
^X	the Ctrl+X keys are depressed and released.
A-	the Alt key is depressed.
S-	(function keys only) the Shift key is depressed.

112[$] Keystroke Syntax (glossary)

113[#] Keystroke

114[+] Glossary:keystroke

115[K] keyboard;binding