

JASSPA
**Getting Started with
JASSPA MicroEmacs**
Version 1.11
Jon Green

JASSPA

www.jasspa.com

support@jasspa.com



JASSPA. www.jasspa.com
support@jasspa.com

Copyright © 2004-2006 JASSPA.

Title: Getting Started with JASSPA MicroEmacs
Reference: jasspame
Version v1.11
Date: 2006/10/07 14:42:25

Typeset with the TexLive 2004 L^AT_EX Documentation System under Sun Solaris 9.

Date	Who	Description	Revision
2006/08/29	Jon	Updated for the 2006 release.	1.11
2005/03/14	Jon	Updated with Buffer Setup information. Added section MicroEmacs Basics.	1.8
2004/10/25	Jon	Draft.	1.0



Preface

Time is short, armed with a reasonable \LaTeX boiler plate it is time to put down some usable notes on getting up and running with JASSPA MicroEmacs. The on-line help that comes with JASSPA MicroEmacs details every public function and variable in the editor however it is not that accessible and certainly does not help the new user.

In this first draft, that will undoubtedly grow over time, we simply present the post installation configuration to help the new user get up and running with the editor.

Jon Green

25th October 2004.



Contents

1	Installation and Setup	6
1.1	Installation	6
1.1.1	Packaged Sun Solaris	6
1.1.2	Packaged Red Hat Linux/Fedore Core	6
1.1.3	Packaged HP-UX	6
1.1.4	Packaged Microsoft Windows	7
1.1.5	Non-packaged UNIX/Linux/BSD - System wide	7
1.1.6	Non-packaged UNIX/Linux/BSD - User Installation	7
1.1.7	MS-DOS	8
1.1.8	Windows Manual Installation	8
1.1.9	Spelling Dictionaries	9
1.2	Building from Source	9
1.3	Running For the First Time	10
1.4	Help Information	10
1.4.1	Mouse Interaction	10
1.4.2	Keyboard Interaction	10
1.5	User Setup	10
1.5.1	Common Options	11
1.6	Buffer Setup	13
1.7	Major Modes	14
1.8	MicroEmacs Integration	15
1.8.1	Desktop Integration	16
1.8.2	Application Integration	17
1.8.3	Utilities Integration	17
1.8.4	Microsoft Windows UNIX utilities	17
1.8.5	Microsoft Windows and Cygwin	18
2	MicroEmacs Basics	19
2.1	Getting Started	19
2.2	MicroEmacs Screen	19
2.3	Files, Buffers and Windows	20
2.4	Modes	20
2.5	Keys and the Keyboard	21
2.6	Entering Text	21
2.7	Cursor Movement	21
2.8	Abort and Exiting	22
2.9	Loading and Saving Files	22
2.10	Insertion and Deletion	23
2.10.1	Region Copy/Delete	23
2.10.2	Yank (or paste)	24



2.10.3 Undo	24
2.11 Buffers and Windows	24
2.11.1 Buffers	24
2.11.2 Windows	24
2.12 Searching and Replacing	25
2.12.1 Searching	25
2.12.2 Replacing	26
2.13 Named Command Execution	26
2.14 Universal Repeat	26
2.15 Mouse Interaction	26



1 Installation and Setup

1.1 Installation

To install JASSPA MicroEmacs then download the appropriate bundle from www.jasspa.com. Most of the common platforms include packages to simplify installation.

1.1.1 Packaged Sun Solaris

As root install the package.

```
cd /tmp
unzip jasspa-mepkg-sun-sparc-59-yyyyymmdd.zip
pkgadd -d jasspa-me
```

The package is installed in `/opt/jasspa`. Edit your local login script file to include the executable on your local path e.g. `.cshrc` or `.profile` – the syntax depends on the shell type that you use.

```
#
# Set up Microemacs
#
if [ -d /opt/jasspa ] ; then
    PATH=$PATH:/opt/jasspa/bin
    MANPATH=$MANPATH:/opt/jasspa/man
fi
...
export PATH
```

The package may be subsequently removed using:

```
pkgrm jasspa-me
```

1.1.2 Packaged Red Hat Linux/Fedore Core

As root install the package.

```
rpm -i jasspa-me-yyyyymmdd-1.i386.rpm
```

The package is installed in directory `/usr/share/jasspa`, the executable is installed in `/usr/bin` and requires no additional setup.

The package may be subsequently removed using:

```
rpm -e jasspa-me
```

1.1.3 Packaged HP-UX

For HP-UX then the package may be installed via SAM, but is easier to install using the command line:

```
/usr/sbin/swinstall -s `pwd`/jasspa-mepkg-hpux-pa-10.20-yyyyymmdd.depot jasspa-me
```



The package is installed in `/opt/jasspa`. The executable is automatically added to the shell profile environment. The user should log out and log back in again to pick up the new environment settings which are added to `/etc/profile`.

The package may be subsequently removed using:

```
swremove jasspa-me
```

1.1.4 Packaged Microsoft Windows

Microsoft Windows environments 95/98/NT/2K/XP use an *Install Shield* package installation which may be used to add and remove the package. The software is installed into `c:\Program Files\JASSPA\MicroEmacs`.

Unpack the *zip* file, run **setup.exe** and follow the installation procedure.

1.1.5 Non-packaged UNIX/Linux/BSD - System wide

For other UNIX/BSD environments where no package is provided then a manual installation is performed. Select the installation directory i.e. `/usr/local`.

```
cd /usr/local  
gunzip -c /path-to/jasspa-metree-yyyyymmdd.tar.gz | tar xvf -
```

or using GNU tar

```
cd /usr/local  
gtar zxvf /path-to/jasspa-metree-yyyyymmdd.tar.gz
```

The directory `/usr/local/jasspa` should now exist. Unpack the binary into the `/usr/local/bin` directory and make it executable.

```
cd /usr/local/bin  
gunzip -c /path-to/jasspa-me-linux-2.x-yyyyymmdd.gz > me  
chmod a+rx me
```

The installation is complete, `/usr/local/bin` is assumed to be on the executable path.

1.1.6 Non-packaged UNIX/Linux/BSD - User Installation

For other UNIX/BSD environments where a local user install is required (i.e. the user has no permissions to write to the system directories) then the following steps may be followed.

Unpack *metree* in your home directory, this creates a directory tree called **jasspa**, once unpacked then rename the directory to **.jasspa**.

```
cd ~  
gunzip -c /path-to/jasspa-metree-yyyyymmdd.tar.gz | tar xvf -  
mv jasspa .jasspa
```

Add downloaded spelling dictionaries by unpacking them into the `~/ .jasspa/spell` subdirectory. Install the binary into your local binary directory - we assume `~/bin` e.g.



```
cd ~/bin
gunzip -c /path-to/jasspa-me-linux-2.x-yyyyymmdd.gz > me
chmod a+rx me
```

Assuming that the binary directory is already on the users path then the installation is complete.

1.1.7 MS-DOS

For MS-DOS then no installation script is provided and a manual installation is performed. Select the installation directory, we shall assume `c:\jasspa`.

Unpack the `metree` zip package to form the macro tree, this creates the directory `jasspa`.

```
c:
cd \
unzip jasspa-metree-yyyyymmdd.zip
```

The binary is placed in the `c:\jasspa` directory. It is important that the binary is placed in the JASSPA tree as this allows the executable to locate the `macros` and `spelling` directories without any further configuration. If the binary is placed in a separate location from the tree then the `$MExxxPATH` environment variables must be defined to enable the editor to locate the tree.

Two different binaries are provided which are built with DJGPP V1.0 and V2.0. If you have DJGPP V2.0 installed in your system then install the V2.0 binary, otherwise install the V1.0 binary (as this is a self contained executable and has no run-time library dependencies).

```
cd \jasspa
unzip jasspa-me-msdos-djgpp1-yyyyymmdd.zip
```

The location of the executable should be added to the `autoexec.bat` file e.g.

```
SET PATH=%PATH%;c:\jasspa
```

The system may be re-booted and the editor should be available on the command line `me`.

1.1.8 Windows Manual Installation

Manual installation is required under **Win32s** and may be optionally performed in **Win32** environments (95/98/NT/2K/XP). This is very similar to the **MS-DOS** installation.

Unpack the `metree` zip package to form the macro tree, this creates the directory `jasspa`. This is typically extracted to `c:\Program Files` (**Win32s** may use `c:\me`).

```
c:
cd "\Program Files"
unzip jasspa-metree-yyyyymmdd.zip
```

The executable is placed in the `jasspa` directory:

```
cd "\Program Files\jasspa"
unzip jasspa-me-win32-yyyyymmdd.zip
```

Unlike **MS-DOS** then short cuts are typically used to launch the editor. The editor short-cut is typically edited with command line option `-c` to restore the previous session when launched.



1.1.9 Spelling Dictionaries

The spelling dictionaries are downloaded separately and are not included in the base packages, multiple dictionaries may be downloaded and installed for each language to be supported. Some languages include a *base* and *extended* dictionary, the *base* contains the most popular words, the *extended* includes more obscure words over and above the *base*. The *base* and *extended* dictionary are packaged together, but the *extended* dictionary may be removed when downloaded to conserve disk space.

The spelling archive contains both the dictionaries and language specific macros required to install the dictionary into the system. All of the dictionaries are derived from *ispell* dictionaries. The Copyrights for each of the dictionaries are contained in the macro files.

To install then the archive is unpacked into the *jasspa/spelling* directory that was created when *metree* was unpacked. Simply unpack the contents of the archive into this directory.

```
cd /path-to/jasspa/spelling
unzip enus.zip
```

With the directory installed then set up MicroEmacs using **Help** → **User Setup** or from the command line **M-x user-setup**.

- Select the **Platform** tab.
- Select the **Language** from the drop-down box.
- Enable **Auto Save Dictionaries** to automatically save the dictionary when new words are added or ignored.
- Enable **Enable Auto-Spell** if the auto spell facility is required. Auto-Spell performs background spell checking and highlights erroneous words.

1.2 Building from Source

Where a particular platform does not exist then MicroEmacs may be built from source. It is expected that the environment is set up with the appropriate tools to perform the build.

A simple build script is provided for both UNIX and DOS/Windows systems called **build**. Download the source bundle *jasspa-mesrc-yyyymmdd* and unpack, this creates a directory **meYYMMDD**. Change directory to **src** and issue a build command:

```
# DOS
c:> build
```

```
# UNIX
>sh> ./build
```

On most systems then **build** is performed, alternatively an explicit **make** operation may be used. As of 2006 then the build script is the preferred method of compiling on UNIX as the script file performs a number of tests to locate the X-Windows **xpm** library and configures the makefile appropriately.

A selection of *make* files exist for different platforms **.mak** indicates a native platform compiler and **.gmk** indicates a GNU compiler.

```
make -f platform.mak
```



The platform specific makefiles may be modified with compiler specific options for the target platform to produce a more optimal executable.

For Microsoft Windows environments then a Microsoft Project file is provided.

1.3 Running For the First Time

On running MicroEmacs for the first time with the command **me** then the user is asked to set up the environment. This process collects some very basic information and creates the user private files that store context and state. The user may decline to perform the set-up and is prompted next time the editor is executed.

The environment is analyzed and the system directory locations and search paths are determined. These can typically be accepted as is.

- **Enter you full name ?** This is *your* name that is inserted in the header files that identify you as the author.
- **Set up the company file (y/n) ?** If you are working in a company environment then this allows you to enter the name of the company that is used in copyright statements automatically inserted into new headers etc.
- **Name the company file ?** This is the name of the macro file that stores company wide company information. The default is simply called **company** and is typically accepted. If you are using a site wide installation then your system administrator may set up a specifically named company file which should be entered here.
- **Company Name ?** the long name of the company i.e. *Acme Building Inc.*

The configuration is complete and the top level help page is displayed. The user should now perform the **User Setup** step detailed in a later section.

1.4 Help Information

The help information is available on-line and may be accessed using the mouse or the keyboard.

1.4.1 Mouse Interaction

Help→General Help

The links are shown highlighted and may be traversed with the mouse by selecting the link with a double click of the left button.

1.4.2 Keyboard Interaction

esc-x help

The links may be traversed using the TAB key to move to the next link, a RETURN on the link causes the link to be followed.

1.5 User Setup

On completion of the basic configuration then a certain amount of customization is typically required. The *User Setup* dialog allows the user to perform this process and is invoked as follows:

`esc-x user-setup ... OR`
Tools→User Setup

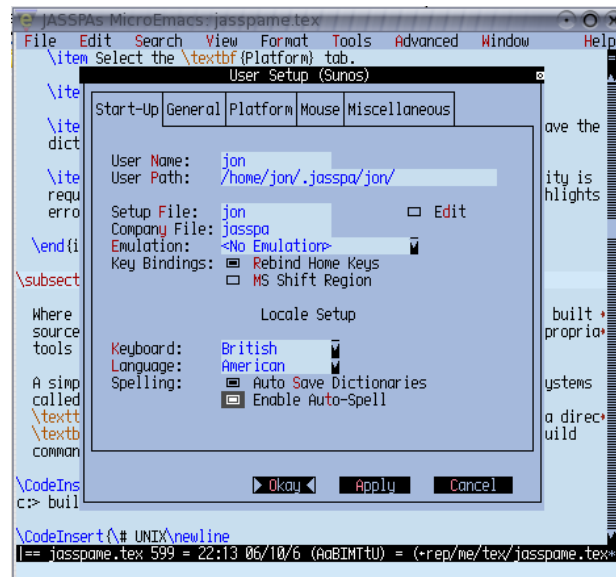


Figure 1: User Setup Dialog

Navigation within the dialog is performed using the mouse and/or TAB and cursor keys. The SPACE key is used to open a drop down dialog. The configuration may be saved permanently using **Save** and installed into the current session using **Current**.

1.5.1 Common Options

For a quick setup then the *User Setup* dialog is traversed by tab, the most common options are described below, other options may be ignored. In a console environment use the cursor keys to move from tab to tab whilst the tab is in focus.

- **Start-up** – start up configuration information.
 - **Setup File** the name of the user setup file, excluding the extension.
 - **Edit** check box determines whether the *user.emf* is presented for editing when the user setup is changed. The default is enabled but is typically disabled.
 - **Company File** the name of the company file, excluding the extension.
 - **Emulation** select the preferred emulation mode of the editor. The standard key bindings are adjusted to be more consistent with the selected emulation. If the emulation is changed then it is advisable to re-start the editor when *User Setup* is complete.
 - **Re-bind Home Keys** when disabled then the keys *Home* and *End* move to the beginning and end of the buffer respectively. When enabled then *Home* and *End* move to the beginning and end of the line respectively.
 - **MS Shift Bindings** enabled performs Microsoft region selection using the cursor keys whilst the SHIFT key is pressed.
 - **Keyboard** select your keyboard type, the keyboard bindings are corrected for the locale.



- **Language** the default spelling language. This has no effect if the dictionaries are not loaded.
- **Auto Save Dictionaries** typically enabled, causes the user defined spelling dictionary to be automatically saved on exiting the editor.
- **Enable Auto-spell** enable this option to allow MicroEmacs to perform spell checking whilst you type. When enabled then spelling errors are highlighted to indicate errors, to correct a misspelling with auto-spell enabled then mouse right click on the mis-spelt word and select an auto correction from the list.
 - M-x auto-spell-buffer** to automatically spell check the whole buffer, typically enabled. The buffer is automatically spell checked on entry and erroneous spellings are highlighted, a right mouse click on the spelling error presents a list of alternative correct spelling guesses.
 - M-x spell-buffer** to spell check the buffer via spell dialog.
- **General** – general defaults and settings.
 - **Full Name** user name that is inserted into new file templates.
 - **Organizer File** defines the organizer file base name, defaults to the log name.
 - **Auto-Save Time** defines the length of time in seconds between buffer automatic saves, a setting of 0 or an empty string disables auto-saving. The default setting is 300 seconds or 5 minutes.
 - **Keep Undo** when enabled then the undo history is kept after a save allowing the **undo** command to back-up changes beyond the last save. The default is disabled and the undo history is discarded after the buffer is saved.
 - **Hide Backups** enables MicroEmacs generated backup files to be hidden. On Windows and DOS platforms the *Hidden* file attribute is used to hide the file, on UNIX the backup file name is prepended with a dot `.`.
 - **Main Menu** disable if the Main Menu is not required.
 - **Alt Action** defines the behavior of the ALT key. Enable **Esc Prfx** if the ALT key may be used as a ESC key prefix (most Emacs users). Disable **Main menu Hot-keys** if the ALT key should not be used as a hot key to the Main menu. When both are enabled then the Main menu has priority.
 - **Abbrev Setup** defines the behavior of the abbreviation key binding **esc esc**.
 - * Enabling **Accent** completes accented characters i.e. :o<esc><esc> is converted to a ö.
 - * Enabling **Lookback** attempts to complete the word based on previous words in the buffer. A further **esc esc** finds the previous etc.
 - * Enabling **Dict'n** attempts to complete a word based on the spelling dictionary.
 - **Tab to Indent** controls the behavior of the TAB key. By default, where indent rules exist for a buffer then the TAB key causes the indentation of the line to be re-evaluated rather than inserting a literal TAB character. Select the operation required, the default is *Always Indent*.
- **Platform** – general platform specific settings.
 - **Termcap Color (Termcap Only)** enables use of colors.
 - **Use Fonts (Termcap Only)** enables use of **bold** and underline fonts.
 - **Font Name** selection of an alternative fixed font. For UNIX then use **xfontsel** to select a font to paste into the window. For MS-Windows then use the **Choose Font** button to select the font.
 - **Enable Tool Bar** enable if the tool bar at the left of the screen is required. Add tools to the tool bar, this provides short cuts, itemized lists for specific buffers. The tool bar is buffer sensitive.



- **Ignore files** the list of files to be ignored in directory listings. e.g. `~ .o CVS`
 - **Fence Display** defines the behavior of the cursor at fence characters (i.e. `(, {, },)`, `. .`). **Always draw and jump on close** is the most popular, the brackets are colorized and the cursor jumps to the opening fence on entering a close. A miss-matched fence is highlighted in an error color.
 - **Scroll Bars** generally **Wide with Splitter** is preferred in desktop environments. A wide scroll bar is two characters wide, the *splitter* is a control point to split the window with the mouse.
 - **Horizontal Scroll** defines the behavior of horizontal scrolling when the cursor is moved to the end of the line. By default only the current line is scrolled *Default, One line*.
 - **Vertical Scroll** defines the behavior of vertical scrolling when the cursor is moved down at the top/bottom line of the window. For terminals then **Default, half screen** is preferred, for graphical environments then **Smooth, Single line** is preferred.
 - **Color Scheme** the highlighting color of the buffer window. In a console environment then use the special scheme for *Termcap*.
- **Mouse** – behavior of the mouse.
 - **Mouse Button Bindings** control the action on pressing a button, optionally in conjunction with the `SHIFT`, `CONTROL` and `ALT` modifier keys. The wheel mouse setting is bound here.
 - **Miscellaneous** – configuration of extended tools.
 - **Session** configures how the user session is saved.
 - **Alias Path** define file short cuts of the form `~/name`, the short name is defined to be an alias of a longer file path.
 - **File Types** associate file extensions with a file type.
 - **File Tools** configure tools and automatic open actions used by the **file-browser**.
 - **Shell Tools** configure launching of external shell tools.
 - **E-Mail** configure MicroEmacs mailer.
 - **CVS** configure CVS. Note that CVS should not be configured in MicroEmacs if the CVS environment variables are defined in the working environment.
 - **ClearCase** configure MicroEmacs to work with IBM Rational ClearCase source control system.

1.6 Buffer Setup

The **User Setup** described in the previous section defines the global parameters and settings for the user session. User modifiable configuration parameters also exist on a buffer basis, settings such as *indentation* are modified, per buffer type, using the command:

esc-x buffer-setup ... OR

Tools→Buffer Setup

The settings for the buffer, Figure 2, are saved in the user session file (*user.esf*) and are retained when the buffer is re-loaded. The buffer setup dialog is tailored to a specific buffer type i.e. the Buffer Setup for the LaTeX programming language includes specific LaTeX options.

buffer-setup may be used to change the *major mode* or language template of the buffer (i.e. **C**, **C++**, **Java** etc.). The buffer mode may also be assigned quickly from the command line using **M-x buffer-major-mode**.

The following notable fields exist in the Buffer Setup dialog:

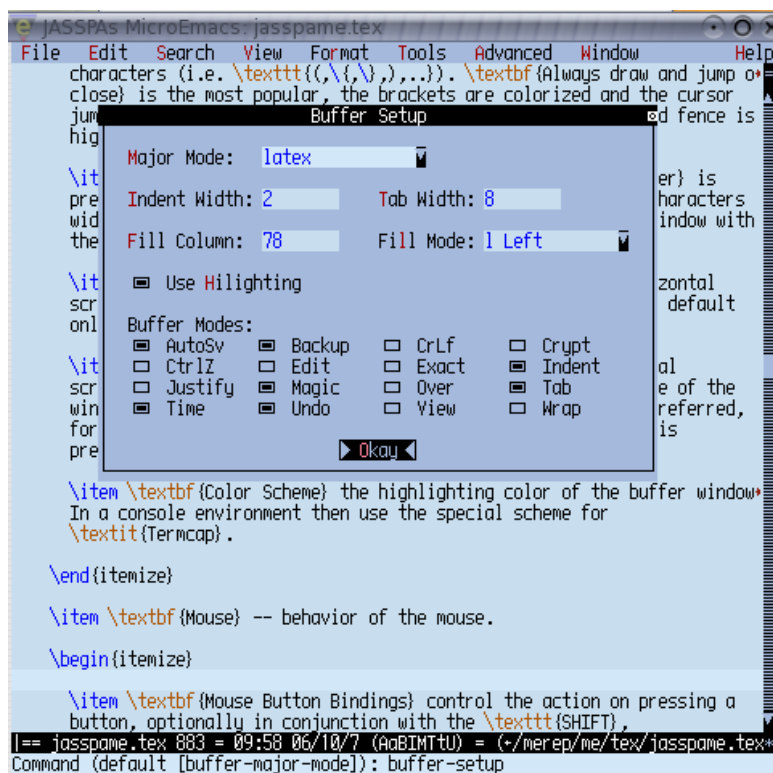


Figure 2: Buffer Setup Dialog

- Major Mode, selects the mode that MicroEmacs should adopt for the buffer.
- Indent Width, the indentation width for the buffer, the default is 4 (controlled by the variable *\$buffer-indent-width*).
- Tab Width, the width of a rendered TAB character, the default is 8 characters.
- Fill Column, the right-hand column where text should be wrapped, used for comments and text modes.
- Fill Mode, the mode which should be used to wrap text.
- Buffer Modes, the general modes of the buffer.

Note: the *indentation* and *tab width* are temporarily controlled for the buffer only. This is useful when dealing with code from a different source where the *indentation* and *Tab width* may be aligned to match the original authoring, the command **M-x tabs-to-spaces** may be used to expand the tabs back to spaces to make the file editor portable.

1.7 Major Modes

The **buffer-setup** effects local changes on the buffer, persistent changes to the buffer mode which are applied to all buffers of the same type are defined using **major-mode-setup**. The command presents a dialog that permits the attributes of the file type to be changed - this is known as the *Major Mode*.

esc-x major-mode-setup ... OR

Tools→Major Mode Setup

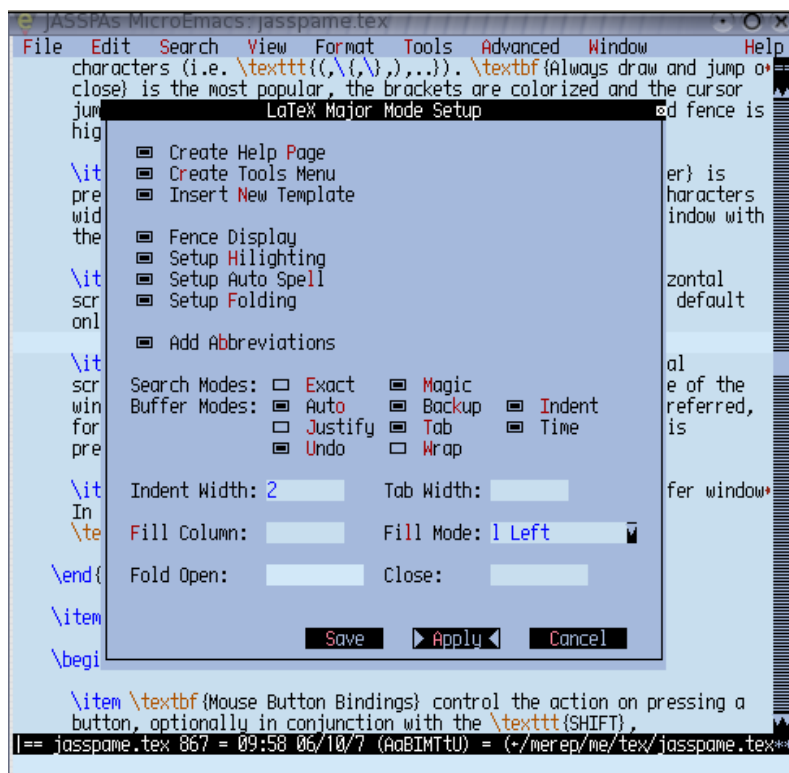


Figure 3: Major Mode Setup Dialog

This is similar to the **buffer-setup** but applies to all subsequent buffers that are loaded of the given type.

The settings for the Major Mode, Figure 3, are saved in the user session file (*user.esf*) and are retained through sessions. The major mode setup dialog is tailored to a specific major mode type i.e. the Major Mode Setup for the LaTeX programming language includes specific LaTeX options.

The following notable fields exist in the Major Mode Setup dialog:

- *Miscellaneous* which are Major mode specific.
- Indent Width, the indentation width for the buffer, the default is 4 (controlled by the variable *\$buffer-indent-width*).
- Tab Width, the width of a rendered TAB character, the default is 8 characters.
- Fill Column, the right-hand column where text should be wrapped, used for comments and text modes.
- Fill Mode, the mode which should be used to wrap text.

The configuration may be saved permanently using **Save** and installed into the current session using **Apply**.

1.8 MicroEmacs Integration

This section discusses how MicroEmacs may be integrated with the desktop environment and other tools.



1.8.1 Desktop Integration

Integration of MicroEmacs with the Window Manager desktop environment.

- Drag and Drop is supported. When a file(s) is dragged over a buffer window and released then the file(s) is opened in the drop window.
UNIX environments support *Xdnd*. Under X-Windows the MicroEmacs window is not raised following the drop.
Microsoft Windows the MicroEmacs window is raised following a drop operation.
- Different size bitmaps are provided for the desktop.
UNIX icons are provided in `/path-to/jasspa/pixmaps` in both **png** and **pixmap** format. A set of additional file type MicroEmacs icons may be downloaded for use with Gnome, KDE and CDE.
Microsoft Windows different size bitmaps are provided in the icon executable file **meicons.exe**. This executable contains the standard JASSPA icon in addition to the file type icons. Select icons from this executable when assigning MicroEmacs to specific file extensions.
- When assigning MicroEmacs to a desktop icon then it is advisable to start MicroEmacs with the command line option **-c**. This option restores the editor to the previously loaded state of the last saved session.
- Where MicroEmacs is assigned as an open action for a given file type or extension then an existing editor session may be used rather than opening a new editor. To enable MicroEmacs to reuse an existing session then invoke **me** with the **-o** command line option and enable the **Client-Server** configuration option (**Help** → **User Setup** → **Platform** → **Client Server**).
- The size and the position of the MicroEmacs frame position may be defined at startup. There are a number of different methods to set the frame characteristics as follows:-

- **Frame Size** the simplest method to fix the frame size at start up is to resize the frame to the desired size. Once resized, with the mouse *Right Click* on the *mode line* and select **Store Frame Size**. The current frame size is saved and restored the next time the editor is started.
- **UNIX – Size and Position** may be controlled using the **.Xdefaults** file, an entry in the file

```
MicroEmacs.geometry: 102x65 60 30
```

defines the frame size to be 102 characters wide, 65 characters deep and should be positioned 60 pixels from the left screen edge and 30 pixels from the top screen edge.

- **Windows – Size and Position** may be controlled using the **me32.ini** file, an entry in the file

```
[Defaults]
; Define startup screen size.
geometry=102x65+60+30
```

defines the frame size to be 102 characters wide, 65 characters deep and should be positioned 60 pixels from the left screen edge and 30 pixels from the top screen edge. This entry is in **[Defaults]** and therefore applies to all users, for a single user configuration then use the section labeled **[logname]** where *logname* is the users Windows login name.



1.8.2 Application Integration

MicroEmacs may be invoked from other tools as the selected text editor (i.e. **WinZip**). Multiple files may be specified on the command line for opening. Where a specific line number is required then the command line option **-l lineNo** may be used to open the next file and move to the given line number. The **-l** must always precede the file name, but may appear multiple times in the command line before each file to be open. e.g.

```
me -l 20 foo.c -l 902 bar.c
```

which opens file `foo.c` displaying line 20 and opens `bar.c` displaying line 902.

1.8.3 Utilities Integration

MicroEmacs optionally uses the standard UNIX utilities such as **grep**, **find** and **diff** for searching for information from the file system when present. These tools are typically found as default on a UNIX system.

Where the tools are not on the execution path, or the tool default options are altered, then the tools should be declared to MicroEmacs in your `username.emf` file. Table 1 defines the variables which are associated with the utility execution.

Variable	Description
<code>%man-com</code>	man utility, find and display reference pages.
<code>%find-com</code>	find utility, find files.
<code>%grep-com</code>	grep utility, search a file for a pattern.
<code>%diff-com</code>	diff utility, compare two files.
<code>%gdiff-com</code>	diff utility, graphical compare of two files.
<code>%cvs-com</code>	cvs utility, Concurrent versions system.
<code>%compile-com</code>	make utility, build command line.

Table 1: External Command Variables

The command defaults are platform specific, but are typically defined as follows:

```
set-variable %man-com "man"  
set-variable %find-com "find"  
set-variable %grep-com "grep -n"  
set-variable %diff-com "diff -r -c"  
set-variable %gdiff-com "diff -w"  
set-variable %find-com "find"  
set-variable %cvs-com "cvs"  
set-variable %compile-com "make"
```

1.8.4 Microsoft Windows UNIX utilities

Microsoft Windows does not include the standard UNIX utilities by default, they may be downloaded from <http://unxutils.sourceforge.net/>. Alternatively **cygwin** binaries may be used if installed.

The tools should be declared to MicroEmacs in your `username.emf` file, typically located in directory `c:\Documents and Settings\username\Application Data\jasspa`.

With the **unxutils** standard installation then the tools are defined in `username.emf` as follows:



```
set-variable %grep-com "c:/unix/usr/local/wbin/grep.exe -n"  
set-variable %find-com "c:/unix/usr/local/wbin/find.exe"  
set-variable %diff-com "c:/unix/usr/local/wbin/diff.exe -c -d -b -r"  
set-variable %gdiff-com "c:/unix/usr/local/wbin/diff.exe -c -w"
```

The directory path and options should be modified to match *your* version of the tools.

1.8.5 Microsoft Windows and Cygwin

MicroEmacs integrates with **Cygwin** (www.cygwin.com) allowing a **Bash** shell to be opened up in a window using the command **M-x cygwin**. Additionally the MicroEmacs commands **M-x man** and **M-x info** are available to access the Cygwin Manual and Info pages, respectively.

If cygwin is installed in the default location `c:\cygwin` then MicroEmacs automatically detects the directory and enables support. If cygwin has been installed in a different location then this should be defined to MicroEmacs in *username.emf* using the variable `%cygwin-path` as follows, correcting for your installation location:

```
set-variable %cygwin-path "c:/cygwin"
```

2 MicroEmacs Basics

In this chapter we introduce the basic information that you need to control MicroEmacs. MicroEmacs is equipped with a wealth of features and it is virtually impossible to describe each and every one in detail. For the new user then MicroEmacs is sometimes incorrectly perceived as difficult, there is a minimal set of commands which are required to drive the editor effectively, once mastered then you will find that MicroEmacs provides a fast and effective platform for handling text.

2.1 Getting Started

To start MicroEmacs then simply type the command **me** (or **me32** on MS-Windows) followed by the file-name(s) to edit e.g.

```
me myfile
```

The editor should start and load the file *myfile*. If the file does not exist no error is reported and the user is presented with an empty file, subsequently saving that buffer creates the file. If the file is a directory then a directory listing is presented.

If MicroEmacs is started with the command option **-c** then the last editing session is restored i.e.

```
me -c
```

additional files may be specified with **-c** if required.

2.2 MicroEmacs Screen

On starting MicroEmacs then the screen layout is as depicted in Figure 4.

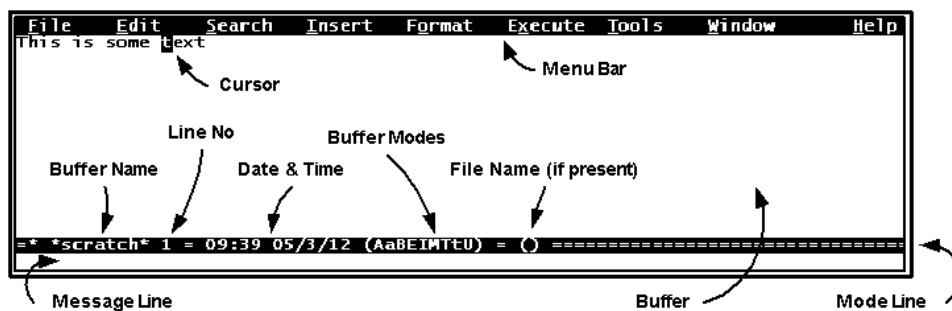


Figure 4: Basic Screen Layout

At the top of the screen is the **Menu Bar**, highlighted characters indicate the hot key that brings up the menu item using **ALT-key**. The menu may be accessed with the mouse or hot keys. If the Main Menu association has been disabled in *User Setup* then the **F1** key may be used to bring up the first menu item, thereafter the cursor and menu hot keys permit navigation.

At the bottom of the screen is the **Mode Line** which provides information on the buffer that is being edited, including the *buffer name*, *file name*, *line number* and abbreviated *buffer modes*. Where several buffers are on screen at the same time then each buffer has its own Mode Line.

The last line of the screen is the **Message Line** which is used to display messages and is used for command entry.



2.3 Files, Buffers and Windows

When editing a *file* then MicroEmacs copies that file into a *buffer* held in memory, the buffer has a name which is typically the base name of the file. The buffer is presented to the user in a *window*, the window typically has a scroll bar allowing the view into the buffer to be changed.

Changes to the buffer modify the in memory copy only, the file contents are not changed until the buffer is saved when it overwrites the original file with the new contents. A backup file(s) may be created with the previous file contents allowing the user to restore the previous version if necessary. During an editing session then a periodic automatic save is performed every few minutes to a recovery file, should the system unexpectedly crash then MicroEmacs recovers the automatic save the next time the file is loaded.

MicroEmacs allows multiple buffers to be loaded into the system at any time, not all buffers are necessarily visible, the user controls which buffers to display in a *window*. The windows may be split horizontally or vertically allowing the same, or different, buffer to be viewed in another window i.e. you are allowed to have two (or more) window views into the same buffer. Editing in one window immediately affects other windows that are viewing the same buffer.

Buffers may be created independently of a file, these are typically named with a preceding asterisk (*) character, i.e. **scratch** which always exists in the system and may be used for temporary working.

Buffers may be saved to files, when a buffer is saved to a different file then a new file is created or existing file is over-written. As a side effect the buffer name is changed to reflect the new file name.

2.4 Modes

MicroEmacs uses *modes* to customize the behavior of commands, a set of global modes exist which are inherited by all buffers that are created. Buffers themselves may change the mode to alter local buffer behavior.

The global modes are set up in *User Setup*, buffer modes are modified in the local buffer using:

- **C-x m** toggle buffer mode.
- Right mouse click in the buffer or on the mode line.
- **C-x major-mode-setup** to permanently redefine the modes associated with the buffer.

The common buffer modes are defined as follows:

- auto** – when enabled performs automatic source file line type detection (i.e. UNIX, DOS, Windows)
- autosv** – when enabled performs periodic auto-save backup of changes.
- exact** – when enabled makes searching and replace case sensitive.
- fence** – when enabled performs automatic fence matching i.e. (..) pairs.
- over** – when enabled overwrites rather than inserts entered characters, **insert** key.
- indent** – when enabled a new line of text automatically inherits the previous lines left indent.
- magic** – when enabled assumes a regular expression search and replace.
- quiet** – when enabled disables audio beep on an error.
- tab** – when enabled spaces are used instead of literal TAB characters.
- time** – when enabled files are automatically time stamped with the edit time.
- undo** – when enabled retains undo information.
- view** – when enabled the buffer is read only and cannot be modified.
- wrap** – when enabled automatic text wrapping is performed at the *fill column*.



2.5 Keys and the Keyboard

MicroEmacs is keyboard biased, this is the fastest way to control the editor. Throughout this manual we discuss commands and the keys on the keyboard that invoke those commands. MicroEmacs uses a special key sequence nomenclature to describe the commands.

C- the control key and another key pressed at the same time. **C-x** is the CONTROL and X key pressed at the same time, this sequence is a prefix key sequence that starts many MicroEmacs commands.

M- this is a special key called META which is used to start many commands. The META key is typically ESC but may also be the ALT key pressed with another key. To enable the use of ALT then this should be enabled in *User Setup*. **M-a** indicates the ESC key followed by the A key or ALT and A, this may also be written as **esc a**.

S- the SHIFT key. **S-pagedown** is the SHIFT key and PageDown keys pressed at the same time.

insert a special named key found on the keyboard i.e. INSERT, RETURN, BACKSPACE. The special key names are always lowercase.

A sequence of multiple keys is described with a minus sign (– to indicate that the keys are pressed together (a slight exception to this rule is **M-a** when ESC is used as ESC is pressed first followed by the A key). Where keys follow each other then a SPACE character is used. e.g.

C-x C-c – CONTROL-X followed by CONTROL-C.

C-x a – CONTROL-X followed by A.

M-space – ALT and SPACE or ESC followed by SPACE.

S-f1 – SHIFT and F1 function key.

2.6 Entering Text

Entering text in MicroEmacs is easy, simply type in text and RETURN to start a new line. MicroEmacs does not wrap text by default and scrolls the line to keep it within the window as more text is typed. MicroEmacs automatically wraps the text if **wrap** mode is enabled.

You can change the wrap mode setting using the following commands:

C-x m wrap.

Right Mouse on mode line and select **Buffer Modes**.

C-x buffer-mode

C-x buffer-setup C-x major-mode-setup

For most text buffer types then **wrap** mode is always enabled, for programming languages then it is typically disabled. Within a text type buffer then a paragraph is denoted by a empty line, paragraphs may be laid out using **M-o fill-paragraph** which re-styles the paragraph.

2.7 Cursor Movement

Navigation around the buffer is performed with cursor keys or with the following control sequences:

C-f *forward-char* – Forward character.

C-b *backward-char* – Backward character.

C-n *forward-line* – Next line.

C-p *backward-line* – Previous line.



C-v, page-down *scroll-down* – Next page.
C-z, page-up *scroll-up* – Previous page.

To move to the extremities of the line or buffer:

C-a *beginning-of-line* – Goto start of line.
C-e *end-of-line* – Goto end of line.
M-<, home *beginning-of-buffer* – Goto start of buffer.
M->, end *end-of-buffer* – Goto end of buffer.

To move between words and paragraphs then the keys are similar to the basic cursor movement but are modified by `META`.

M-f, C-right *forward-word* – Goto the next word.
M-b, C-left *backward-word* – Goto the previous word.
M-n, M-] *forward-paragraph* – Goto next paragraph.
M-p, M-[*backward-paragraph* – Goto previous paragraph.

Finally, to move to a specific line:

M-g *goto-line* – Prompt for a line number and jump to it.

2.8 Abort and Exiting

The most important command is **C-g** *abort*. If you are in the middle of executing a command or are in a unrecognized state then **C-g** aborts the current command and restore the user to a sane state.

To exit MicroEmacs:

M-z *quick-exit* – Saved unsaved buffers and exit.
C-x C-c *save-buffer-exit-emacs* – Prompts user to save unsaved buffers and exits.

2.9 Loading and Saving Files

When the editing is complete then the file may be saved using the command **C-x C-s** *save-buffer*. If there is no name associated with a buffer then no save is possible and the buffer should be written to a different file. To save a buffer to a different file then the command **C-x C-w** *write-buffer* is used. The user is prompted for the name of the file to save.

To load a file into a new buffer then use the command **C-x C-f** *find-file*. Control is moved to the *Message Line* for file entry. Within the Message Line then use the `TAB` key (or `esc esc`) to auto-complete, a second `TAB` displays a list of possible completions. Enter a few characters and use the `TAB` to complete again. Once the filename has completed then use `RETURN` to perform the read.

Whilst in the Message Line performing a **C-x C-f** then if `RETURN` is entered at a directory then a directory listing is displayed. The directory may be traversed by moving the cursor to the filename and pressing `RETURN` or by selecting a directory name with the mouse. If the selected item in the directory listing is a file then the file is loaded into a buffer.

A summary of the buffer commands:



- C-x C-f** *find-file* – Loads a file into a buffer.
- C-x C-s** *save-buffer* – Save buffer back to the file.
- C-x C-w** *write-buffer* – Save buffer to a new file.
- C-x C-i** *insert-file* – Insert a file at the current cursor position.
- C-x C-v** *view-file* – Loads a file into a buffer read-only (i.e. cannot be changed).
- C-x k** *kill-buffer* – Deletes the current buffer.

2.10 Insertion and Deletion

By default MicroEmacs operates in **insert** mode, the `insert` key toggles *insertion* and *overwrite*.

2.10.1 Region Copy/Delete

Text may be deleted using the following keys:

- backspace** *backward-delete-char* – Delete the previous character behind the cursor.
- delete** *forward-delete-char* – Delete the character under the cursor.

Sections of text may be copied (*yanked*) or deleted after they have been marked up. MicroEmacs uses a *Mark* position **C-space** *set-mark* to define the start of the block and *Point*, which is the current cursor position, to identify a *region*.

To mark up a region then use one of the following methods:-

- **Using the mouse**
 - Use the mouse to place the cursor at the start of the region.
 - Press the left key, keeping it pressed move the cursor to the end of the region.
- **Using the keyboard**
 - Use the cursor keys to move to the start of the region.
 - Place a mark using **C-space** *set-mark*.
 - Use the cursor keys to move to the end of the region.
- **Using the Keyboard – Microsoft Shift Region method** this is only if Microsoft Shift region is enabled in *User Setup*.
 - Use the cursor keys to move to the start of the region.
 - Hold down the shift key whilst moving the cursor keys, a region is defined.

Once the region is defined then the region may be copied to the *yank buffer* or deleted:

- M-w** *copy-region* – Copy the selected region.
- C-w** *kill-region* – Delete the selected region.
- M-x** *copy-rectangle* – Copies a rectangular section (or vertical columns) at the current position, the left and right limits are defined by the cursor keys.



2.10.2 Yank (or paste)

A copied or deleted region may be inserted back into the text by moving the cursor to the insertion position and yanking (*paste*) the text back using **C-y** *yank*. **yank** is typically bound to the middle mouse button, inserting text under the mouse pointer position.

The yank buffer has 20 entries, if a **M-y** *reyank* operation is performed immediately after a *yank* operation then the previous yank buffer contents are used. This allows previous text that has been copied or killed to be restored. The yank buffer remains valid when moving between different buffers allowing text to be copied from one buffer to another.

- C-y** *yank* – Restore a copied or deleted text region.
- M-w** *reyank* – Restore the previous copied or deleted text region.
- M-x** *yank-rectangle* – Copies a rectangular section at the current position.

2.10.3 Undo

MicroEmacs by default maintains an *undo* history allowing recovery from editing mistakes. *undo* is by default enabled and is changed by the **undo** mode. The undo information remains valid until a save operation is performed when it is discarded. The undo may be retained between saves by enabling the user option *User Setup::General::Keep Undo*.

- C-_** *undo* – Undo the last edit operation.

2.11 Buffers and Windows

Experienced users of MicroEmacs typically use more than one buffer (sometimes hundreds when running a long session) and continually move between different buffers. Typically only two or three *windows* are used (depending on screen estate), a working window and other window.

2.11.1 Buffers

The buffer commands are:

- C-x C-b** *list-buffers* – List the buffers in the system.
- C-x b** *find-buffer* – Find the named buffer (TAB to complete on the Message Line).
- C-x x** *next-buffer* – Move to the next buffer.

If you are not familiar with the buffer commands then the simple way to change buffer is to use the mouse. Right click in the buffer window and select the **Buffer..** item and a list of current buffers is displayed. Left click to select the required buffer.

On a console then the simple way to change the buffer is to use **C-x C-b** *list-buffers* change windows and move to the required buffer, enter 1 and the buffer is opened.

2.11.2 Windows

The *window* provides a view into a *buffer*, MicroEmacs supports up to 64 windows on screen at once. The windows may be created from the keyboard or mouse. Refer to the notes in a later section on using the mouse in the windowing system.

When buffers are swapped in and out of the windows then the buffer position is retained so that when the



buffer is restored to the window then the cursor and line position are the same as when the buffer was previously in view.

The essential window commands are:

- C-x 2** *split-window-vertically* – Split window into 2, *stacked*.
- C-x 3** *split-window-horizontally* – Split window into 2, *side-by-side*.
- C-x 1** *delete-other-windows* – Delete the other windows leaving current.
- C-x p** *previous-window* – Move to the previous window.
- C-x o** *next-window* – Move to the next window.

2.12 Searching and Replacing

Searching and replacement of text are frequently used in MicroEmacs, the search is probably the most important and runs *incrementally*. This is different from most other systems which require the user to enter the whole search string. The incremental search starts searching from the first character that is entered allowing the search to be refined as it continues through the buffer.

2.12.1 Searching

The search command is initiated using **C-s** *isearch-forward*, the user is placed on the Message Line. As characters are entered then the search proceeds through the buffer until a match is found and is displayed. If the user wishes to continue then refine the search by entering more characters or find the next match by entering another **C-s**. If the correct location in the file has been located then any other command quits the search leaving the cursor at the current position (e.g. **C-f** or a cursor key). If the end of the buffer is reached then the user is warned, a further **C-s** continues the search from the start of the buffer, effectively wrapping the buffer.

A search in a backward direction is initiated with **C-r** *isearch-backward* this operates in the same way as the forward search except the direction is changed.

The search is sensitive to the buffer modes:

exact when enabled performs a case sensitive search, the default is case insensitive.

magic when enabled performs a regular expression search. In **magic** mode then some characters are reserved as follows:

- . Wildcard character.
- ^ Start of the line.
- \\ Backslash character, \ is treated as an escape character and must be escaped for a literal backslash.
- \$ End of the line.
- [a-z] Range of characters, use a caret (^) at the start of range to negate the character set.
- \(pattern\|altpattern\) *pattern* group, optionally with *alternative patterns*.
- pattern** Zero or more occurrences of the previous *pattern*.
- pattern*+ One or more occurrences of the previous *pattern*.



2.12.2 Replacing

Replacing text is a powerful way to make global changes to a file and is invoked with **M-%** or **M-C-r** (**esc C-r**) *query-replace-string*. This is an interactive replace where the user controls each replacement, other replacement commands exist without the query.

On invoking the command then the user is prompted for a search string followed by a replacement string. MicroEmacs then commences the search and prompts the user for the required action when the search pattern is found. The user responses are defined as follows:

? provide help on the command response.

y perform the replacement.

n no skip the replacement and find the next.

! or **a** perform the rest of the replacements with no query.

u undo the last replacement.

C-g abort the replacement and leave the cursor at the current position.

. abort the replacement and leave the cursor at the point where the replace command was started.

As with the search, the replacement is affected by the buffer **exact** and **magic** modes. The regular expression search and replace allows patterns found in the search string to be copied to the replacement string, refer to the manual pages for more information.

2.13 Named Command Execution

Not all of MicroEmacs commands are bound to keys, for example the named command that has no key binding is **user-setup**. To execute a named command then we use **M-x *command* execute-named-command**. MicroEmacs prompts the user for the *command* to execute, at the prompt the **TAB** or **esc esc** keys attempt to complete the command, this is similar to the file name completion.

2.14 Universal Repeat

All of MicroEmacs commands may be executed *n* times using the **esc n *command* universal-argument**. Entering **esc** followed by an integer *n* executes the next command *n*-times. e.g.

```
esc 12 C-n
```

moves forward 12 lines (**C-n** is *forward-line*).

```
esc 100 C-y
```

inserts 100 copies of the yank buffer.

2.15 Mouse Interaction

The mouse interaction shown in Figure 5 depicts multiple buffer windows and a toolbar (leftmost Window). The mouse interaction is configured from *User Setup*

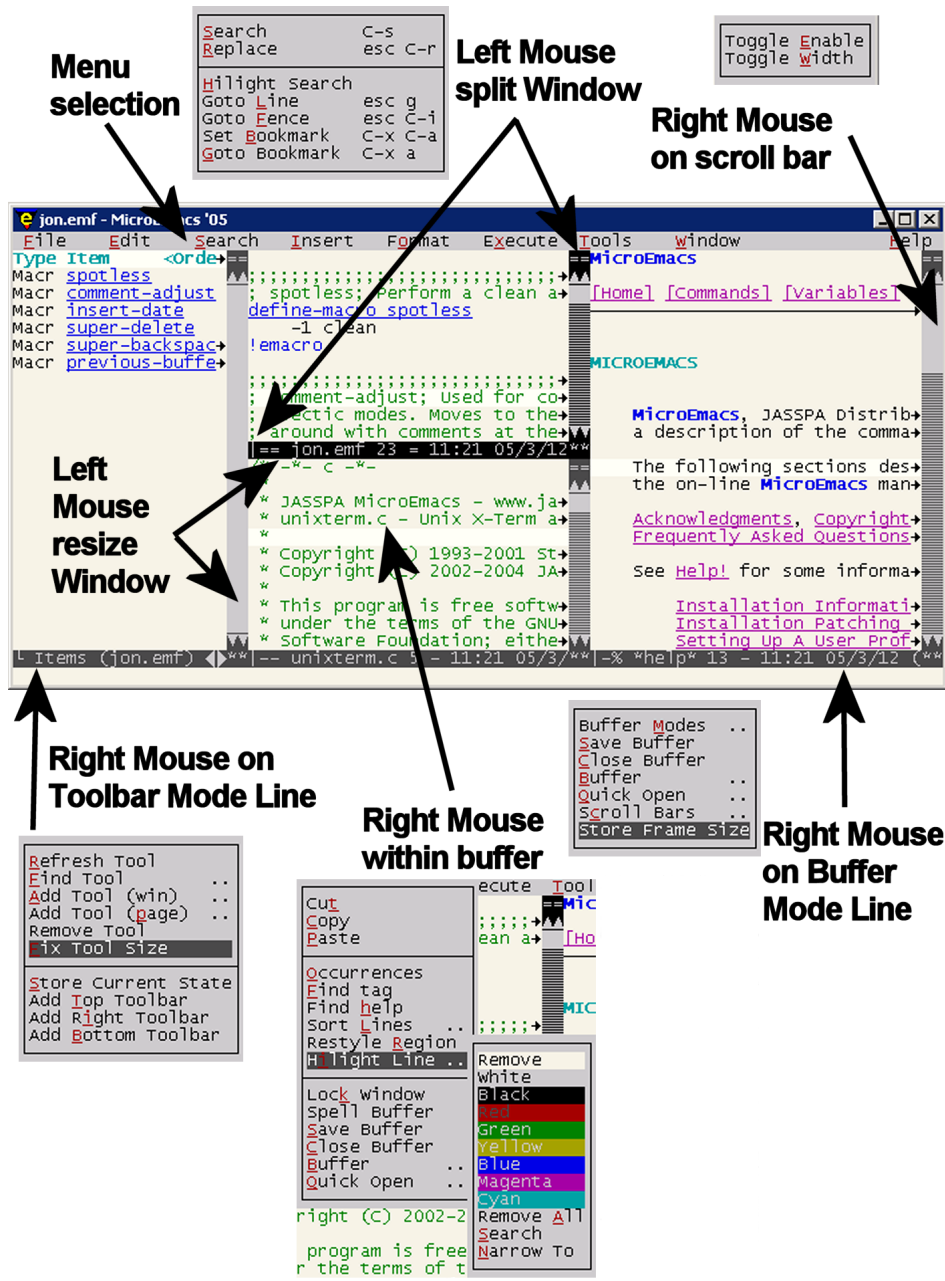


Figure 5: Basic mouse interaction

esc-x user-setup ... OR

Tools→User Setup

using the **Mouse** tab. The tab defines the behavior of the mouse, including the mouse wheel. The default mouse interaction is summarized in Table 16.

Screen Region	Mouse Operation	Description
Buffer	Left Mouse <i>drag motion</i>	Selects a region of text. The text is automatically added to the <i>yank buffer</i> and is available to external programs via the clipboard – no explicit <i>copy</i> operation is performed.

Table 16: Mouse controls (continued ...)



Screen Region	Mouse Operation	Description
Buffer	Middle Mouse	Yanks the last selected region of text from the <i>yank buffer</i> or <i>clipboard</i> and inserts at the current mouse position. The behavior and button binding may be changed in <i>User Setup::Mouse</i> . If the preferred behavior is for the text to be inserted at the cursor position rather than the mouse position then change the middle mouse button binding from Move to Yank to No Move Yank .
Buffer	Right Mouse	Pop-up context menu appears to interact with the buffer. The menu is context sensitive, if the right mouse is positioned over a spelling error when auto spell is enabled then alternative spelling corrections are displayed. The context menu may be invoked in a mouse-less system using M-x context-menu or via the short-cut key S-F1 .
Scroll Bar	Right Mouse	Pop-up menu to change the width or enable state of the scroll bar. The default scroll bar attributes are set in <i>User Setup</i> .
Scroll Bar	Left Mouse <i>drag motion</i>	Vertical motion scrolls the buffer, horizontal motion resizes the width of the buffer window. If a window width is reduced to zero then the window is deleted.
Scroll Bar	Left Mouse <i>splitter select</i>	If the window splitter option is enabled in <i>User Setup::Platform::Scroll bars</i> then selecting the splitter at the top of the scroll bar causes the window to be split into two.
Mode Line <i>Buffer</i>	Left Mouse <i>drag motion</i>	Vertical motion resizes the depth of the buffer window. If a buffer window depth is reduced to zero then the window is deleted.
Mode Line <i>Buffer</i>	Right Mouse	Pop-up menu to control the buffer window appears. The menu allows the buffer to be closed, a new buffer to be opened in the window, etc.
Mode Line	Left Mouse <i>splitter select</i>	If the window splitter option is enabled in <i>User Setup::Platform::Scroll bars</i> then selecting the splitter at the left of the mode bar causes the window to be split into two.
Mode Line <i>Toolbar</i>	Right Mouse	Only valid when the toolbar is enabled (<i>User Setup::Platform::Enable Toolbar</i>). A Pop-up menu appears to control the toolbar. The menu allows the toolbar content to be configured.

Table 16: Mouse controls