



EMACS Extensions

That you too might find useful

2010 Igor Lapitski, Hyde Park

Image from [Wikipedia](#)

This work is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or any later version. This work is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See version 3 of the GNU General Public License for more details.

“Emacs is not an editor. Emacs is a way of thinking about the world, and as such is a way of thinking about editors. When you ask what Emacs does, you are asking a question with no answer, because Emacs doesn't do, it is done to. Emacs just is ... I hope this makes things clearer.” ~

Scott Dorsey



Why is it called – Emacs?

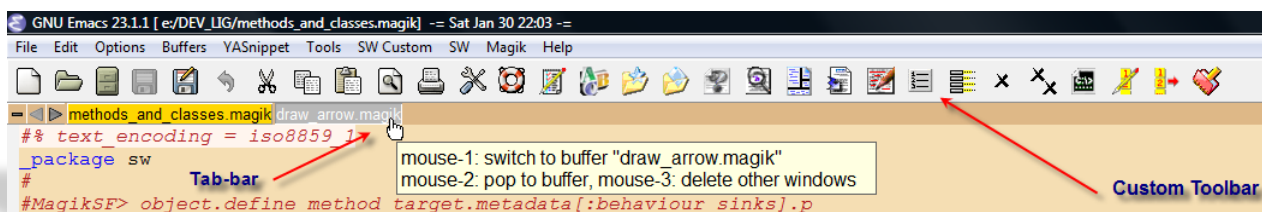
The name 'Emacs' was originally chosen as an abbreviation of Editor **MAC**roS and was created by Free Software Foundation.

This document includes highlights of some new and existing features and fine points in this customised version of Emacs.

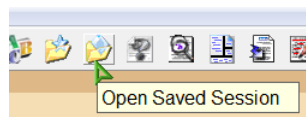
This customised version based on the Smallworld additions to Emacs issued with GE Smallworld 4.0.

All comments and suggestions please forward to <igor@hydepark-consulting.com> . You can download this Emacs from www.hydepark-consulting.com/Emacs .

- Custom **Toolbar** and **Tab-bar** (use <Ctrl-Tab> to navigate between tabs).



- **Save/Restore editing Sessions**, which allows you with one click of a button to load multiple files from a previous closed editing session. Even a cursor positions in the open files will be preserved from the previous session.



- **Dynamic Abbreviation** - it is standard function in the Emacs, but, from my experience, not everybody knows about. It is allowing you to write just a few characters of words you've written earlier to be able to expand them. Possible completions searched in the text in the current buffer, and if not found there, in all open buffers. To expand a word, just put the point right after the word and press **<Alt-/>**. As soon as you start typing – you are able to use this feature. Typing a next letter will automatically narrow your possible completions.

```
# Push it out to the owning frame
# respond to.
_self.framework.databus.make_data

_self.app
endmethod
$
```

```
# Push it out to the owning frame
# respond to.
_self.framework.databus.make_data

_self.application
endmethod
$ completed
```

- **Dynamic Abbreviation with Expansion of Multiple choices** – press the same key second time and you will get a multi-choice candidates in a tooltip.

```
# Push it out to the owning frame
# respond to.
_self.framework.databus.make_data

_self.app
endmethod
$
```

(a): app_dev_examples
(s): appear
(d): applies

- **Predictive Abbreviation**. It's fairly similar to Dynamic Abbreviation expansion, which works based on the contents of the current buffer (or other buffers). Predictive abbreviation expansion works based on the previously written text. Unlike Dynamic Abbreviation, the text is analysed during idle time, while Emacs is doing nothing else. As you type you will be offered possible expansion for the word at the left from the cursor position. Typing new characters will automatically offer different completion. If the correct completion is offered – just press **<Tab>** to complete the word. Light blue colour indicates that there are possible different candidates for the expansion. To see other candidates – press **<Tab>** once again and choice of possible completions will be shown. If the completion is shown in the dark blue colour – that's mean this is the only possible candidate.

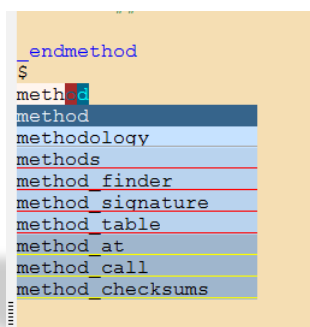
```
# Determine which data
# should do with the data
_self.data
```

```
# Determine which data
# should do with the data
_self.databus
```

```
# Determine which data types we should do with the data that is
_self.databusdatabus 1
data_type_name 2
database 3
```

```
# Determine which data
# should do with the data
_self.data_type_name
```

- **Auto Complete** - provides a way to complete any strings you expected by a graphical pull down menu. It looks like what Visual Studio provides as IntelliSense, Eclipse provides as CodeAssist. It is not configured to start automatically, you will need to use **<Alt-[>** to get Auto Complete menu and **<Alt-]>** to close it if you not finding suitable candidate. But it is possible to configure that Auto Complete will start automatically after you typed few characters. If this is enabled – then you will see the list of possible completions as you type. Use **<Tab>** to navigate through the list.



Auto Complete unlike previous expansion tools

working

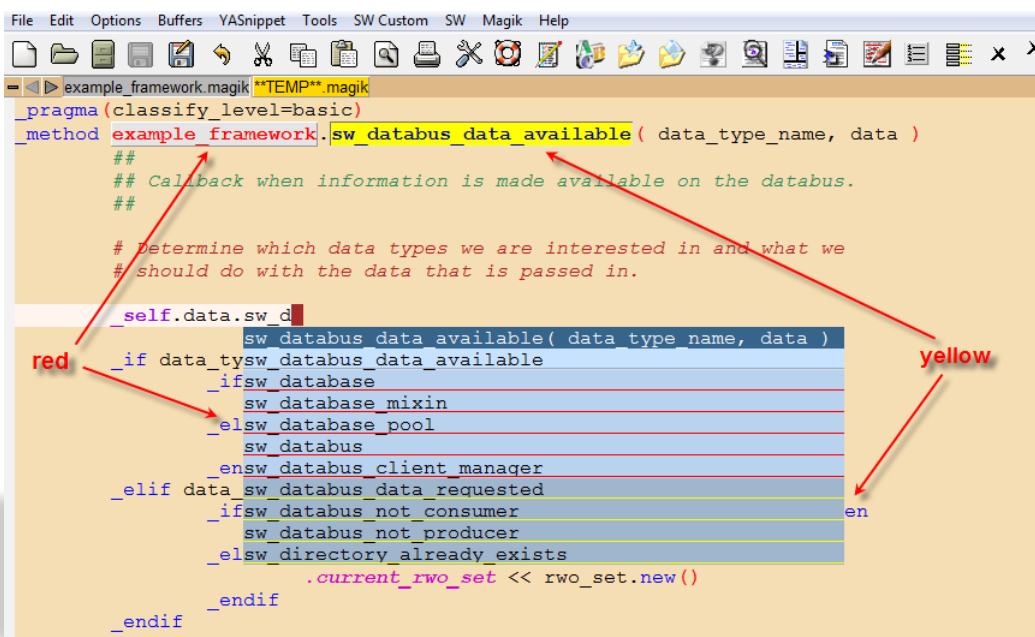
with the

Method Finder. All

possible candidates are

shown from the **Classes and Methods available in the current application image**. Candidates are shown in accordance with the code font locking – Classes fetched by the Method Finder underlined in **red**, Methods found by the Method Finder underlined in **yellow**, words found in open buffers shown without underlining.

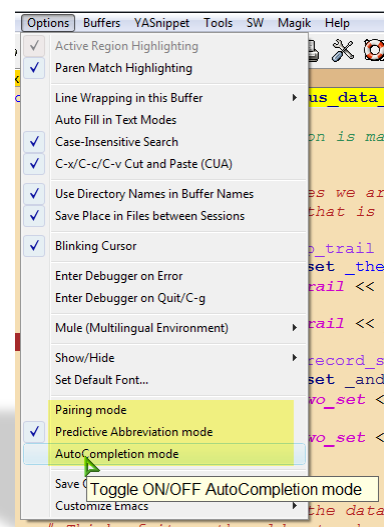
I am planning to place ON/OFF toggles on the Toolbar for Auto Complete and Predictive Abbreviation modes in order for users to decide which mode is preferable.



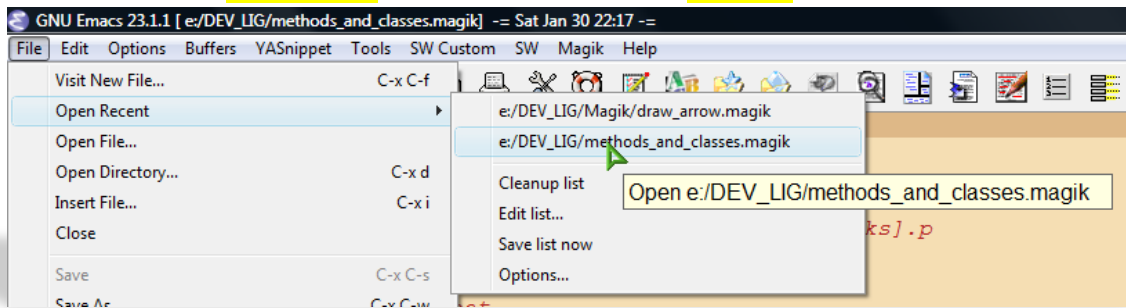
- **Automatic insertion of corresponding { and }**. Thus typing '(' will normally insert ')' and put cursor between them. Otherwise, when the **region** is active, it will be wrapped in the parentheses.

- Use **Options menu** to activate/deactivate those modes.

- **Highlight text matching PATTERN**. Examples: In a source code highlight a variable to quickly see all places it is modified or referenced. New occurrences of that variable will be highlighted as they are typed. In a *gis* or other buffer that is showing lots of output, highlight the parts of the output you're interested in. Those patterns will be highlighted as soon as they appear in the buffer. Useful if you have a lot of output during debugging. This option is located on the **Edit menu**.



- Additional menu item for **visiting files** that were operated on **recently**



- **Snippets for Magik**. It allows you to type an abbreviation and automatically expand it into function templates (similar to electric-magik). Navigation between snippet fields – using **<Tab>**. Templates will expand automatically as you type them and they are available via menu too.

```

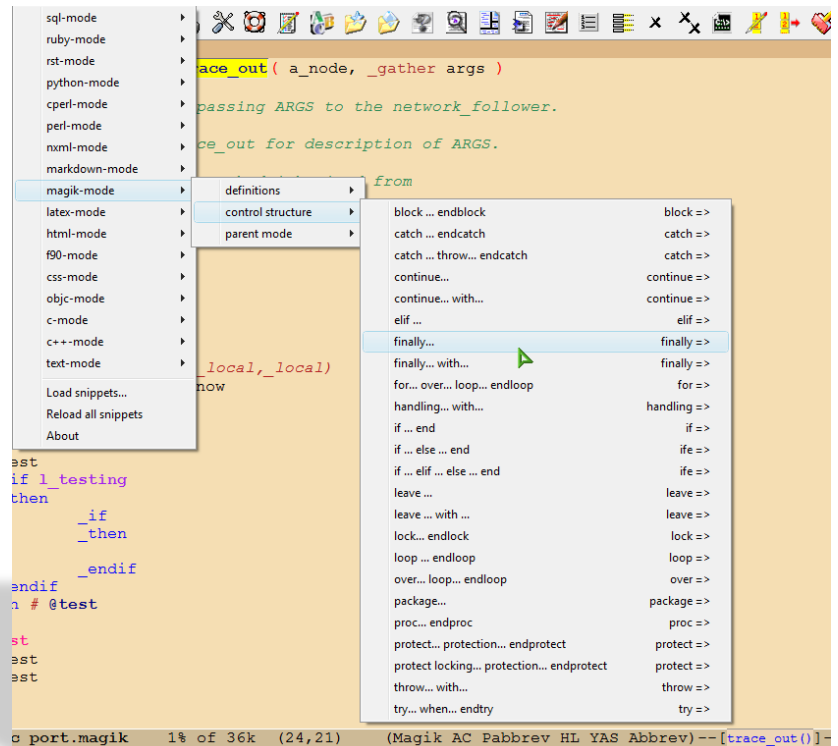
    _then      >> _true
    _else      >> _false
  _endif
_for <lvalue,tuple> _over <iter invocation>
_loop
_endloop


```

```

example_framework.magik **TEMP** magik
_method example_framework.example_test_method()
##
##
_for i_el_over self.fast_elements()
_loop
  _if i_el_is_unset
  _then
    leave
  _endif
_endloop
_endmethod
$

```

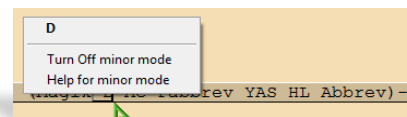


- **Dedicated mode** -  when a buffer is "dedicated" or locked, Emacs will not select files into that buffer. Dedicated buffers will have "D" shown in the mode line and marked lock icon. I found it very useful to lock the *gis* buffer. NOTE: All minor modes are shown in the mode line and you can use mouse to switch it OFF or get Help on the specific mode.

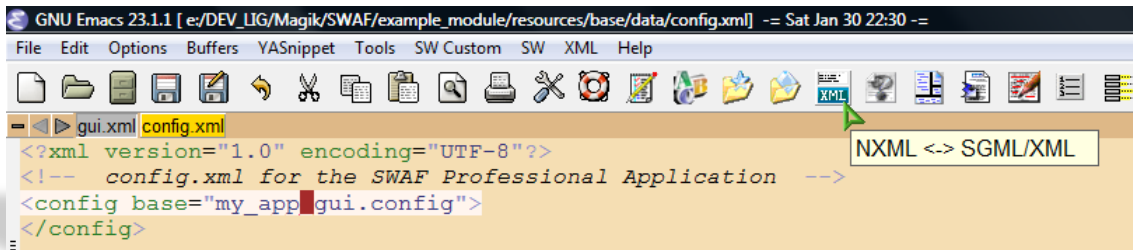
```

(Magik D AC Pabbrev YAS HL Abbrev) --[trace_out()]--

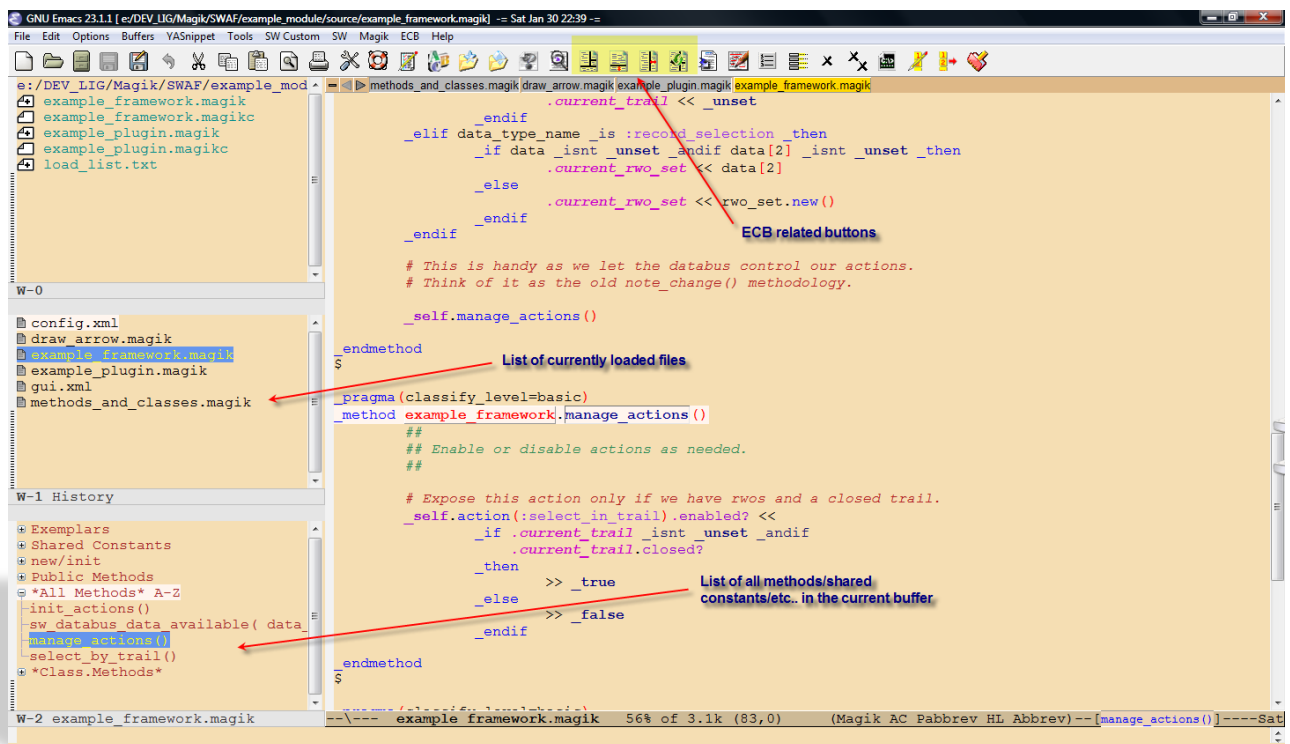
```



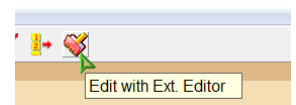
- NXML <-> SGML/XML mode switch



- Class Browser – F3-j / F3-b – allows you to go back in the File Navigation history, so you are able to return to the point in a code where you came from during your F3-j key journey.
- F4-m / F12-m – copy current magik method to the Work (**Temp**.*magik) buffer. Very helpful while making temporary changes in the code or just testing/debugging.
- F2-b – to compile message files <*.msg> directly from Emacs.
- ECB – Emacs Code Browser which displays a number of informational windows that allow for easy source code navigation and overview.

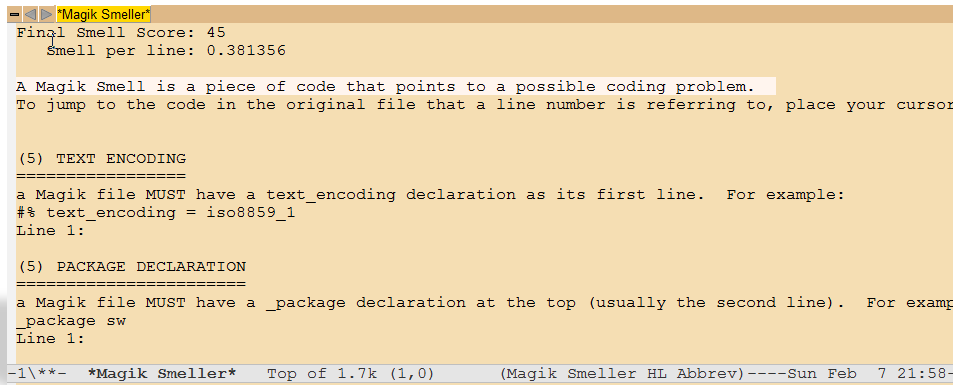


- Integration with an External Editor. This will allow you to load the file from the Emacs, if necessary, into an external editor of your choice to do some additional editing/searching or other file manipulations.



- **Magik Smeller**, which helps you to find possible problems in the code. A tool to interactively report actual or possible code syntax problems that are acceptable to the Magik compiler but do not meet development standards or may be considered inefficient.

A new buffer called ***Magik Smeller*** will appear showing a list of all possible code "smells", their line number in the Magik file and a short description. To jump back to a line of code that has been reported as a smell, place your cursor on that line in the ***Magik Smeller*** and press "F3-j".



```

-1\*- *Magik Smeller* Top of 1.7k (1,0) (Magik Smeller HL Abbrev)----Sun Feb 7 21:58-

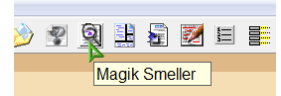
Final Smell Score: 45
Smell per line: 0.381356

A Magik Smell is a piece of code that points to a possible coding problem.
To jump to the code in the original file that a line number is referring to, place your cursor

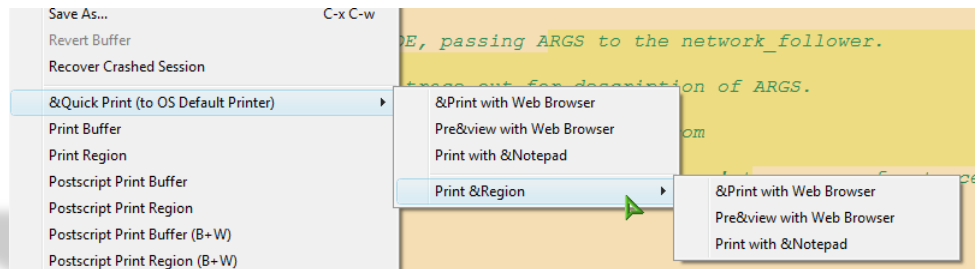
(5) TEXT ENCODING
=====
a Magik file MUST have a text_encoding declaration as its first line. For example:
%% text_encoding = iso8859_1
Line 1:

(5) PACKAGE DECLARATION
=====
a Magik file MUST have a _package declaration at the top (usually the second line). For example:
_package sw
Line 1:

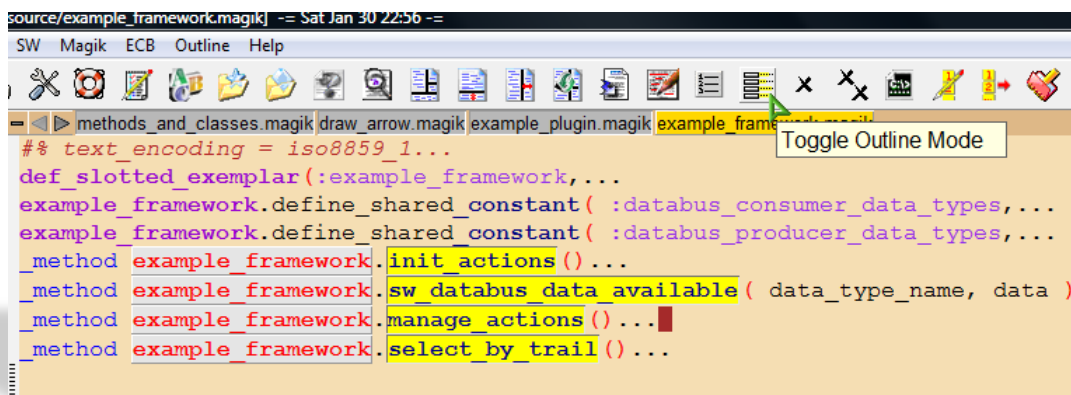
```



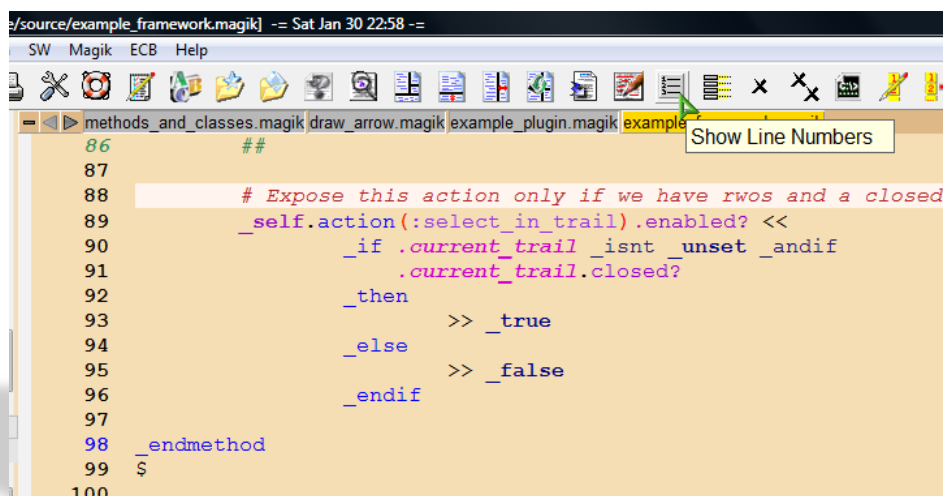
- **Print in colour or black&white** directly from the Emacs. Preview in colour – will create HTML page with current fontlocking highlights in the code.



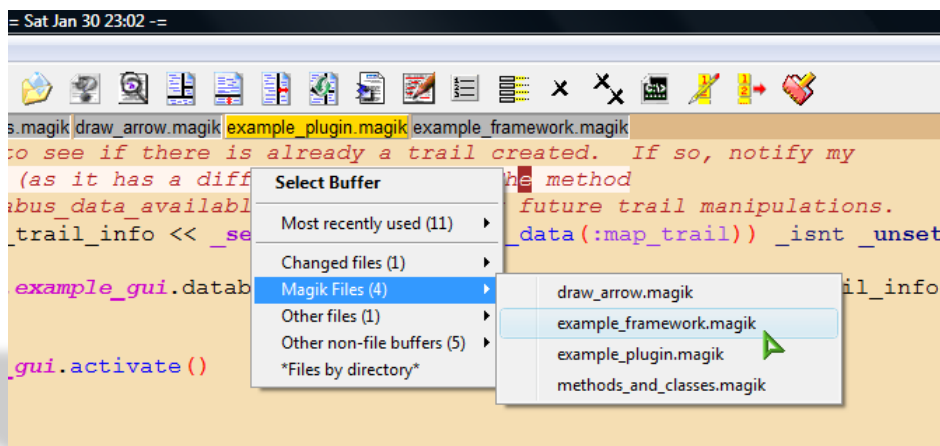
- **Outline mode** for Magik files. Using keys **<Alt>+Up/Down** in the Outline mode the Magik file can be collapsed to display only the method names and methods can be unfolded one by one for editing.



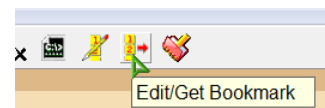
- Show/hide Line numbers in a buffer.



- Buffer selection- **<Ctrl>+<left mouse button>** will bring up the floating menu at the cursor position OR use **<Ctrl>+<Tab>** to navigate between tabs in the Tab-bar.



- Save all file **backups** (*.magik~) in the single configurable directory (system %TEMP% – by default).
- Use icons to work with **bookmarks** OR use 'breadcrumbs' **<Esc-Spc>** or **<Alt-Spc>** - a series of breadcrumb bookmarks with quick cycling though them using **<F5-b>**. Bookmarks are persistent across Emacs editing sessions.



- Method name in the mode line always visible

The screenshot shows an Emacs buffer named 'example_framework.magik'. The mode line at the bottom displays 'example_framework.magik 32% of 3.1k (65,22) (Magik AC Pabbrev HL Abbrev) [sw_databus_data_available()] ---Sat'. A red arrow points from the text 'Current method name' to the method name 'sw_databus_data_available()' in the mode line.

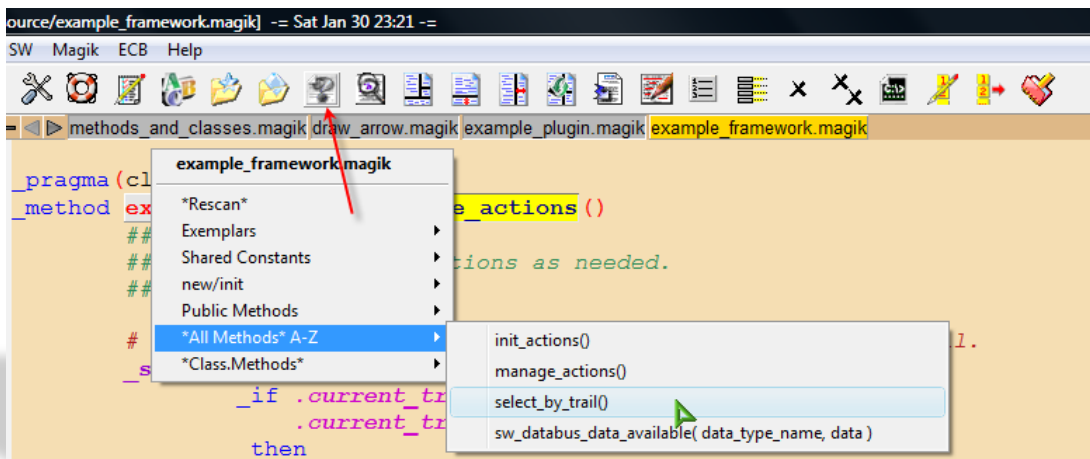
```
## Callback when information is made available on the databus.
##

# Determine which data types we are interested in and what we
# should do with the data that is passed in.

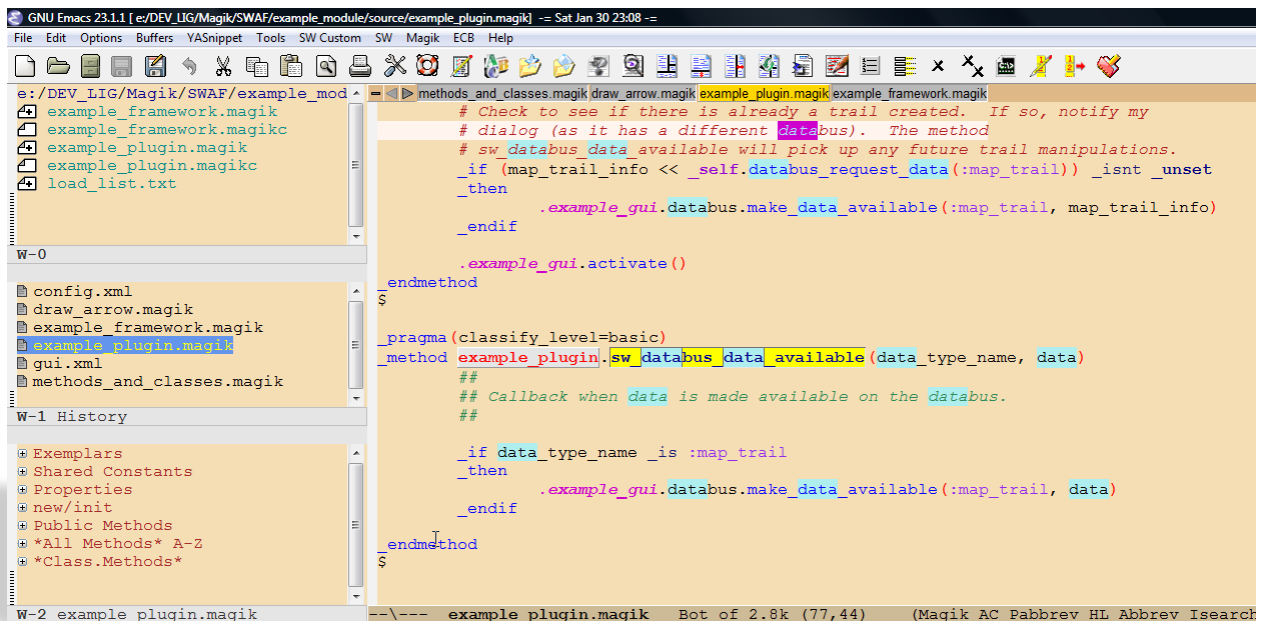
_if data_type_name_is :map_trail_then
  _if data_isnt_unset_then
    .current_trail << data[2]
  _else
    .current_trail << _unset
  _endif
_elif data_type_name_is :record_selection_then
  _if data_isnt_unset_andif data[2]_isnt_unset_then
    .current_rwo_set << data[2]
  _else
    .current_rwo_set << rwo_set.new()
  _endif
_endif

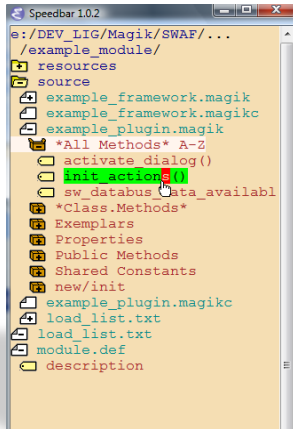
--\--- example_framework.magik 32% of 3.1k (65,22) (Magik AC Pabbrev HL Abbrev) [sw_databus_data_available()] ---Sat
```

- Show Methods menu allowing quickly navigate between methods.



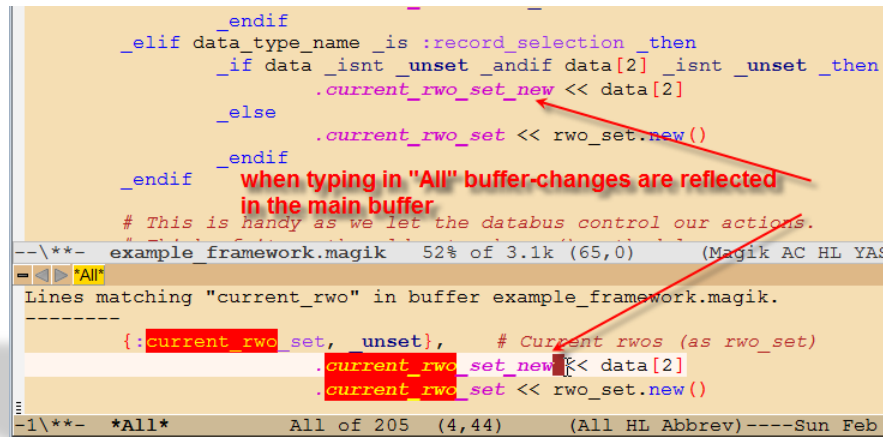
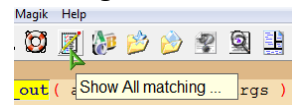
- Additional highlighting during the incremental search



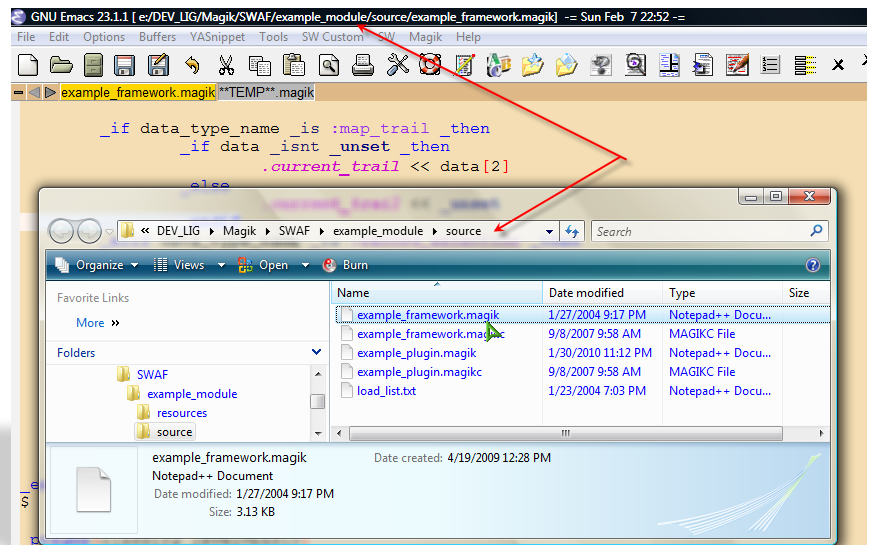


- **Speedbar (F11)** - a special skinny frame with a specialized directory listing in it. This listing will have both directories and filtered files in it. You can then load files into your Emacs buffer, or expand the files to display all the methods/progs/constants/.... in them and jump to those tags. You can also expand multiple directories into your Speedbar frame.

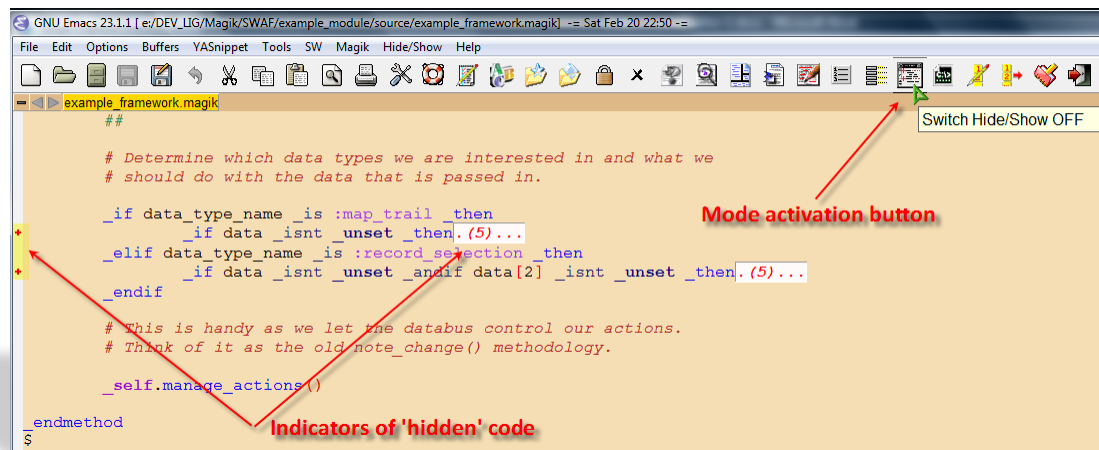
- Use '**Alt-x all <Enter>**' or the icon to **list\EDIT** all lines matching a given regexp in ***All*** buffer. ***Changes made in the *All* buffer is propagated to the original buffer.*** Use '**F3-j**' from ***All*** buffer to go to a match in the original file. **Note that changes to the ***All*** buffer are propagated back to the original buffer.**



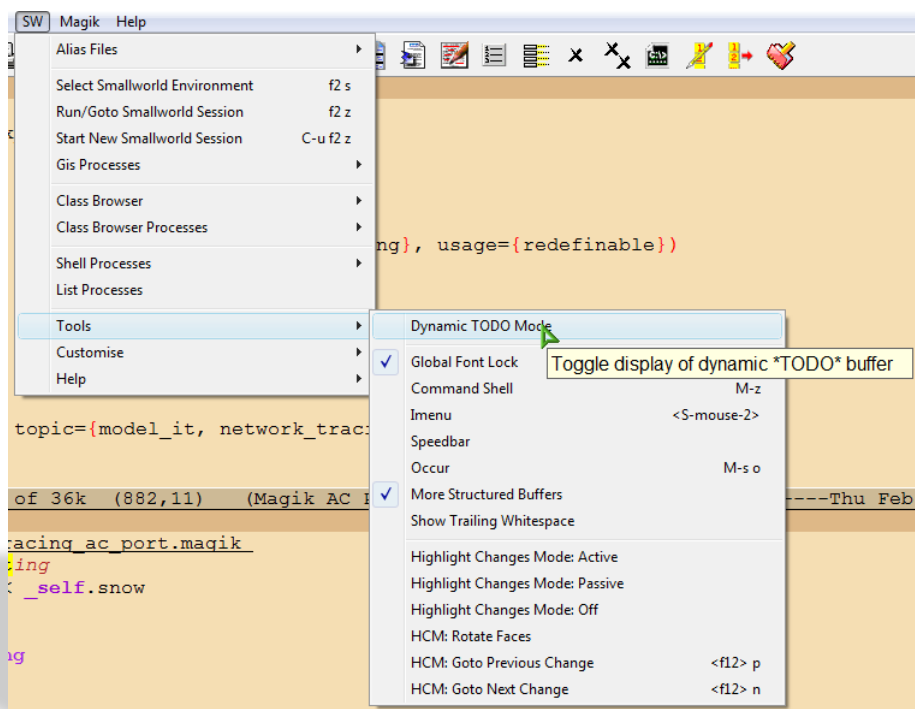
- While you have a file loaded in to the Emacs buffer pressing **<F12-F12>** will open Windows Explorer in the directory of the current file.



- **Magik code folding** – where large blocks of code can be distracting. With the code folding feature you can hide the code not important to your current work. Code folding cleans your work space. Use <Ctrl-.> key to hide the block of code while your cursor is inside the desired block.



- Use set of **very useful Tools** available on the SW menu.



- **Magik Language Reference** available within Emacs.

Up: (dir)
(MagikLangRef) Top

Magik Language Reference

The sections in Magik language reference duplicate the information in Using the Magik language, but the description here is more rigorous and assumes some experience with using the language.

- * see [Syntactic elements](#) summarises the simplest units of a program such as literals, keywords, operators and simple statements.
- * see [Expressions assignment and invocation](#) describes simple expressions and also the language constructions used for compound expressions. Assignment and routine invocations are included here as they are both discrete operations which can themselves be expressions.
- * see [Compound statements](#) lists all the compound statements of Smallworld Magik, including blocks, conditionals, loops, statements for transferring control and statements for handling unexpected events.
- * see [Defining procedures and methods](#) specifies the syntax for all possible ways of defining a procedure or method, and includes handling argument lists and results.
- * see [Variable declarations](#) gives the five types of variable declaration with a description of the differences between them, and also describes constant declarations, variable initialisation, use of packages and the resolution of free variables.
- * see [Special characters and keywords](#) lists all the special characters and keywords used in Magik, with references to where they are introduced.

* Menu:

- * [Syntactic elements](#)
- * [Expressions assignment and invocation](#)
- * [Compound statements](#)

--\%%- *info* (MagikLangRef) Top 3% of 1.8k (33,0) (Info HL YAS Abbrev Narro)

- Use **%USER_EMACS%** environment variable in gis_aliases to load your local Emacs settings, like
(setq programmer-tag "User Name, Company Name")
(setq project-name "Project Name")
(setq programmer-time-stamp-format "%:y-%02m-%02d"); %02H:%02M:%02S")

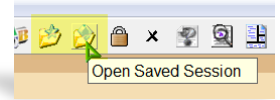
- **[Alt-n]** to insert-programmer-note;
[Alt-m] to insert-programmer-modified-note;

[Alt-c] to insert-change-note around selected block – just select the block of code and press <Alt-c> and note change comments will be placed around the selected code; OR just use this to create a note change.

```
# START CHANGE
# (2010-02-20 Igor Lapitski, Hyde Park) :
# Push it out to the owning framework databus so that the Application Framework can
# respond to.
_self.framework.databus.make_data_available(:geometry_to_highlight,
                                             geom_set)
# END CHANGE
```

[Alt-h] to insert-method-comment;

- **Save/Restore Emacs Sessions**. That allows you to organize your different projects into "sessions". Each Session has a name (file name). When you 'Save Session', the list of all opened files is saved into the current session and all the files you were editing will be reloaded the next time you 'Restore Session'. This allows you to keep different files open in different Emacs sessions. By using 'Save/Restore Session' you are activating 'Session auto-saving mechanism' - on Emacs exit current Session will be saved automatically.



- To make **prominent marks** in '*gis*' buffer or comments in 'Magik' buffer use '@' at the beginning of the line in '*gis*' buffer and '#@' in 'Magik' buffer. I found this useful when you have a lots of output in the *gis* buffer and you need to see specific line – just use this in your code *write("@Important info: ", a_variable)* and you will be able to see this line in the output very clearly.

```
# Determine which data types we are interested in
# should do with the data that is passed in.

#@ This is the comment that I want to see
#@ Review
_if data_type_name _is :map_trail _then
  _if data _isnt _unset _then
    .current_trail << data[2]
  _else
    .current_trail << _unset
```

- Use **F2-b** to send <*.msg> file to running GIS buffer. This will load current *.msg file into an open currently image.

- Emacs will display highlighting on whatever **parenthesis** matches the one before or after point. See message in Minibuffer describing the match when the matching parenthesis is off-screen. Message contains the linenumber.

```
...:enabled?, _false,
...:action_message, :select_by_trail|()|)
endmethod
--/*- example framework.magik 39% of 3.2k (47,72) (Magik ns+ AC Pabbrev
Matches self.add action(...) [-7-]
```

- Use **Ctrl-TAB** to switch\cycling between Emacs buffers.

- When using <Ctrl-x/Ctrl-b> to bring up Buffer Selection Menu – use following keys : <c> - to change the configuration; <s> - to save buffers marked with '*' ; <d> - to delete buffer

- List of key-bindings which might be useful in everyday Emacs operations with Smallworld environment:

| | | | |
|-------------|-------------------------------|-----------|--------------------------------|
| <f2> RET | magik-transmit-thing | <f3> / | cb-and-clear |
| <f2> SPC | explicit-electric-magik-space | <f3> ? | cb-help |
| <f2> # | magik-comment-region | <f3> b | cb-jump-back |
| <f2> \$ | magik-transmit-\$-chunk | <f3> c | cb-paste-class |
| <f2> X | dev-tools-object-inspector | <f3> d | toggle-transmit-magik-debug-p |
| <f2> b | magik-transmit-buffer | <f3> j | cb-jump-to-source |
| <f2> h | magik-heading | <f3> m | cb-paste-method |
| <f2> m | magik-transmit-method | <f3> <f3> | cb |
| <f2> q | fill-magik-public-comment | | |
| <f2> r | magik-transmit-region | <f4> | magik-f4-map |
| <f2> t | magik-trace-curr-statement | <f4> c | magik-copy-method |
| <f2> x | deep-print | <f4> d | dev-tools-vs-d-transmit-method |
| <f2> <down> | forward-method | <f4> e | magik-ediff-methods |

| | |
|---|--|
| <code><f2> <f7></code> magik-transmit-method <code><f2> <f8></code> magik-transmit-region <code><f2> <up></code> backward-method <code><f2> M-#</code> magik-uncomment-region <code><C-f2></code> magik-translate-old-vec-notation <code><f2> TAB</code> hippie-expand <code><f2> e</code> electric-magik-mode <code><f2> k</code> sw-reload-dotemacs <code><f2> s</code> gis-version-selection <code><f2> z</code> gis <code><f2> g</code> gis | <code><f4> m</code> magik-copy-method-to-buffer <code><f4> n</code> magik-set-work-buffer-name <code><f4> o</code> dev-tools-object-inspector <code><f4> r</code> magik-copy-region-to-buffer <code><f4> s</code> magik-add-debug-statement <code><f4> w</code> magik-compare-methods <code><f4> <f4></code> magik-symbol-complete <code><f4> b</code> bury-buffer <code><f4> g</code> goto-line <code><f6></code> magik-copy-method <code><f7></code> magik-transmit-method <code><f8></code> magik-transmit-region <code><f9></code> magik-mark-method |
| <code><M-Spc></code> set breadcrumb bookmark <code><M-f5></code> breadcrumb-list <code><f5> b</code> breadcrumb-next <code><f5> d</code> bookmark-delete <code><f5> e</code> edit-bookmarks <code><f5> j</code> edit-bookmarks <code><f5> l</code> bookmark-load <code><f5> m</code> bookmark-set <code><f5> s</code> bookmark-write <code><f12> x</code> switch-xml-mode <code><f12> <f12></code> w32shell-explorer-here <code><f12> m</code> magik-copy-method-to-buffer <code><f12> n</code> highlight-changes-next-change <code><f12> p</code> highlight-changes-previous-change | <code><f11></code> speedbar-get-focus <code><f12> -</code> ecb-cycle-maximized-ecb- buffers <code><f12> =</code> ecb-maximize-window-methods <code><f12> b</code> ecb-minor-mode <code><f12> c</code> ecb-toggle-compile-window <code><f12> e</code> send-file-to-external-editor <code><f12> h</code> ecb-toggle-compile-window- height <code><f12> q</code> ecb-redraw-layout <code><f12> r</code> ecb-restore-window-sizes <code><f12> t</code> ecb-toggle-layout <code><f12> u</code> ecb-rebuild-methods-buffer <code><f12> w</code> ecb-toggle-ecb-windows |
| <code><M-/></code> dabbrev-expand <code><M-d></code> close window <code><M-b></code> move cursor to another window <code><M-f></code> move cursor to another frame <code><M-m></code> insert-programmer-modified- note <code><M-n></code> insert-programmer-note | <code><M-S-down></code> buf-move-down <code><M-S-left></code> buf-move-left <code><M-S-right></code> buf-move-right <code><M-S-up></code> buf-move-up <code><M-c></code> insert-change-note <code><M-h></code> insert-method-comment <code><C-Tab></code> toggle between files |

- For customisation of the empty magik file header and method comment – make necessary changes in the `<~\site-lisp\custom-lisp\custom-magik-info.el>` file.

Many of those settings are possible to customise, use the menu **SW->Customose->Customise Smallworld Emacs**.

The background colour of the custom version of the Emacs (wheat) also possible to change. All custom settings will be saved in the **<.emacs-custom>** file in your %HOME% directory. See http://www.hydepark-consulting.com/Emacs/Emacs_colours.pdf for all possible colours.

```
GNU Emacs 23.1.1 - Mon Feb 15 00:07 -
File Edit Options Buffers YASnippet Tools SW Custom Help

"Customize Group: Smallworld" "Customize Group: Misc Custom"
Operate on all settings in this buffer that are not marked HIDDEN:
Set for current session Save for future sessions
Undo edits Reset to saved Erase customizations Exit

Parent groups: Emacs
Parent group documentation: Manual.

/- Smallworld group: Smallworld Development group-----\
State: visible group members are all at standard values.

Menu Sw Tools Submenu:
State: HIDDEN, invoke "Show" in the previous line to show.
List of Menu entries for SW->Tools submenu.

Module Option Save Magik: Toggle on (non-nil)
State: STANDARD.
If t, save .magikc files when loading module.

Module Option Force Reload: Toggle on (non-nil)
State: STANDARD.
If t, save .magikc files when loading module.

Resources : Group for Smallworld Emacs Message internationalisation.

Aliases : Customise Magik aliases files group.

Magik : Customise Magik Language group.

Class Browser : Running Magik Class Browser.

Gis : Running Smallworld GIS.

Misc Custom : LIG Custom Emacs User Setting

Product : Customise Magik product.def files group.

-1\*- *Customize Group: Smallworld* 13% of 1.8k (40,0) (Custom YAS H
```

```
GNU Emacs 23.1.1 - Mon Feb 15 00:09 -
File Edit Options Buffers YASnippet Tools SW Custom Help

"Customize Group: Smallworld" "Customize Group: Misc Custom"
ECB load on Emacs start up?

Pair Mode: Value Menu OFF
State: STANDARD.
Pairing mode on Emacs start up?

Toggle Pabbrev Mode: Value Menu ON
State: STANDARD.
Predictive Abbreviation on Emacs start up?

Toggle Ac Auto Start: Value Menu Require: 3
State: SAVED and set.
Non-nil means completion will be started automatically.

Initial Frame Width: 135
State: STANDARD.
WIDTH of the Initial Frame

Initial Frame Height: 43
State: STANDARD.
HEIGHT of the Initial Frame

Custom Background Color: wheat
State: STANDARD.
Emacs background color

Custom Modeline Background Color: wheat3
State: STANDARD.
Emacs modeline background color

Custom Foreground Color: black
State: STANDARD.
Emacs foreground color

Custom Backup Directory: C:\TEMP
State: STANDARD.

-1\*- *Customize Group: Misc Custom* 37% of 1.7k (44,45) (C'
```