How to add iMenu and Speedbar support for a major mode

Last updated on: 2025-03-25

Overview

See:
- **Menus**
- **Speedbar**
- **iMenu Support (in PEL Manual)**
- **Speedbar/iMenu Mode Compatibility**

The iMenu and Speedbar features provide the ability to list various items inside a file.

⚠️Emacs supports these features in several major-modes but not all.

This document describes how PEL provides iMenu and Speedbar support for major modes that does not already support them. It describes:
- The Emacs commands you can use to investigate the existing support.
- The Emacs Lisp code PEL uses to add support.

Speedbar uses information gathered by the iMenu system.
- To support Speedbar, a major mode must support iMenu and then identify the files that are processed by the Speedbar.

Investigation Commands

Use the following commands to check what is currently supported by the major mode of the current buffer.

Print imenu controlling variables See also: **Menus**	`<f11> ? e i`	(pel-imenu-print-vars)	Print the value of the imenu variables used to control the imenu functionality for the current buffer. • Print this information in a *imenu-dbg* buffer. • symbols are clickable buttons to help on the symbol. • Use this when investigating the imenu support for a major mode: use as a primitive Emacs development tool.

How to add iMenu support for a major mode

See:
- **Emacs Manual - iMenu**
- **Search/Replace** for Emacs Lisp regexp syntax.
- **Emacs AList (association lists)**

There are several ways to provide iMenu support for a given major mode.

Simplest Method

Create a set of regular expressions to detect various types of elements and add these in one of potentially many `(MENU-TITLE REGEXP INDEX)` list elements of the **imenu-generic-expression** variable. Where:
- `MENU-TITLE` describes the item. Use `nil` to put this entry at the top of the menu or when you only have one category of items.
- `REGEXP` is the regexp string used to identify the elements. These are match expressions with potentially several groups.
- `INDEX` is the index integer of the regular expression match, where 1 is for the first match group. Use 0 to include the complete expression.

For example, the following simplified code, executed once the mode starts, sets the syntax to extract and list Class, Function, Enum and Constant entries, extracting the name from regexp group 1 in each case. Speedbar and various listers will show the Class first, then Function, Enum and Constant entries.

```
(setq-local imenu-generic-expression '(
                                       ("Constant"
                                        " *constant +\\([[:alnum:]_]+\\)[ \n]*="
                                        1)
                                       ("Enum"
                                        " *enum +\\([[:alnum:]_]+\\)[ \n]*{"
                                        1)
                                       ("Function"
                                        "[[:alpha:]][[:alnum:]_]+ +\\([[:alpha:]][[:alnum:]_]+\\) *(.*)[ \n]+{"
                                        1)
                                       ("Class"
                                        " *class +\\([[:alnum:]_]+\\)[ \n]*{"
                                        1))))
```

Most Flexible Method
- Write a function that creates a menu-specific data structure similar in format to **imenu--index-alist** :
  - This is an Emacs Lisp **alist** that can hold the following element types:
    - *simple element*: `(INDEX-NAME . POSITION)` , where:
      - `INDEX-NAME` is a string describing the element type,
      - `POSITION` can be a character position integer, a marker, or an overlay.
        - Only one entry may use a negative value for `POSITION` to force a rescan of all entries.
    - *special element*: `(INDEX-NAME POSITION FUNCTION ARGUMENTS…)`, where
      - a menu entry means executing the following Emacs Lisp expression: `(FUNCTION INDEX-NAME POSITION ARGUMENTS…)`
    - *nested list element*: `(INDEX-NAME . SUB-ALIST)` , where `SUB-ALIST` is a nested alist. Use this for hierarchical menus.
- Set **imenu-create-index-function** to that function in a hook for your major mode.

How to add Speedbar support for a major mode

See also:
- **Auto Mode Alist @ Emacs Wiki**

Speedbar support requires iMenu support

To support Speedbar, the major-mode must have support for iMenu. If it does not have iMenu support, add it using the instructions above.

Identify the file extensions used by files using that major-mode

The speedbar system must be informed about the files it must check. You need to ensure that:
- There is an association between the file and the major mode: the file identification (regexp) must be part of the **auto-mode-alist**. If this association is not done by the major mode code, then it must be done in the initialization code.
- For the major modes supported by PEL if the file association is missing PEL adds them in the **pel_keys.el** file. That code is executed by when pel-init runs.
  - For example, the default Make file support does not support files with the .mak extension that is sometimes used. The following code is used by PEL to associate the .mak file extension to the makefile-nmake-mode:

```
(add-to-list 'auto-mode-alist '("\\.mak\\'" . makefile-nmake-mode))
```

- The speedbar system must also be informed about file extensions if it is not already informed.
  - This is done by calling the **speedbar-add-supported-extension** function and passing the file extension.
  - PEL provides a proxy function that also loads speedbar lazily: **pel-add-speedbar-extension** which is only called when the **pel-use-speedbar** user-option is turned on.
    - The following code snippets are taken from the pel_keys.el file and show various call scenarios:
      - A single file extension

```
(when pel-use-speedbar
  (pel-add-speedbar-extension ".go"))
```

      - Several file extensions in a list:

```
(when pel-use-speedbar
  (pel-add-speedbar-extension '(".y"
                                ".yacc"
                                ".lex"
                                ".l"
                                ".jison")))
```

      - Several file extensions identified using a regular expression:

```
(when pel-use-speedbar
  (pel-add-speedbar-extension ".li?sp")
```

iMenu — Reference

Topic/URL	Comment
Articles describing iMenu	• Exploiting Emacs Imenu's Potential • Emacs: Rapid Buffer Navigation with Imenu