🚦 Speedbar - Major Modes Compatibility 🚧

Language	PEL PDF Table	Emacs Support file	Feature name	Major Mode Name	Modeline Lighter	Description

Speedbar and SR-Speedbar
 	
▰iMenu/ Speedbar support
 	
Last updated on:
	The table describes the capabilities and compatibility of Speedbar and Speedbar-SR with various major modes.
	    It complements the information on Speedbar and Speedbar-SR available in the 𝚺 **Speedbar** table.
	 • The types shown in Speedbar correspond to the information extracted by imenu and therefore the table shows what imenu based commands can show.
	        • See also 𝚺 **Menus** and 𝚺 **Navigation** tables.
	 • This is an early version. More information will be included as more Emacs major modes are tested with Speedbar.
	 • Note that these correspond to test using PEL with the current packages used. You may get other results with other packages. If you do, please let me know.					

	2025-04-15					
Ada		**ada-mode.el**				
Assembler		**asm-mode.el**				
Arc	𝔓𝔩 **- Arc**					
Asciidoc	𝔐 **AsciiDoc**	**adoc-mode.el**	adoc-mode	adoc-mode	adoc	• .adoc files ❌ • Document sections ❌
AWK	𝔓𝔩 **- awk**					
Batch files		**bat-mode.el**				

C	𝔓𝔩 **- C**	**cc-mode.el**				• .c files: ✅
						    • function definitions ✅
						    • pre-processor macros ❌
						 • .h files: ✅
						    • inline function definitions ✅
						    • pre-processor macros ❌

C++	𝔓𝔩 **- C++**	C++				• .cpp files: ✅
						    • class definition ❌
						    • class member function definitions ✅ shows namespace,
						    • function definitions ✅
						    • template function instance definition ✅
						    • template function definitions ❌
						    • pre-processor macros ❌
						 • header files (.hpp): ✅
						    • class definition ❌
						    • class member function definitions ❌
						    • function definitions ❌
						    • template function definitions ❌
						    • pre-processor macros ❌

C#						
Ceylon						
Chapel						
CLU						
Clojure						
CMake						
Crystal						
Curl						
CWEB						
D	𝔓𝔩 **- D**	D				
Dart						
Eiffel						
Elixir	𝔓𝔩 **- Elixir**					
Elm						

Emacs Lisp	𝔁𝔓𝔩 **- Emacs Lisp**	**elisp-mode.el**				• .el files: ✅
						    • defun top-level & indented forms ✅
						    • defmacro top-level forms ✅ (Note: does not distinguish from defun)
						    • defsubst top-level forms
						    • defvar top-level and indented forms ✅
						    • defconst top-level forms
						    • defcustom top-level forms
						    • defgroup top-level forms
						    • defhydra indented ❌

Erlang	𝔓𝔩 **- Erlang**					⚠️ Does not work by default. PEL provides a fix ✅. • .erl files: ✅ • .escript files: ✅ • .hrl files: ✅ • Erlang functions
F#						
F*						
Factor	𝔓𝔩 **- Factor**					

Forth	𝔓𝔩 **- Forth**					• .fth files ✅ when **pel-use-forth** is **t** .
						    • Variables ✅
						    • Words ✅

Fortran F-90 and F-95		**f90.el**				

Language	PEL PDF Table	Emacs Support file	Feature name	Major Mode Name	Modeline Lighter	Description
Fortran F-77, F-90		**fortran.el**				
Gambit	𝄞ɭ - **Gambit**					
Gerbil	𝄞ɭ - **Gerbil**					
Go	𝄞ɭ - **Go**	**go-mode.el**	go-mode	go-mode	Go	⚠️ Does not work by default. PEL provides a fix ✅. • .go files ✅ • func ✅ • type ✅ • go.mod files ❌
Groovy						
Guile	𝄞ɭ - **GNU Guile**					
Haskell	𝄞ɭ - **Haskell**					
Hop						
HTML				mhtml-mode	XHTML+	• .html files ✅ • headings <h1>, <h2>, …✅
Hy	𝄞ɭ - **Hy**					
Icon		**icon.el**				
IDL		**idlwave.el**				
Janet	𝄞ɭ - **Janet**					
Java						
Javascript		**js.el**				
Julia	𝄞ɭ - **Julia**					
Kotlin						
LaTeX						
Ld - GNU Linker		**ld-script .el**				
Limbo						
Lisp - **Common Lisp**	𝄞ɭ - **Common** **Lisp**					⚠️ Does not support .lisp files by default. PEL provides a fix ✅. • .l, .lsp, .lisp files ✅ • func ✅ • type ✅
LiveScript						
LFE	𝄞ɭ - **LFE**					
Lua	𝄞ɭ -**Lua**					
M4		**m4-mode.el**				
Make	𝄞ɭ - **Make**	**make-mode.el**	make-mode	• makefile-bsdmake-mode	• BSDmakefile	• Macro assignments ✅ • Dependencies ✅ • ❌Includes extra/invalid dependencies in operations when the line includes a colon. Seen on Emacs 26.3. However **makefile-previous-dependency** does work properly. Since this is what's used, need investigation.🚧
Markdown						
ML						
Modula 2	𝄞ɭ-**Modula**	**modula2.el**				
Modula 3	𝄞ɭ-**Modula**					
Nemerle						
NetRexx	𝄞ɭ - **NetRexx**	**netrexx-mode.el**	netrexx-mode	netrexx-mode	Netrexx	• .nrx files: ✅ • class definitions ✅ • method definitions ✅
Newspeak						
Nim	𝄞ɭ - **Nim**					
Object Pascal		**opascal.el**				
Objective-C						
OCaml	𝄞ɭ - **OCaml**					
Octave		**octave.el**				
Odin	𝄞ɭ - **Odin**					
OpenCL						
Opa						
Org Mode	𝄜 **Outline/Org-Mode**			org-mode	Org	• .org files: ✅ • sections ✅ • extension-less files that are identified as org-mode files with file level variables: ❌ (🚧Check for ways to activate Speedbar support. Activating a menu Index does not seem to help).
Oxygene						
Pascal	𝄞ɭ-**Pascal**	**pascal.el**				
Perl	𝄞ɭ - **Perl**	**perl-mode.el**				
Pike	𝄞ɭ - **Pike**					
Postscript		**ps-mode.el**				
Prolog		**prolog.el**				
PureScript						

Language	PEL PDF Table	Emacs Support file	Feature name	Major Mode Name	Modeline Lighter	Description
Python	**𝕻𝕷 - Python**	**python.el**		python-mode	**Python**	• .py files: ✅ 　• class definitions ✅ 　• class method definitions ✅ 　• function definitions ✅ 　• ❌ functions defined inside logical statements (like if statements) and therefore indented are no always reported. 🚧 More investigation needed.
		python-mode.el		python-mode	**Py**	• .py files: ✅ 　• class definitions ✅ 　• class method definitions ✅ 　• function definitions ✅
		⚠️ Both files use the **same** major mode name. python.el is distributed with Emacs, python-mode.el was written before but is not part of the standard distribution and is quite large (1677 functions in version 20201230.2132). Once python-mode.el is installed it normally overrides python.el by being identified before python.el in Emacs load-path. It is not possible to use both and select one dynamically. 🚧 Although this was written first, at this point I'd recommend to rename the mode to py-mode (it would match the mode-line lighter). ⚠️ There is a difference of behaviour between Speedbar (in graphics mode) and SR-Speedbar (in terminal mode). Speedbar seems to detect more Python items. 🚧 More investigation is needed.				
Racket	**𝕻𝕷 - Racket**					
ReasonML						
reStructuredText	**𝕹 reStructuredText**					• .rst, .rst.txt, .stxt files,: ✅ 　• sections ✅
Rexx	**𝕻𝕷 - REXX**	**rexx-mode.el**	rexx-mode	rexx-mode	REXX	• .rex, .rexx, .elx, .ncomm, .cpr files ✅ 　• procedure definitions ✅
Rebol						
Red						
Ruby	**𝕻𝕷 - Ruby**	**ruby-mode.el**				
Rust	**𝕻𝕷 - Rust**					
Sather						
Scala						
Scheme	**𝕻𝕷 - Scheme**	**scheme.el**				
MIT Scheme		**xscheme.el**				
Seed7	**𝕻𝕷-Seed7**	**seed7-mode**	seed7-mode	seed7-mode	seed7	• .sd7, .si7 files. ✅ • iMenu & Speedbar show: function, procedure, interfaces, enum, struct.
Self						
Shell Script	**𝕻𝕷 - UNIX Shell**	**sh-script.el**				
Shen						
Simula		**simula.el**				
Smalltalk						
Solidity						
SQL		**sql.el**				
Swift	**𝕻𝕷-Swift**					
Tcl	**𝕻𝕷 - Tcl**	**tcl.el**	tcl			
TeX						
TypeScript						
V	**𝕻𝕷 - V**					
Vala						
Vera, OpenVera		**vera-mode.el**	vera-mode			
• **Verilog** • **SystemVerilog**		**verilog-mode.el**				
VHDL		**vhdl-mode.el**				
XML						
Zig	**𝕻𝕷 -Zig**					