


















Description	Keystroke	Function		Note		
<div>Seed7-specific abbreviations</div> <div>See also: <a href="#">ℹ Abbreviations</a></div>	<div>The <a href="#">seed7-mode</a> supports Seed7-specific abbreviations for <a href="#">Emacs abbrev-mode</a> when the <a href="#">seed7-support-abbrev-mode</a> customizable user-option is on (the default).</div> <div><ul style="list-style-type: none"><li>All abbreviation and their text expansion are set by the <a href="#">seed7-abbreviations</a> customizable user-option list.<ul style="list-style-type: none"><li>The default list is shown below. All abbreviations start with a semi-colon.</li><li>You can modify the default and add other abbreviations through customization.</li></ul></li><li>These abbreviations are <i>system</i> abbreviations, treated specially by the abbrev-mode in the sense that you can not modify them dynamically via the abbrev-mode commands. But you don't need to since they can be modified by customization.</li><li>Of course you can create other abbreviations that help you write code or comments. See <a href="#">ℹ Abbreviations</a> for more details related to abbreviations in Emacs.</li><li>To expand the abbreviations, the <a href="#">abbrev-mode</a> must be active: type the abbreviation followed by a word-separating character, such as <b>&lt;space&gt;</b>, <b>&lt;RET&gt;</b>, semi-colon, period, comma, etc..</li></ul></div>					
Pragmas & in-statement keywords	pragmas		in-statement keywords		in-middle statement keywords	
	<b>;de</b>	decls	<b>;fo</b>	forward	<b>;dt</b>	downto
	<b>;in</b>	info	<b>;n</b>	new	<b>;exc</b>	exception
	<b>;li</b>	library	<b>;no</b>	noop	<b>;lo</b>	local
	<b>;msg</b>	message	<b>;ra</b>	raise	<b>;pa</b>	param
	<b>;na</b>	names	<b>;rt</b>	return	<b>;rg</b>	range
	<b>;syn</b>	syntax			<b>;rs</b>	result
	<b>;sys</b>	system			<b>;st</b>	step
<b>;tr</b>	trace					
Block clause keywords	block clause keywords					
	<b>;ct</b>	catch	<b>;e</b>	else	<b>;o</b>	otherwise
			<b>;ei</b>	elsif	<b>;w</b>	when
Pre-defined types	pre-defined types					
	<b>;a</b>	array	<b>;db</b>	database	<b>;rat</b>	rational
	<b>;bi</b>	bigInteger	<b>;du</b>	duration	<b>;rf</b>	reference
	<b>;br</b>	bigRational	<b>;en</b>	enum	<b>;rfi</b>	ref_list
	<b>;b3</b>	bin32	<b>;ex</b>	expr	<b>;s</b>	set
	<b>;b6</b>	bin64	<b>;fi</b>	file	<b>;sq</b>	sqlStatement
	<b>;bt</b>	bitset	<b>;fs</b>	fileSys	<b>;sti</b>	string
	<b>;bo</b>	boolean	<b>;fl</b>	float	<b>;stu</b>	struct
	<b>;bs</b>	bstring	<b>;h</b>	hash	<b>;tx</b>	text
	<b>;ca</b>	category	<b>;i</b>	integer	<b>;ti</b>	time
	<b>;c</b>	char	<b>;ob</b>	object	<b>;ty</b>	type
	<b>;cf</b>	clib_file	<b>;pro</b>	process	<b>;v</b>	void
	<b>;co</b>	color	<b>;pr</b>	program	<b>;pw</b>	PRIMITIVE_WINDOW
	<b>;cx</b>	complex				
Pre-defined constants	pre-defined constants					
	<b>;em</b>	empty	<b>;f</b>	FALSE	<b>;inf</b>	Infinity
			<b>;t</b>	TRUE		
Pre-defined variables	pre-defined variables					
	<b>;ck</b>	CONSOLE_KEYBOARD	<b>;sc</b>	STD_CONSOLE	<b>;sn</b>	STD_NULL
	<b>;gk</b>	GRAPH_KEYBOARD	<b>;se</b>	STD_ERR	<b>;so</b>	STD_OUT
	<b>;kb</b>	KEYBOARD	<b>;si</b>	STD_IN		
Errinfo values	errinfo values					
	<b>;ok</b>	OKAY_NO_ERROR	<b>;dse</b>	DESTROY_ERROR	<b>;me</b>	MEMORY_ERROR
	<b>;ae</b>	ACTION_ERROR	<b>;fe</b>	FILE_ERROR	<b>;ne</b>	NUMERIC_ERROR
	<b>;ce</b>	COPY_ERROR	<b>;ge</b>	GRAPHIC_ERROR	<b>;oe</b>	OVERFLOW_ERROR
	<b>;cre</b>	CREATE_ERROR	<b>;ie</b>	INDEX_ERROR	<b>;re</b>	RANGE_ERROR
	<b>;dbe</b>	DATABASE_ERROR	<b>;ine</b>	IN_ERROR		

<b>Syntax-aware automatic Indentation</b>	Unless explicitly disabled by setting the <b>seed7-auto-indent</b> user-option to nil, the <b>&lt;tab&gt;</b> and <b>&lt;return&gt;</b> key perform syntax-aware automatic indentation of Seed7 code. The <b>&lt;return&gt;</b> key also supports the auto-fill-mode. <ul style="list-style-type: none"> <li>The number of columns used for each indentation level is controlled by the <b>seed7-indent-width</b> user-option, which defaults to 2.</li> <li>Emacs can use hard tabs as appropriate when you activate the <b>indent-tabs-mode</b>. If it is off Emacs only uses space characters.</li> </ul> See the <a href="#">ℹ Indentation</a> page for more information related to indentation control and commands.		
<b>Auto-fill-mode</b>	The <b>seed7-mode</b> supports Emacs auto-fill-mode, useful when typing comments. See the <a href="#">ℹ Fill/Justify</a> page and the <b>pel-comment-style</b> command above.		
<b>Marking</b>	The <b>seed7-mode</b> support specialized marking. It is also compatible with other Emancs native and package commands. See <a href="#">ℹ Marking</a> for more information.		
<b>Mark current callable</b>	<b>C–M–h</b>	(seed7-mark-defun)	Mark the current Seed7 function or procedure. <ul style="list-style-type: none"> <li>Put the mark at the end and point at the beginning.</li> <li>If point is before or between 2 functions or procedure, mark the next one.</li> </ul>

Description	Keystroke	Function	Note
Code Navigation	The <b>seed7-mode</b> supports syntax-aware procedure/function as well as block aware navigation commands <ul style="list-style-type: none"> <li>• PEL provides some extra key bindings to Emacs native navigation commands.</li> <li>• The <b>seed7-mode</b> also supports imenu-compliant parsing which enables the ability to use a large set of navigation packages.               <ul style="list-style-type: none"> <li>• See <a href="#">navigation by symbol definition</a> in the <a href="#">Navigation</a> page for more information.</li> </ul> </li> <li>• The <b>seed7-mode</b> navigation commands display the name and type of block found when the <b>seed7-verbose-navigation</b> user-option is turned on (set to <b>t</b>).</li> </ul>		
Shift-Selection	If you press and hold the <b>shift</b> key while typing a movement command, that sets the mark before moving point (Emacs name for cursor) so that the region extends from the original point to its new position. This is called: <b>Shift-Selection</b> . <ul style="list-style-type: none"> <li>• <b>Shift selection</b> is supported by some navigation commands, not all. The following symbols are used to identify whether the command supports shifts selection:               <ul style="list-style-type: none"> <li>•  This command supports shift selection in GUI and terminal mode.</li> <li>•  This command supports shift selection only in GUI mode.</li> <li>•  This command supports shift selection in GUI mode and also in terminal mode under some conditions (described in the description cell for the command).</li> <li>•  This command does <b>not</b> support shift selection. Sometimes for this you can first set the mark before moving.</li> </ul> </li> <li>• Pressing the Shift key when using the key binding for commands that do not show any of these 3 arrows have no impact on the shift selection (and may be inappropriate for the command).</li> </ul>		
Move Point	The following sub-sections describe how to navigate across various types of textual and syntactical entities.		
<ul style="list-style-type: none"> <li>• <a href="#">by defun</a></li> </ul>	The commands move point by Seed7 function and procedure definitions. <p> In PEL:</p> <ul style="list-style-type: none"> <li>• The <b>&lt;f12&gt;</b> cursor key mappings use <b>&lt;up&gt;</b> and <b>&lt;down&gt;</b> to move to the beginning or end of the function, procedure or other blocks.</li> <li>• The <b>&lt;f6&gt;</b> cursor key mapping use <b>&lt;up&gt;</b> and <b>&lt;down&gt;</b> to move to the beginning or end of the function or procedure.</li> <li>• The <b>&lt;f6&gt;</b> cursor key mapping use <b>&lt;right&gt;</b> and <b>&lt;left&gt;</b> to move to the beginning or end of the next/previous function or procedure.</li> </ul> <ul style="list-style-type: none"> <li>• The advantage of the <b>&lt;f6&gt;</b> and <b>&lt;f12&gt;</b> key bindings is they support <b>Shift-Selection</b> for Emacs in terminal mode, as opposed to the key bindings that sue the Control key which can only support <b>Shift-Selection</b> when Emacs is running in Graphics mode.</li> </ul>		
Backward to beginning of defun  	<div> <b>&lt;f6&gt;</b> <b>&lt;up&gt;</b> </div> <div> <ul style="list-style-type: none"> <li>• <b>C-M-a</b></li> <li>• <b>C-M-&lt;home&gt;</b></li> <li>• <b>C-[ C-a</b></li> <li>• <b>Esc C-a</b></li> </ul> </div>	(seed7-beg-of-defun &optional N SILENT DONT-PUSH-MARK)	Move backward to the beginning of a defun. <ul style="list-style-type: none"> <li>• With ARG, do it that many times. Negative ARG means move forward to the ARGth following beginning of defun.</li> <li>• Prints the name of the function or procedure in the message area.</li> <li>• On successful move, you can move back to original position by typing <b>M-` , &lt;f6&gt; &lt;f6&gt;</b> or <b>&lt;f11&gt; . `</b></li> <li>• Supports <b>Shift-Selection</b> in graphics mode. <b>&lt;f6&gt;&lt;up&gt;</b> supports it in terminal mode too.</li> </ul>
Forward to end of defun  	<div> <b>&lt;f6&gt;</b> <b>&lt;down&gt;</b> </div> <div> <ul style="list-style-type: none"> <li>• <b>C-M-e</b></li> <li>• <b>C-M-&lt;end&gt;</b></li> <li>• <b>C-[ C-e</b></li> <li>• <b>Esc C-e</b></li> </ul> </div>	(seed7-end-of-defun &optional N SILENT DONT-PUSH-MARK)	Move forward to next end of defun. <ul style="list-style-type: none"> <li>• With argument, do it that many times. Negative argument -N means move back to Nth preceding end of defun.</li> <li>• Prints the name of the function or procedure in the message area.</li> <li>• On successful move, you can move back to original position by typing <b>M-` , &lt;f6&gt; &lt;f6&gt;</b> or <b>&lt;f11&gt; . `</b></li> <li>• Supports <b>Shift-Selection</b> in graphics mode. <b>&lt;f6&gt;&lt;down&gt;</b> supports it in terminal mode too.</li> </ul>
Forward to start of next defun 	<b>&lt;f6&gt;</b> <b>&lt;right&gt;</b>	(seed7-beg-of-next-defun &optional N SILENT DONT-PUSH-MARK)	Move forward to the beginning of the next function or procedure. <ul style="list-style-type: none"> <li>• With optional argument N, repeat the search that many times.</li> <li>• Move back to previous position with <b>M-` , &lt;f6&gt; &lt;f6&gt;</b> or <b>&lt;f11&gt; . `</b></li> <li>• Supports <b>Shift-Selection</b>.</li> </ul>
Backward to end of previous define   will be replaced	<b>&lt;f6&gt;</b> <b>&lt;left&gt;</b>	(pel-end-of-previous-defun &optional SILENT DONT-PUSH_MARK)	Move backwards to the end of the previous function definition. <ul style="list-style-type: none"> <li>• Issue user error not find end of previous function unless SILENT is non-nil.</li> <li>• If the end of previous function is found, push the start location to the mark ring unless DONT-PUSH_MARK is non-nil.               <ul style="list-style-type: none"> <li>• Move back to previous position with <b>M-` , &lt;f6&gt; &lt;f6&gt;</b> or <b>&lt;f11&gt; . `</b></li> </ul> </li> <li>• Supports <b>Shift-Selection</b>.</li> </ul>
Forward to end of current block statement 	<b>&lt;f12&gt;</b> <b>&lt;down&gt;</b>	(seed7-to-block-forward)	Move forward from the beginning of a Seed7 block to its end. <ul style="list-style-type: none"> <li>• Supports the Seed7 <code>if/end if</code>, <code>block/end block</code>, <code>case/end case</code>, <code>enum/end enum</code>, <code>for/end for</code>, <code>repeat/until</code>, <code>struct/end struct</code>, <code>while/end while</code>. It also supports moving to the end of a function or a procedure.               <ul style="list-style-type: none"> <li>• Move back to previous position with <b>M-` , &lt;f6&gt; &lt;f6&gt;</b> or <b>&lt;f11&gt; . `</b></li> </ul> </li> <li>• Supports <b>Shift-Selection</b>.</li> </ul>
Backward to beginning of current block statement 	<b>&lt;f12&gt;</b> <b>&lt;up&gt;</b>	(seed7-to-block-backward)	Move backward from the end of a Seed7 block to its beginning. <ul style="list-style-type: none"> <li>• supports the Seed7: <code>if/end if</code>, <code>block/end block</code>, <code>case/end case</code>, <code>enum/end enum</code>, <code>for/end for</code>, <code>repeat/until</code>, <code>struct/end struct</code>, <code>while/end while</code>. It also supports moving to the end of a function or a procedure.               <ul style="list-style-type: none"> <li>• Move back to previous position with <b>M-` , &lt;f6&gt; &lt;f6&gt;</b> or <b>&lt;f11&gt; . `</b></li> </ul> </li> <li>• Supports <b>Shift-Selection</b>.</li> </ul>
Compilation	The Seed7 source code is either interpreted or compiled. In both cases you can verify it's validity by performing a static check of the code, an operation that does not generate any binary file but perform the same language checking that the compiler will do.		
Static check or compile Seed7 file	<b>&lt;f12&gt;</b> <b>c</b>	(seed7-compile &optional COMPILE)	Static check current Seed7 file, show errors in compilation-mode buffer. <ul style="list-style-type: none"> <li>• If optional COMPILE argument set, compile the file to executable instead.</li> </ul>
See <a href="#">Navigation Mode</a>	<ul style="list-style-type: none"> <li>• For example: type <b>C-u &lt;f12&gt; c</b> for compiling the file. Without the <b>C-u</b> prefix it just static checks the file, an operation that is much faster.</li> <li>• The static analysis is performed by the command identified by the <b>seed7-checker</b> user-option, which defaults to s7-check.</li> <li>• You can specify any command with or without its path.</li> <li>• The compilation is performed by the command identified by the <b>seed7-compiler</b> user-option, which defaults to s7c.</li> <li>• You can specify any command with or without its path.</li> <li>• Any detected error is shown in a "compilation" <a href="#">Navigation Mode</a> buffer. Use it to navigate to the line of the code in error.</li> </ul>		

## Emacs & Seed7 — References

Document	Notes	
The Seed7 Programming Language	<ul style="list-style-type: none"> <li>• <a href="#">Seed7 @ Wikipedia</a></li> <li>• <a href="#">Seed7 Home</a></li> <li>• <a href="#">Seed7 @ Github</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Seed7 Manual</a></li> <li>• <a href="#">Seed7 Language Reference</a></li> </ul>
	<ul style="list-style-type: none"> <li>• <a href="#">Seed7 @ reddit</a></li> <li>• <a href="#">Seed7 @ Rosetta code</a></li> </ul>	
Presentations	<ul style="list-style-type: none"> <li>• <a href="#">The Seed7 Programming Language @ Youtube</a></li> <li>• <a href="#">The Seed7 Programming Langage Presentation at CPP Vienna @ Youtube</a></li> <li>• <a href="#">Another speech about the Seed7 Programming Language</a></li> </ul>	
	Modern Extensible Languages. Daniel Zingaro, McMaster U. April 11, 2007 (pdf)	
Emacs support  is partial, not yet completed.	<ul style="list-style-type: none"> <li>• <a href="#">seed7-mode @ Github</a></li> </ul>	
Other tools that support Seed7	<ul style="list-style-type: none"> <li>• <b>ripgrep</b> a very fast grep replacement - supports seed7 file types with <a href="#">this pull request accepted April 7 2025</a> <ul style="list-style-type: none"> <li>• With this version of ripgrep, you can use <a href="#">deadgrep</a> to identify Seed7 files by name in Emacs. See <a href="#">Navigation Mode</a></li> </ul> </li> <li>• <b>ugrep</b> another very fast grep replacement - supports seed7 files <a href="#">with this pull request</a> .</li> </ul>	