





graphviz-dot— mode: Graphviz Dot Mode

Description	Keystroke	Function	Note
Graphviz Dot	 With the graphviz-dot-mode package external package installed, Emacs can support Graphviz-Dot markup files and can also render the resulting graphic files.  PEL downloads, installs and activates it when the pel-use-graphviz-dot user option is set to t . <ul style="list-style-type: none">By default, Emacs treats files with the .dot and .gv extensions as DOT files.<ul style="list-style-type: none">With PEL, you can add more file types by adding the association to the pel-auto-mode-alist user option. PEL adds these to Emacs auto-mode-alist.		
Open this PDF file. See also: 🔗 Help/Info	<ul style="list-style-type: none"><f11> SPC g <f1><f12> <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the local copy of the M Graphviz Dot PDF file unless a command prefix (like C-u) was used. In that case it opens the Github-hosted file instead.
Customize PEL Graphviz Dot Support See also: 🔗 Customize	<ul style="list-style-type: none"><f11> SPC g <f2><f12> <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Graphviz-Dot support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
Customize Emacs Graphviz Dot Support See also: 🔗 Customize	<ul style="list-style-type: none"><f11> SPC g <f3><f12> <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Graphviz-Dot support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
Activate Graphviz Dot mode	<f11> M-g	(graphviz-dot-mode)	Toggle use of the graviz-dot-mode to edit Graphviz Dot files.
Editing Dot syntax	The following commands help writing Graphviz DOT code. See the following reference documents: <ul style="list-style-type: none">Documentation, which contains the following and much more:<ul style="list-style-type: none">The DOT LanguageNode, Edge and Graph AttributesNode ShapesArrow ShapesColors		
Indent current line See also: 🔗 Indentation	<tab>	(indent-for-tab-command &optional ARG)	Indent current line of dot code (regardless of point position within the line)
		(graphviz-dot-indent-line)	
Indent the graph statement at point	<ul style="list-style-type: none">C-M-q<f12> <tab>	(graphviz-dot-indent-graph)	Indent the graph/digraph/subgraph where point is at. <ul style="list-style-type: none">This will first reach the beginning of the graph were point is at, and then indent this and each subgraph in it.
Break line and indent	M-j	(indent-new-comment-line &optional SOFT)	Break line at point and indent, continuing comment if within one. <ul style="list-style-type: none">This indents the body of the continued comment under the previous comment line.This command is intended for styles where you write a comment per line, starting a new comment (and terminating it if necessary) on each line.If a fill column is specified, it overrides the use of the comment column or comment indentation.The inserted newline is marked hard if variable ‘use-hard-newlines’ is true, unless optional argument SOFT is non-nil.
Compiling			
Compile the graph	<ul style="list-style-type: none">C-c C-c<f12> c	(compile COMMAND &optional COMINT)	Compile the graph using the program identified by graphviz-dot-dot-program to the output format identified by graphviz-dot-preview-extension. The command is shown in the mini buffer, allowing modification before execution.
Jump to next match	<ul style="list-style-type: none">C-x `M-g nM-g M-n	(next-error &optional ARG RESET)	A prefix ARG specifies how many error messages to move; negative means move back to previous error messages. Just C-u as a prefix means reparse the error message buffer and start at the first error.
Jump to previous match	<ul style="list-style-type: none">M-g pM-g M-p	(previous-error &optional N)	Prefix arg N says how many error messages to move backwards (or forwards, if negative).
Viewing			
Preview image in Emacs buffer	<ul style="list-style-type: none">C-C C-p<f12> p	(graphviz-dot-preview)	Compile the graph and preview it in an other buffer.   Only works in graphics mode.
View image using external program	C-c C-v	(graphviz-dot-view)	Run an external viewer. <ul style="list-style-type: none">This creates an external process every time it is executed. If ‘graphviz-dot-save-before-view’ is set, the current buffer is saved before the command is executed.
Graphviz Dot Variables			
graphviz-dot-dot-program	Program used to compile the graphs. Default := ‘dot’.		
graphviz-dot-preview-extension	File type to use for output. Default := ‘png’.		
graphviz-dot-view-command	Command to run when ‘graphviz-dot-view’ is executed. Default:= ‘dotty %s’		
graphviz-dot-view-edit-command	If the user should be asked to edit the view command. Default:= nil		
graphviz-dot-save-before-view	Automatically save current buffer berore ‘graphviz-dot-view’.		
graphviz-dot-mode-hook	Hook variable for initialization of the mode.		

Graphviz Dot Mode — References

Description & URL	Notes
Graphviz Dot	
GraphViz @ Wikipedia	Overview description with various links.
GraphViz.org - Graphviz Graph Visualization Software	Graphviz home page. Includes: <ul style="list-style-type: none">GalleryDocumentation, which contains the following and much more:<ul style="list-style-type: none">The DOT LanguageCommand-line UsageOutput formatsNode, Edge and Graph AttributesNode ShapesArrow ShapesColorsResources
Graphviz Dot Mode for Emacs	
graphviz-dot-mode @ GitHub	