

PlantUML-Mode - UML Drawings🚧

| Description | Keystroke | Function | Note |
|--|---|---|--|
| PlantUML - Mode | <div><div>📦</div>The plantuml-mode external package allows you to render PlantUML markup to create UML drawing image or ASCII-art text. See also the The Hitchhiker's Guide to PlantUML.</div> <div><div>📦</div>The flycheck-plantuml external package integrates plantuml-mode with flycheck to automatically check the syntax of the plantuml files on the fly.</div> <div><div>🔗</div>PEL activates the plantuml-mode package when the pel-use-plantuml is non-nil. You can set it to t or to server. When set to t it forces the use of a local platuml.jar program to render the markup. When set to server it sends the plantuml markup to the PlantUML server to render it. These values are set in the plantuml-default-exec-mode user option.</div> <div><div>🔗</div>PEL activates the flycheck-plantuml package when the pel-use-flycheck-plantuml user option is set to t.</div> <div><div>⚙️</div>The plantuml-mode package can also be customized via the plantuml customization group: use <F11> <f1> G plantuml</div> <div><div>📋</div>PEL provides 2 different key prefixes:<ul style="list-style-type: none">pel:plantuml, mapped to <f11> D u, for global commands that be be issued from any major mode,pel:for-plantuml, mapped to <f12> once the buffer major mode is plantuml-mode. To put a buffer in plantuml-mode use <f11> D u u.</div> <div><div>⚠️</div>Text rendering of PlantUML diagrams is limited to 4096 bytes of text. Attempting to render a larger description in 'txt' (text) mode fail. Rendering of diagrams larger than 4096 bytes of text works in graphics mode. Tested on macOS with Graphviz Dot 2.44.0 and PlantUML 1.2020.15.</div> | | |
| Open this PDF file. See also: 📖 Help/Info | <ul style="list-style-type: none"><f11> SPC u <f1><f12> <f1> | (pel-help-pdf) | Open 📖 PlantUML local PDF file. |
| 📖 Customize PEL PlantUML support | <ul style="list-style-type: none"><f11> D u <f2><f11> SPC u <f2><f12> <f2> | (pel-customize-pel &optional OTHER-WINDOW) | Customize PEL PlantUML support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window. |
| 📖 Customize Emacs PlantUML support | <ul style="list-style-type: none"><f11> D u <f3><f11> SPC u <f3><f12> <f3> | (pel-customize-library &optional OTHER-WINDOW) | Customize Emacs PlantUML support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window. |
| Download & install latest version of PlantUML JAR file | <f11> D u M-d | (plantuml-download-jar) | Download the latest PlantUML JAR file and install it into 'plantuml-jar-path' (a user option that can be customized). |
| Se the PlantUML Execution mode | <f11> D u M-x | (plantuml-set-exec-mode MODE) | Set the execution mode MODE for PlantUML. Prompts for mode. The following modes are available: executable, jar, server. |
| Activate PlantUML mode | <f11> D u u | (plantuml-mode) | Activate plantuml-mode |
| Enable PlantUML debug messages | <ul style="list-style-type: none"><f11> SPC u M-d<f12> M-d | (plantuml-enable-debug) | Enables debug messages into the "PLANTUML Messages" buffer. |
| Disable PlantUML debug messages | <ul style="list-style-type: none"><f11> D u M-D<f12> M-D | (plantuml-disable-debug) | Stops any debug messages to be added into the "PLANTUML Messages" buffer. |
| Indent the PlantUML line | <ul style="list-style-type: none"><f11> SPC u <tab><f12> <tab> | (plantuml-indent-line) | Indent the current line to its desired indentation level. <ul style="list-style-type: none">Restore point to same position in text of the line as before indentation. |
| Preview UML diagram from plantUML source in current region (if marked) or entire buffer | <ul style="list-style-type: none"><f11> SPC u p<f12> pC-c C-c | (plantuml-preview PREFIX) | Preview diagram from the PlantUML sources. <ul style="list-style-type: none">Uses the current region if one is active, or the entire buffer otherwise.Uses prefix (as PREFIX) to choose where to display it:<ul style="list-style-type: none">4 (when prefixing the command with C-u) -> new window16 (when prefixing the command with C-u C-u) -> new frame.else -> new buffer |
| Preview UML diagram from plantUML source in current plantUML region of commented source code | <f11> D u p | (pel-render-commented-plantuml PREFIX &optional POS) | Render the PlantUML markup embedded in current mode comment. <ul style="list-style-type: none">Use region if identified otherwise use PlantUML block at point.Uses prefix (as PREFIX) to choose where to display it:<ul style="list-style-type: none">4 (when prefixing the command with C-u) -> new window16 (when prefixing the command with C-u C-u) -> new frame.else -> new bufferThis can be used inside buffer using any major mode, when PlantUML markup is embedded inside source code comment. <div>👉 Use this in source code to describe your code architecture with PlantUML markup, then generate the UML rendering by moving point inside the PlantUML block and issuing this command.</div> |
| Preview UML diagram from plantUML source in current buffer | <ul style="list-style-type: none"><f11> SPC u b<f12> b | (plantuml-preview-buffer PREFIX) | Preview diagram from the PlantUML sources in the current buffer. <ul style="list-style-type: none">Uses prefix (as PREFIX) to choose where to display it:<ul style="list-style-type: none">4 (when prefixing the command with C-u) -> new window16 (when prefixing the command with C-u C-u) -> new frame.else -> new buffer |
| Preview UML diagram from plantUML source in current plantUML block | <ul style="list-style-type: none"><f11> SPC u c<f12> c | (plantuml-preview-current-block PREFIX) | Preview diagram from the PlantUML sources from the previous @startuml to the next @enduml. <ul style="list-style-type: none">Uses prefix (as PREFIX) to choose where to display it:<ul style="list-style-type: none">4 (when prefixing the command with C-u) -> new window16 (when prefixing the command with C-u C-u) -> new frame.else -> new buffer |
| Preview UML diagram from plantUML source in current plantUML region | <ul style="list-style-type: none"><f11> SPC u r<f12> r | (plantuml-preview-region PREFIX BEGIN END) | Preview diagram from the PlantUML sources in from BEGIN to END. <ul style="list-style-type: none">Uses the current region when called interactively.Uses prefix (as PREFIX) to choose where to display it:<ul style="list-style-type: none">4 (when prefixing the command with C-u) -> new window16 (when prefixing the command with C-u C-u) -> new frame.else -> new buffer |
| Complete PlantUML symbol | <ul style="list-style-type: none"><f11> SPC u /<f12> / | (plantuml-complete-symbol) | Perform keyword completion on word before cursor. |
| Select output type | <ul style="list-style-type: none"><f11> SPC u o<f12> o | (plantuml-set-output-type TYPE) | Set the desired output type (as TYPE) for the current buffer. <ul style="list-style-type: none">If the major mode of the current buffer mode is not plantuml-mode, set the default output type for new buffers.The basic output modes are: svg, png and text. The sag and png mode are only available when Emacs runs in graphics mode. |

PlantUML — References

| Topic & Link | Note |
|---|---|
| PlantUML @ Wikipedia | Provides an overview |
| PlantUML home page | PlantUML home page. This site has a lot of publicity, but it's really the official site. It has links to important documents, like: <ul style="list-style-type: none">• PlantUML PDF Guide• The Hitchhiker's Guide to PlantUML• PlantUML Standard Library• Preprocessing• F.A.Q• and others... |
| plantuml-mode @ GitHub | Emacs plantuml-mode package to use PlantUML within Emacs |
| flycheck-plantuml @ Github | Emacs package: Flycheck for plantuml syntax |