The Todo-Tree Extension

> **≡ Abstract**
>
> *This documentation provides a quick reference to VSCode's `Todo-Tree` extension. Its primary purpose is to describe the way it is configured in this project's workspace and how it is meant to be used.*

Todo-Tree Overview

The extension Todo-Tree uses ripgrep to search the workspace for keywords like `TODO`, `FIXME` or `NOTE` and displays the results in an interactive *tree view* that can be sorted, filtered and exported.

The `TAGS` it looks for can be customized, and it can also be configured to further bundle the results into `SUBTAGS`. Those can also be highlighted in the editor according to a customizable color scheme.

The following tables list the tags that are currently made available by the extension's configuration defined in vscode-index.code-workspace.

> **ⓘ Note on icons**
>
> The icons in use are the ones provided by the octicons set. Due to poor and unstable support of colors, using alternative sets is not recommended.

Todo-Tree's Tag Categories

The tags defined in this project's workspace are spliced into 7 disjoint categories which definitions are summarized in the table below. More details as well as the tags available in each category are provided in the next sections.

Category	Description
Issues	Tags identifying major issues and calling for immediate action
Alerts	Tags pointing to issues which severity is not yet established
Helpers	Tags which are used to hold information about the project
Markers	Tags identifying editable parts of the project like templates
Dividers	Tags which are used to visually segment the project's file
Records	Tags containing general tasks meant to produce tasks' lists
Hunches	Tags referring to modifications not to be in the next release

Issues Tag Category

Tags identifying major issues and calling for immediate action. Freeing the project from all tags falling into that category should be the developers' highest priority. No release should be considered until every single one of them has been addressed and removed from the project, no matter its present state and the location of the tags. For this reason, the extension should be configured to display the tags from this category wherever the tree is generated from.

	TAG	Description
	BUG	Describes an identified bug (still) needing to be addressed.
	FIXME	Formally identifies and describes something needing a fix.
	FINISH	Documents a non-critical task that still needs to be done.

Alerts Tag Category

Tags pointing to issues whose severity is still to be established, but that are potentially damaging enough to be considered as a threat. The tags from this category are typically transitional, meaning that addressing such a tag is expected to yield its requalification as either a *record* or an *issue* tag. Cleaning the project from all *alerts* should be every developer's 2nd highest priority, right after clearing all tags marked as *issues*. Again, no release should be considered as long as the todo-tree displays any of these tags.

	TAG	Description
	CHECK	Points out that something needs to be checked or tested.
	TEMP	A temporary piece of code to be deleted in the short term.
	REVIEW	Calls for something's critical in an improvement aim.

Records Tag Category

Tags that are used to produce to-do lists of different natures. Tags from this category are mostly used to keep track of the project's progress and to make sure that none of the features expected to be part of the next release is forgotten. For the project to be considered as ready for release, all tags from this category should be either addressed or moved to the *hunches* category. As a result, while this category is expected to be the most densely populated one among the generated tree at the heart of every development phase, it should vanish completely when approaching a release.

	TAG	Description
	TEST	Documents a non-critical task that still needs to be done.
	TODO	Documents a non-critical task that still needs to be done.
	IMPROVE	To Be Done: something that has not even been started yet

Hunches Tag Category

Low-priority tags refer to modifications that are not meant to be part of the next release, whether because they are considered to be non-essential, or because the underlying features are not advanced enough to qualify for inclusion. This category also includes all the tags referencing changes that have been planned (independently of their status) as well as the eventual mentions of ideas or proposals that have not yet been evaluated. Tags from this category have no impact on the project's readiness for release, the reason why the extension may be safely configured to hide them whenever the tree is generated from outside the development environment.

	TAG	Description
	IDEA	Keeps track of a suggestion seemingly worth a consideration.
	LEARN	A topic of interest that should be studied, an article to read, etc …
	ROADMAP	To Be Done: something that has not even been started yet

Markers Tag Category

Tags identifying editable parts of the project, either by delimiting customizable areas or by being themselves placeholders meant to be replaced. Tags from this category typically call for actions on a template that are meant to happen at a pre-determined point in the project's life cycle. Depending on its nature, each of these tags may be repeatedly needed and, thus, persist throughout all project's life cycles, or serve only once and then be removed. As a result, whether or not their presence impacts the project's release readiness must be evaluated on a case-by-case basis. It is however very unlikely that any of them should be part of the project's final distribution.

	TAG	Description
Aa	@@	Prefixes keywords that are placeholders for customizable values
	OPTION	Highlights a chunk of code that is one among other alternatives

	TAG	Description
✏️	EDIT	Highlights the beginning of an editable chunk of code like a template's customizable area.

Helpers Tag Category

Tags which are used to hold information about the project that is central enough to its development that it is worth using the tree to keep track of them. In the same aim, tags from this category may also be used to bookmark locations in the project that are of particular interest and, thus, need to be accessed frequently and/or simply as a reminder for some easily forgettable action that is important to be taken. These tags' lifetime typically spans over the whole development and has no impact on whether the project can be released or not. As a consequence, the extension should be configured to hide this category's tags whenever generating the tree from the release or the deployment environment.

	TAG	Description
🔗	LINK	Serves as shortcut to an editable region within, typically, templates
ⓘ	NOTE	A general note or comment calling for no action of any kind
⚠️	CAUTION	A general note or comment calling for no action of any kind

Dividers Tag Category

Tags that are used to visually segment the project's file. The extension is configured to provide 6 levels of sectioning, starting with *parts* and *chapters* and ending with *sub-sub-sub-sections*. Obviously, tags from this category call for no action at all and are intrinsically meant to never be deleted; their number even has a vocation to increase with the size of the project and, thus, over time. It is worth noting that the case of *dividers* is in complete contrast to this of *helpers*. Indeed, unlike *helpers*, *dividers* have no interest at all in the listing ability of the extension. Instead, they benefit from the fact that tags can be highlighted within the project's source files.

	TAG	Description
📑	PART	Whole-line speparator.
📖	CHAPTER	A chapter using the *line* setting.
▤	SECTION	A section using the *text-and-comment* setting.
🔖	SUBSECTION	A sub-section using the *tag-and-comment* setting.
☰	BLOCK	A paragraph or any enumeration like, typically, a list.
▶	ITEM	Highlights a given element in a block like, typically a list entry.