vscode-texinfo

Texinfo Support for Visual Studio Code, version 0.2.0
Short Contents

1 Overview .................................................. 1
2 Installation ............................................. 2
3 Basic Usage ............................................. 3
4 Advanced Usage ................................. 5
# Table of Contents

1 Overview ................................................. 1
   1.1 Contributing ........................................... 1

2 Installation ............................................ 2
   2.1 Manual Installation ...................................... 2
   2.2 Build from Source ...................................... 2

3 Basic Usage ........................................... 3
   3.1 Syntax Highlighting ...................................... 3
   3.2 Code Completion ........................................ 3
   3.3 Block Folding .......................................... 3
   3.4 Breadcrumb Navigation ................................. 4

4 Advanced Usage ....................................... 5
   4.1 Version Indicator ....................................... 5
   4.2 HTML Preview .......................................... 5
      4.2.1 Custom CSS .......................................... 6
      4.2.2 Goto Node .......................................... 6
      4.2.3 Misc Settings ...................................... 6
   4.3 Diagnosis .............................................. 6
1 Overview

Texinfo is a typesetting language designed for writing software manuals. It’s the official documentation format for GNU projects, but not as popular in modern non-GNU free software projects.

One of the main reasons is the lack of editor support. While Emacs does offer a “texinfo-mode”, however, Emacs is not widely used among average software developers.

We believe that Texinfo deserves more users, for it is an excellent format for writing software manuals, as well as other technical materials. We chose Visual Studio Code, one of the most popular code editors as of year 2020, and developed this very extension, vscode-texinfo, which provides some useful features for Visual Studio Code regarding the Texinfo format, in the hope that more developers can benefit from it.

vscode-texinfo is free software. You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

1.1 Contributing

This project is hosted on Savannah. Any kind of contribution is welcome, including bug reports, patches, and general discussions regarding the usage of vscode-texinfo.

Before you submit something, please make sure that you have read this manual, and no one else has posted the same content.
## 2 Installation

vscode-texinfo cannot be run standalone. It can only be used as an extension of Visual Studio Code.

It is recommended to use a free distribution of Visual Studio Code, VSCodium, instead of the official binary release, which is proprietary software. You can also build it from source yourself.

**Note:** Theoretically vscode-texinfo can work with any version of Visual Studio Code since 1.40, but not all versions are tested. It’s recommended to install a latest release.

The easiest way to install vscode-texinfo is from the extension marketplace. Click the “Extensions” tab on Visual Studio Code’s sidebar, type “texinfo” and you can find and install this extension (Extension ID: CismonX.texinfo).

### 2.1 Manual Installation

You can manually download the .vsix file, from either Savannah’s download area, Open VSX Registry, or the proprietary Visual Studio Marketplace.

Before you install a .vsix file downloaded from a third party, you should check whether the file matches a trusted signature:

```bash
wget -O cismonx.gpg.asc "https://sv.gnu.org/people/viewgpg.php?user_id=214244"
gpg --import cismonx.gpg.asc

wget "https://dl.sv.gnu.org/releases/vscode-texinfo/texinfo-0.2.0.vsix.sig.asc"
gpg --verify texinfo-0.2.0.vsix.sig.asc texinfo-0.2.0.vsix
```

Finally, install the .vsix file to Visual Studio Code via command palette: **Extensions**: Install from VSIX....

### 2.2 Build from Source

You can generate the .vsix file from the source code of vscode-texinfo.

First, clone the source code repository:

```bash
git clone https://git.sv.gnu.org/git/vscode-texinfo.git
cd vscode-texinfo
```

Then, install dependencies with Node Package Manager, and build the project:

```bash
npm ci
npm run package
```

If the operation is successful, a `texinfo-{VERSION}.vsix` file will be generated under the root directory of the source code repository.
3 Basic Usage

There are several basic features of vscode-texinfo which can be used out-of-the-box.

3.1 Syntax Highlighting

The syntax highlighting solution is provided by this TextMate Grammar, which is originally made for Atom, and also used in GitHub Linguist.

To enable Texinfo syntax highlighting when editing a file, it should be recognized as a Texinfo document by Visual Studio Code. For file names with suffix .texi, .txi or .texinfo, this process should be automatic. If not, find and click the status bar item with “Select Language Mode” tooltip, then choose “Texinfo” in the menu which just popped up.

If syntax highlighting does not look satisfactory, try another color theme where keyword/operator colors are distinct. Some good examples are Solarized Light/Dark, Monokai, etc.

For details about how to change language mode or select color theme, see the Visual Studio Code User Guide.

3.2 Code Completion

When typing a @-command in a Texinfo document, vscode-texinfo can display a completion list so that you don’t have to type the entire command.

There are two kinds of completion items: The command itself, and code snippets related to that command.

A typical example of code snippet is the completion of a block command, say, @example. When applying completion, the generated code looks like:

```latex
@example
@end example
```

The cursor falls between `@example` and `@end example`, where you can finish the content of the `@example` block. After that, press Tab to bail out of the block.

Code snippet completion can be disabled by switching off the configuration item texinfo.completion.enableSnippets.

When code snippet completion is enabled, completion of commands which relate to code snippets is disabled by default. You can re-enable it on by switching off texinfo.completion.hideSnippetCommands.

Note: Code completion provided by vscode-texinfo does not recognize much of Texinfo’s semantics, and completion may appear in contexts where it should not exist. This is a known bug (which cannot be fixed in near future, unless a language server for Texinfo is implemented, which is not trivial).

3.3 Block Folding

Block folding allows you to collapse a block of code in a Texinfo document, so that you can navigate through the remaining part of the document more easily.
Three types of code blocks can be recognized by vscode-texinfo:

- Block commands
- Chapters, sections and subsections
- Consecutive lines of comments

While editing a Texinfo document, you can collapse or expand a code block either by clicking the folding icon to the left of the first line of the block, or by invoking a corresponding command. See the Visual Studio Code User Guide for details.

Note: Due to performance issues, the block hierarchy of a Texinfo document is re-calculated only when total line count changes, or when the document is saved.

3.4 Breadcrumb Navigation

Using the navigation bar, you can navigate through the code blocks defined in Section 3.3 [Block Folding], page 3, (excluding blocks of comment lines). Titles of chapters and sections are shown.

See the Visual Studio Code User Guide for more information about Breadcrumbs.
4 Advanced Usage

Some more advanced features of vscode-texinfo is available if GNU Texinfo is correctly installed and configured on your device.

**GNU Texinfo** is the official (and the only known) full implementation of Texinfo. On most platforms, it can be easily installed using a package manager. For example, if you’re using a Debian-based GNU/Linux distribution, you can install GNU Texinfo with:

```
sudo apt-get install texinfo
```

GNU Texinfo provides a CLI program `makeinfo`, a.k.a. `texi2any`, which converts a Texinfo document to some other format, like HTML, PDF, plain text, etc.

To specify the location of `makeinfo`, edit the configuration item `texinfo.makeinfo`. If it’s not located in `$PATH`, an absolute path should be specified. Also note that the path should not contain any command line arguments.

To check whether GNU Texinfo is correctly installed and configured, see Section 4.1 [Version Indicator], page 5.

4.1 Version Indicator

The version indicator is a status bar item with text “GNU Texinfo”. It is located on the right side of the status bar, which is shown when the active text editor contains a Texinfo document.

If you see a **check icon** and the version of GNU Texinfo, then congratulations, you’re all set. If a **cross icon** is displayed, it means that GNU Texinfo is *not* correctly installed and configured.

If a **warning icon** is displayed, it means that the currently installed GNU Texinfo is outdated, or has an unrecognizable version number. In that case, some features may not work as expected.

**Note:** The version indicator does not automatically refresh since the activation of the extension. To manually trigger a refresh, click the status bar item.

4.2 HTML Preview

You can generate the HTML preview of a Texinfo document in Visual Studio Code, to see how the document looks like when displayed online.

In the active text editor which contains a Texinfo document, click the “Show Preview” button on the top right of the editor. A webview will be created in a split editor (if not already), and the HTML preview will be displayed there. The “Show Preview” command is also available in command palette, and has a default `Ctrl+V` key binding (on GNU/Linux).

The HTML used for preview is generated by `makeinfo --html --nosplit`, and Texinfo source is read from disk, instead of taken from a `vscode.TextDocument`. The editor may prompt you to save to file when trying to display preview of a document which does not yet exist on disk, so that the preview can be displayed.

An opened preview automatically tracks the behaviour of the corresponding document. When the document is saved, the preview will refresh. When the editor holding the document is closed, the preview also closes.
Texinfo documents tend to be rather large, and makeinfo may take some time to finish. When you see an “(Updating)” in the title of a preview, it means that the preview is being updated.

### 4.2.1 Custom CSS

You can use a custom CSS to make the HTML preview look prettier. To configure this, edit the configuration option texinfo.preview.customCSS. The CSS file can either be an online or a local (starting with file://) resource.

A good example is [https://www.gnu.org/software/gnulib/manual.css](https://www.gnu.org/software/gnulib/manual.css), which is popular among manuals of GNU projects. (Note: May require some tinkering when used with darker editor themes)

### 4.2.2 Goto Node

Above each @node line in a Texinfo document, a code lens with text “Goto node in preview” is shown. On click, it makes the HTML preview jump to the corresponding location.

**Note:** @node lines allow @-commands, which is not handled in vscode-texinfo due to performance considerations. For these nodes, this feature does not work.

To disable this feature and hide the code lenses, switch off the configuration item texinfo.enableCodeLens.


### 4.2.3 Misc Settings

There are some other settings which can affect the behaviour and appearance of HTML previews.

- texinfo.preview.errorLimit: Number of errors which makeinfo can produce before quitting. (--error-limit=NUM)
- texinfo.preview.includePaths: Array of extra paths to search for @include files. (-I PATH)
- texinfo.preview.maxSize: Max allowed size for the generated HTML file before it’s displayed in the preview. Files larger than this limit will trigger an error.
- texinfo.preview.noHeaders: When enabled, headers and menus are no longer displayed in preview. (--no-headers)
- texinfo.preview.noNumberSections: When enabled, do not display chapter and section numbers in preview. (--no-number-sections)
- texinfo.preview.noValidation: When enabled, node cross-references are not validated. (--no-validate)
- texinfo.preview.variables: Array of variables to define (as with @set). If a variable has a value, use the ASCII space character to separate key and value.

### 4.3 Diagnosis

Diagnosis can help you locate syntactic and semantic problems in a Texinfo document, by parsing the error output of makeinfo and display the diagnostic information on the editor.

The diagnostic information of a document is calculated automatically, whenever an attempt to generate preview is made.
By default, all error and warning messages are shown. To disable warnings, switch on the configuration item `texinfo.diagnosis.noWarnings`.