

TKE User Guide

Version: 2.9

Author: Trevor Williams

Table of Contents

- Introduction
 - [Feature Set](#)
- Installation
 - [Dependencies](#)
 - [Installing for Linux](#)
 - [Installing for MacOS](#)
 - [Installing for Windows](#)
- Starting the Application
 - [The Command-Line](#)
 - [MacOS Desktop](#)
 - [Windows](#)
- Application Settings
 - [Setting Up a Share Directory](#)
 - [Setting Up On First Install](#)
 - [Changing Sharing Setup via Preferences GUI](#)
- User Interface
 - [Title Bar](#)
 - [Menu Bar](#)
 - [File Menu](#)
 - [Edit Menu](#)
 - [Find Menu](#)
 - [View Menu](#)
 - [Tools Menu](#)
 - [Sessions Menu](#)
 - [Plugins Menu](#)
 - [Help Menu](#)
 - [Sidebar](#)
 - [Root Directory](#)
 - [Directory](#)
 - [Files](#)
 - [Editing Pane](#)
 - [Tab Bar](#)
 - [Line Number Bar](#)
 - [Editing Buffer](#)
 - [Scrollbars](#)
 - [Bird's Eye View](#)
 - [Find Highlighting](#)
 - [Find in Files](#)
 - [Status Bar](#)
 - [Vim Mode](#)
 - [Vim Macro Recording Status](#)
 - [Cursor Position Status](#)
 - [Message Display](#)
 - [Auto-Indent Indicator](#)

- Syntax Display
- Difference Viewer
 - Line Number Bar
 - Main File Viewing Area
 - Difference Map
 - Control Panel
- Remote File Editing
 - Connection Bookmark Sidebar
 - Remote File Viewer
 - Connection Editor
 - Remote Save File Dialog
 - Main Sidebar
- Sessions
- Command Launcher
- Vim Commands
 - Standard Vim Commands
 - Extended Vim Commands
 - Tab or Text Pane Traversal
 - Marker (Bookmark) Creation
 - Multicursor Functionality
 - String/Bracket Insertion
 - Line Bubbling
 - Deletion
 - Vim Options
 - Vim Modelines
- Snippets, Emmet and Templates
 - Snippet Variables
 - Transform Format
 - Creating Snippets
 - Emmet
 - Templates
- Preferences
 - Editing Preference Information
- Menu Binding
- Theme Editor
 - Interface
 - Title Bar
 - Color Swatch Editor
 - Category Table
 - Option Detail Pane
 - Option Description Pane
 - Button Action Bar
- Plugins
 - Installing a Plugin
 - Uninstalling a Plugin
 - Showing Installed Plugin Usage Information
 - Reloading Plugins

Introduction

TKE is a source code editing environment built using Tcl/Tk which provides a clean user interface yet rich set of editing features and tools.

Feature Set

The following is a high-level list of built-in features.

- **Syntax highlighting**
 - Over 75 languages currently supported, including:
 - ABAP, ActionScript, Ada, AppleScript, Assembly, Awk
 - Bash
 - C#, C, C++, Clojure, CriticMarkup, Cobol, CoffeeScript, CSS, Cuda
 - D, Delphi, DocBook
 - Eiffel, Erlang
 - Fickle, Forth, Fortran
 - Gherkin, Go, Groovy
 - HTML, Haml, Haskell, Haxe, HelpSystem
 - J, JSON, Jade, Java, JavaScript, Julia
 - Lex, Lisp, Lua
 - MATLAB, Makefile, ManPage, Markdown, MultiMarkdown, MySQL
 - Neko
 - Objective-C, OCaml
 - PHP, Pascal, Perl, Prolog, Python
 - R, RSS, Ruby, Rust
 - SASS, SCSS, Scala, Scheme, ShellScript, SQL, Swift, SystemVerilog
 - Taccle, Tcl
 - Vala, Verilog, VisualBasic, VHDL
 - XML, XSLT
 - YAML, Yacc
 - Syntax description files can be easily added
 - Preference control can select syntax types to highlight/not highlight
 - Built-in and customizable color themes
 - Support for mixed and embedded language syntax
- **Multi-cursor support**
 - Cursor alignment
 - Enumeration insertion
- **Built-in file difference viewer**
 - Built-in support for Perforce, Git, Mercurial, Bazaar, Subversion and file diff. Also supports a custom difference command. Includes ability to jump to the file version where a given line was last changed.
- **Internationalization support**
 - 17 languages currently supported
- **Customizable menu bindings**

- **Clipboard history**
- **Unlimited undo/redo**
- **Language-specific snippet support**
 - Support for tab stops, variable substitution, and special value substitutions.
- **Built-in Emmet abbreviation support**
 - HTML, XML, XSL and CSS abbreviation syntax supported.
 - Support for custom Emmet commands.
- **Auto and smart indentation features**
 - Selected code can have indentation policies applied
 - Pasted code can have indentation policies applied
- **Code line marking support**
- **Line number display**
 - Includes support for both absolute and relative line numbering
- **Code folding**
 - Based on indentation syntax markers found in the code or manually added (modal).
- **Customizable tab and shift/indentation stops**
- **Dozens of text transformation tools**
- **Built-in Vim support**
- **Expanded Vim functionality**
 - Vim commands for setting/clearing multiple cursors
 - Auto-numbering functionality
 - Line bubbling
- **Regular expression in-file search (and optional replace)**
 - Find and replace
 - Includes ability to quickly save and recall search input
- **Regular expression multi-directory file search**
 - Includes ability to quickly save and recall search input
- **Symbol search function**
 - Jump to a named procedure or function call
- **Auto refresh**
 - Files modified outside of editor will be automatically updated (unless file is in a modified state)
- **File locking support**
 - File can be set to be read-only within the editor (regardless of actual file permissions)
- **Command launcher**
- **File system sidebar**
 - Contains functionality for creating, renaming, deleting, and trashing files/directories
- **Multiple files can be opened at once**
 - Sidebar and tabs used to switch between opened files
- **Dual editor panel support**
 - Useful for viewing two files side-by-side
 - Line alignment
 - Synchronized pane scrolling support
- **Split view support**
 - Create two independent editable views into the same file.
- **Bird's Eye text viewer**

- See your file's code at a glance and quickly navigate to other points in your code.
- **Maximum column width display**
- **Support for NFS mounted file systems**
- **Plugin support**
 - Full plugin development documentation available.
- **Customizable theme and built-in theme editor**
 - Edit existing themes
 - Create new themes
 - Import TKE and TextMate themes
 - Export TKE themes that can be shared with others"
- **Favorite file/directory support**
- **Automatic matching character insertion**
 - Curly bracket, square bracket, angled bracket, parenthesis, double and single string character matches are inserted as you type.
 - Preference item to enable/disable any of the above character types.
 - Each language syntax file specifies which characters are valid for auto-insertion.
- **Multi-platform EOL character support**
 - Supports CR, CRLF and LF characters for end-of-line indicators. Also supports an 'orig' option which automatically infers the original EOL character when the file is read and an 'auto' option which infers the correct EOL character for the current platform. This setting is available through the preference system.
- **Session support**
 - Allows for the creation and switching of customized TKE sessions.
 - Saves the following reloadable information:
 - Opened files/directories
 - UI state information
 - Custom preferences
 - Saved search items
- **Templates**
 - Special snippet-like files that can be used for quickly generating new files. TKE allows for saving, using and managing template files.
- **Automatic session save**
- **In-app update mechanism**
 - Preference option to follow stable or development release track (not available for Windows).
- **Remote file editing**
 - FTP/SFTP/WebDAV supported
 - View and edit remote files
 - Duplicate, rename and delete remote files/folders via the sidebar
- **Tab hiding**
 - Temporary hide tabs in the tab bar for improved workflow.

Installation

Dependencies

The installation of TKE has a few dependencies that will need to be preinstalled before the application can begin to work. The dependencies are listed in the table below along with the URL path to find the source packages. The installation of these packages is outside the scope of this document. Please refer to each packages installation notes for this information.

Package	Download URL
Tcl (8.5.x or 8.6.x versions)	http://sourceforge.net/projects/tcl/files/Tcl/
Tk (8.5.x or 8.6.x versions)	http://sourceforge.net/projects/tcl/files/Tcl/
Tcllib	https://sourceforge.net/projects/tcllib/files/tcllib/
Tklib	https://sourceforge.net/projects/tcllib/files/tklib/
Extended Tcl (Library should be installed in one of the standard Tcl paths)	http://sourceforge.net/projects/tclx/files/TclX/
Tkdnd (optional)	https://sourceforge.net/projects/tkdnd/files/?source=navbar
Expect (optional - used to help provide SFTP support - not necessary for macOS)	https://sourceforge.net/projects/expect/files/
PuTTY PSFTP client (Windows only, optional - used for SFTP support)	http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Important Note: Tcl/Tk version 8.5.19 or later is recommended to avoid application crashes during editing.

All other Tcl/Tk packages required by TKE have been bundled in the TKE package.

Installing for Linux

Prior to downloading/installing the TKE package, you will need to make sure that you have all of the required packages installed on your system. Because various Linux distributions have different package managers, I will leave the exact details of how to accomplish this up to you. However, if you have an Ubuntu-based distribution, you can get the needed packages by performing the following command:

```
sudo apt-get install tcl8.5 tk8.5 tclx8.4 tcllib tklib tkdnd expect
OR
sudo apt-get install tcl8.6 tk8.6 tclx8.4 tcllib tklib tkdnd expect
```

The TKE installation package is downloaded in a gzipped tarball. You can get the latest version of this tarball from the following URL: [TKE Download](#).

Select a tarball (i.e., *.tar.gz file) to download within this page and save the resulting tarball into a temporary directory. After the download has completed, unzip and untar the file using the given command:

```
gzip -dc <tarball_filename> | tar xvf -
```

After the tke directory has been untarred, you can delete the original tarball using the following command:

```
rm -rf <tarball_filename>
```

After all of the files have been uncompressed, change the working directory to the resulting “tke-X.X” directory using the following command:

```
cd tke-X.X
```

Once inside the TKE source directory, run the installation script found in that directory using the following command:

```
tclsh8.5 install.tcl
```

OR

```
tclsh8.6 install.tcl
```

At the beginning of the installation process, the install script will check to make sure that you have both Tcl and Tk 8.5 installed along with a usable version of TclX. If all checks are good, the installation will continue; otherwise, it will provide an error message indicating the offending check. After the checks occur, you will be asked to provide a root directory to install both the TKE library directories/files and the TKE binary file. This can be any directory in your filesystem; however, popular directories are:

- /usr/local
- /usr

After specifying a file system directory, TKE will indicate the names of the directory and binary file that it will install. If everything looks okay, answer “Y” or “y” (or just hit the RETURN key); otherwise, hit the “N” or “n” keys to enter a different directory. Once you enter a directory, the installation script will check to see if a previous version of TKE has been installed at that directory location. If one is found, it will ask if you would like to replace the old version with the new version. Hit the “Y” or “y” key (or just hit the RETURN key) to confirm the replacement. To cancel the installation and select a new directory, hit the “N” or “n” key. If you have specified that the given directory should be replaced (or no replacement was necessary), the script will continue with the full installation. At any time you can quit the installation script by entering the CONTROL-c key combination.

Installing for MacOS

If you only plan on running tke from a terminal and are satisfied with running the application through the X11 server that runs on Mac, you can follow the same installation steps that is used for Linux-based systems. However, if you would like to install TKE like a native Mac OS X application (i.e., application available in the Applications folder, TKE icon displayed in the dock, etc.), follow these installation steps.

After downloading the TKE disk image into the Downloads folder, double-click the disk image file and then drag and drop the TKE application icon in the resulting window to the Applications directory.

Important Note: The wish shell that is used is based on Cocoa and, as such, for Mac OS X versions 10.7 (Lion) and later have a feature that stops certain keys from being automatically repeated when its key is held down. This will make Vim-mode on these systems from working as expected. To disable this on your system, enter the following command within the Terminal application prior to starting TKE:

```
defaults write -g ApplePressAndHoldEnabled -bool false
```

Installing for Windows

The easiest installation process for Windows is fairly straightforward and creates a native Windows application on your machine. Download the TKE Windows executable installer from the SourceForge website, run the resulting download file, and follow the installation wizard steps. The application will then be available through the start window and, if enabled in the installation process, through a desktop shortcut.

You can alternatively install a Unix-like environment such as Cygwin and then install the tarball in a similar manner to installing for Linux (with the exception that Cygwin does not have a software update tool like 'apt-get' but rather maintains its own software packages available through the Cygwin installer). The process of installing Cygwin and configuring its environment properly for TKE is beyond the scope of this document.

It is important to note that on Windows, the in-app update mechanism is not available. Only stable releases of TKE will be available and only from the SourceForge website. Updating the application will require downloading and running the new installer.

Starting the Application

After TKE has been installed on your system, there are a variety of ways to start the application, depending on your usage.

The Command-Line

For Unix-based systems that support a terminal, you can invoke TKE using the command-line. To make tke easier to use, it is recommended that you add the TKE installation's bin directory into your environment path variable (see your shell's documentation for how to do this as this will be different for different OS types as well as shells).

Next, if you want TKE to always use just one window for editing all files, make sure that your xhost is setup correctly. If you get a new TKE window every time you open a file in the terminal, it is likely that you have an xhost issue.

Assuming that you have added the TKE installation bin directory to your path, invoking TKE is as simple as typing the following at the shell prompt:

```
tke
```

If this is the first time that the application has been started, this will create a single TKE window with no tabs opened and an empty sidebar. If the application is not currently running, this will start the application and load the last TKE session information into the application, including the following information:

- Window dimensions and location
- Previously opened files when the application last exited
- Sidebar entries
- Current tab of previous session will be the current tab of this session

If TKE is already running, this command will simply bring the application to the foreground of the desktop.

This, however, is not the only way of starting the application from the command-line, you can also specify any number of directories and/or files as arguments to TKE. Any directories specified will be added to the sidebar while any specified files will be opened in new tabs in the editor and their respective directories will be added to the sidebar (if they don't already exist).

In addition to files and directories, the following options are also available on the command-line invocation.

Option	Description
-h	Displays command-line usage information to standard output and exits immediately.
-v	Displays tool version information to standard output and exits immediately.
-nosb	Starts the UI without the sidebar being displayed.
-e	Exits the application when the last tab is closed (overrides preference setting)
-m	Creates a minimal editing environment (overrides preference settings)
-n	Opens a new window without attempting to merge with an existing window.
-s <i>session_name</i>	Starts a new session with (if a window does not exist) or switches the current window to a previously saved session specified with <i>session_name</i> .

MacOS Desktop

On Mac OS X, the application installation will place the TKE application in the Applications directory. To launch the application, simply open the Application directory in the Finder and double-click on the application. This will launch TKE using the same session settings that TKE had when last launched.

You can also start the application using any other method available on your Mac OS X system, including Launchpad, Spotlight, third-party application launchers, etc.

If you want to execute tke from the Terminal on the command-line, you can do so by executing the `/Applications/TKE.app/Contents/bin/tke` script. The command-line options can then be used when invoking TKE.

Windows

On Windows, the application installation will place TKE in its own subfolder within the Programs directory available through the Start menu. To launch the application, simply traverse in the program hierarchy to the application icon and double-click it. This will launch TKE using the same session settings that TKE had when last launched.

Like its Mac OS X counterpart, you can also start the application using any other method available methods normally used for starting an application.

Application Settings

There are a number of files/directories that are created by TKE at application startup, during execution and at application exit. The following is a list of this application data:

- Emmet aliases
- Favorited files/directories
- Command launcher data
- Preferences
- Installed plugins
- Saved sessions data and preferences
- User and language snippets
- Saved file templates
- Imported themes
- Remote file settings

These files have historically been saved in the user's home directory in the `.tke` directory. As of version 2.7, TKE now has the ability to save these files in either the `~/.tke` directory or another directory, called from this point on as the "share directory", which can be any directory accessible from the file system (but preferably in an automatically sync'ed directory like Dropbox, iCloud Drive, Google Drive, OneDrive, etc.) Placing files in a shared directory allows the application files to be shared between different computers which can access the shared directory. Additionally, if any computer setup to use the shared directory makes a change to this directory, other computers will see these changes when they are started.

In addition to specifying a share directory, you can also specify where each of the above items will be stored. If an items files/directories are stored in the user's home directory, any updates to those files will remain local to the machine and will not be seen by machines who are using the share directory. However, if an item is being shared then any changes made to that item will be seen by other sharers.

Setting Up a Share Directory

There are two ways the user can specify whether or not to use a share directory, specify its location and specify which items will be shared:

1. The first time that TKE is started after installation.
2. Using the Preferences GUI General / Sharing tab.

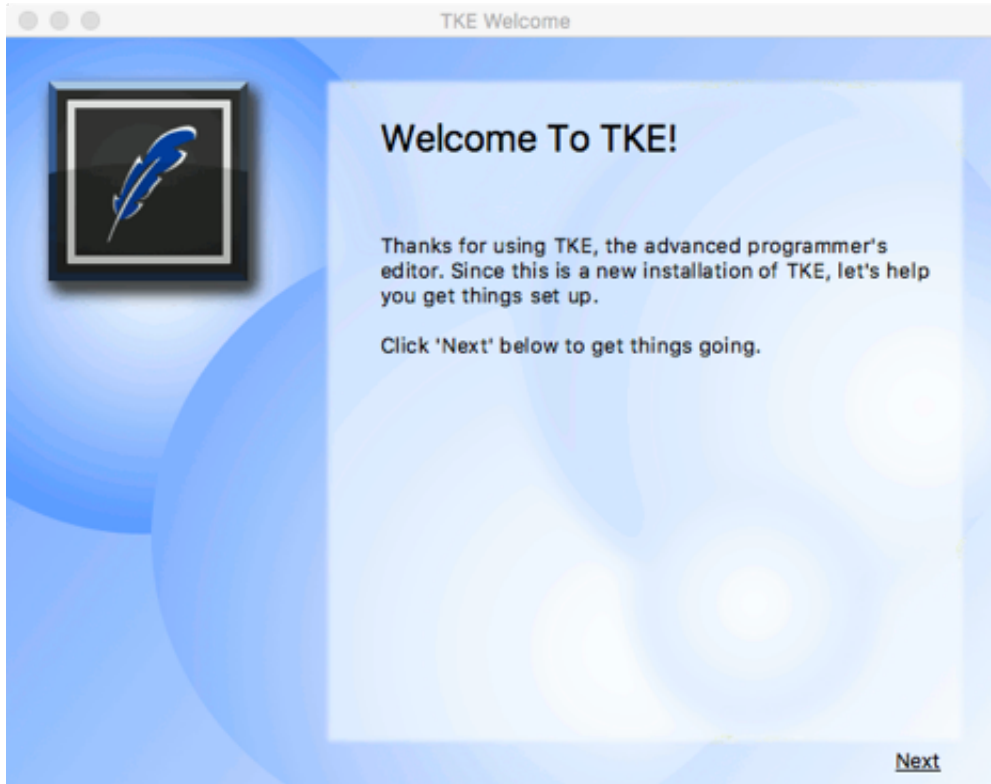
Let's take a look at each of these setup options.

Setting Up On First Install

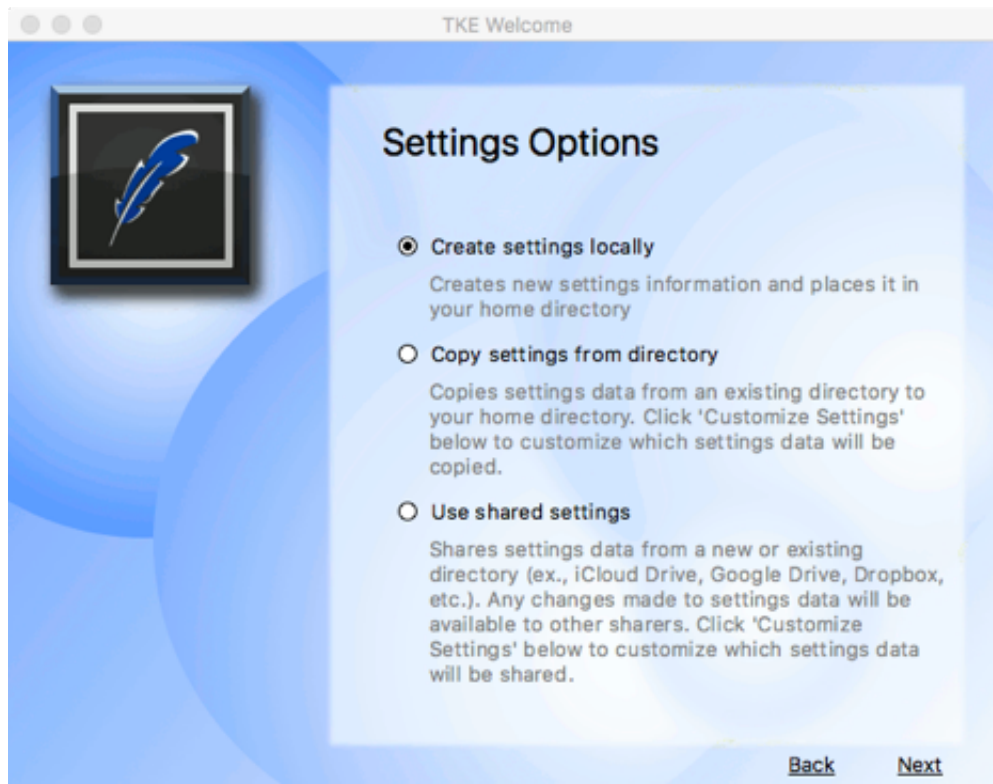
After TKE is installed on a system, if the `.tke` directory in the user's home directory does not exist, TKE will display a first start wizard to aid the user in setting up the location of the application data prior to starting the application. The wizard is made up of four screens:

- Welcome screen
- Application data directory selector
- Share/Import directory and item selector
- Finish screen

These screens are shown below:



The Welcome Screen



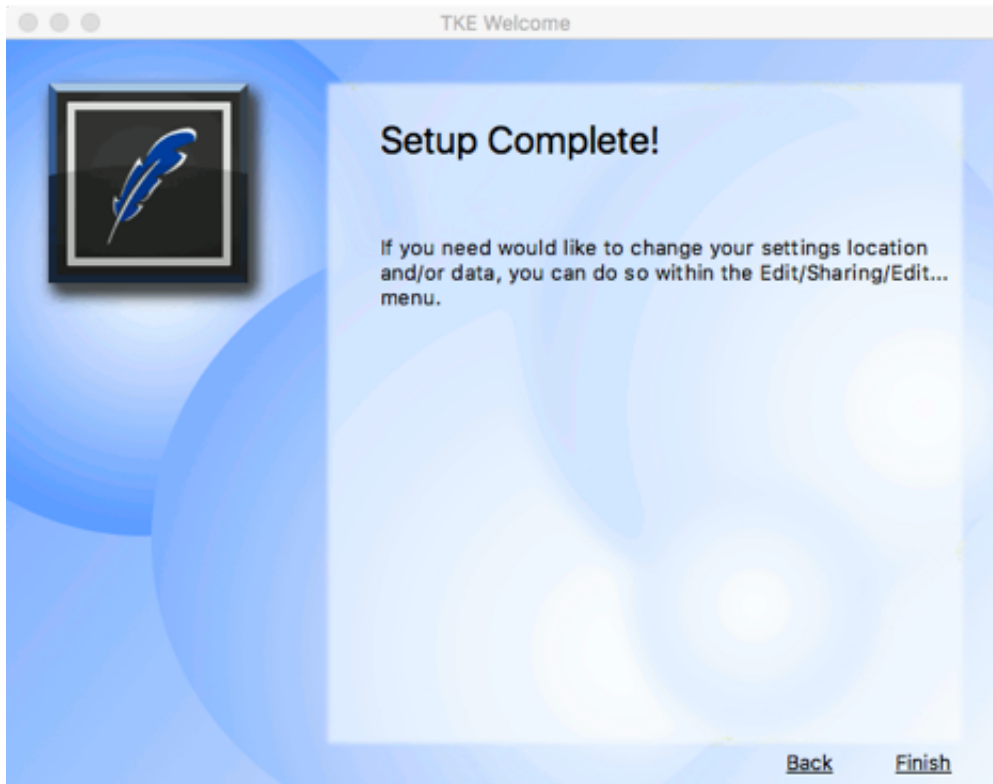
The Application Data Directory Selection Screen

This screen allows the user to specify where the application data should be stored. If the “Create settings locally” option is selected all application data will be stored in the `~/tke` home directory (i.e., no items will be shared). If the “Copy settings from directory” option is selected, the user will be asked for the location of a directory containing existing application data which will be copied to the user’s `~/tke` home directory. If the “Use shared settings” option is selected, the user will be asked to specify a directory location to share application data. After selecting an option, click the “Next” button to continue.



Share/Import Directory and Item Selector Screen

This screen will display the selected directory if either the import or share options were selected on the previous screen. You can change the directory by clicking on the “Change Directory” button. At the bottom of the screen is the list of application data items to import/share. If the user is doing an import, all selected items in the list will be copied to the user’s .tke home directory (if that application data is available in the selected directory). If the user is sharing data, all selected items will be shared with other computers in the specified directory. All unselected items will be placed in the user’s .tke home directory. Click “Next” after all items are filled out as desired. Click the “Back” button to return to the previous screen.



Setup Completion Screen

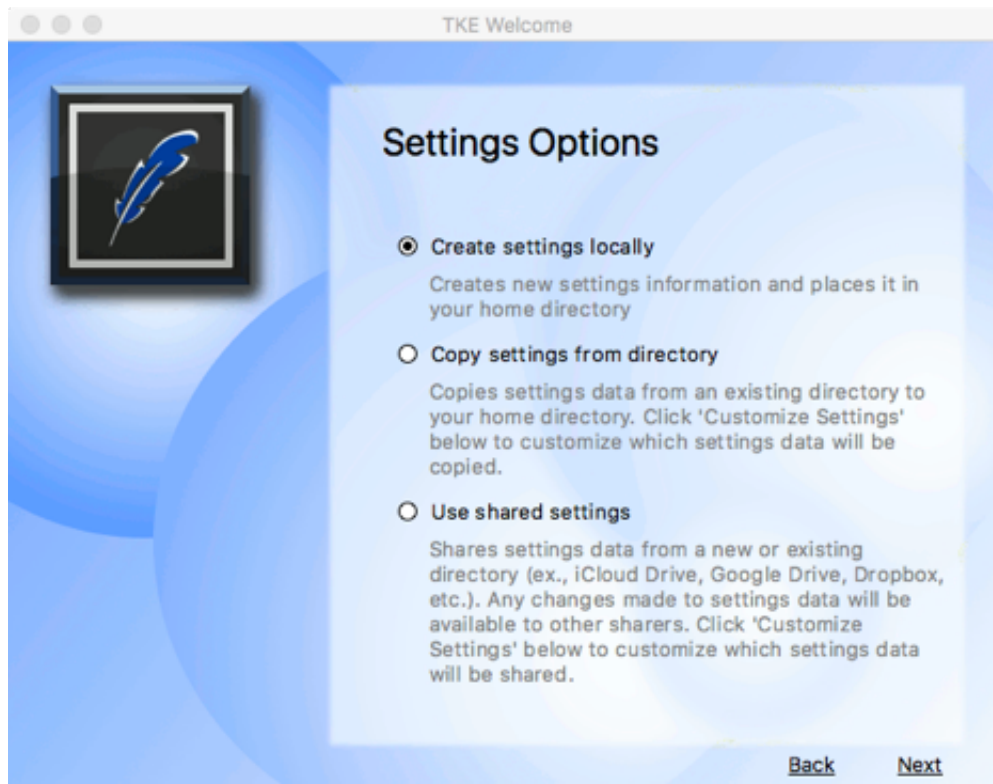
This screen just reminds the user that they can always change their settings from with TKE. Click on the “Finish” button will perform the application data setup and will start the application for use. Click on the “Back” button to return to the previous screen.

Changing Sharing Setup via Preferences GUI

Application data sharing options can be changed at any time within TKE using the preferences GUI. Start the preferences GUI using the “Edit / Preferences / Edit User - Global” menu option. This will display the preferences GUI. Make sure the “General” panel is selected on the right side of the preferences window and select the “Sharing” tab in the options panel. The following image is a depiction of this preference panel:

The Sharing panel displays the location of the share directory. If the field is empty, the checkbox will be unchecked. If the share directory is cleared, the application data is being stored in the user’s local .tke directory within their home directory and the selected items in the “Sharing Items” list are unused. If the shared directory is specified, the “Sharing Items” list specifies which application data are stored in the shared directory (selected) and which are stored locally (unselected).

If the shared directory is empty and you would like to cause your application data to be stored in a shared directory, click on the checkbox to the right of the directory name. This will display a directory chooser window. If a directory is selected, its pathname will be displayed in the directory field. When the preferences window is closed, the sharing changes will be applied.



If the sharing directory was changed from local to shared, TKE will copy any local files/directories that were selected for sharing to the shared directory if those files/directories do not currently exist in the shared directory. If the sharing directory was changed from shared to local, TKE will copy the selected shared items from the sharing directory to the local directory, overriding any files/directories in the local directory.

User Interface

By default, the editing environment consists of two panels: a file/directory sidebar and the tab-controlled editing buffer itself. As much as it possible, busy and redundant UI elements are removed from the screen when they are not in use. Most of the UI is only displayed as needed when the user calls up its functionality.



Title Bar

Within the title bar at the top of the window is the base name of the file currently being edited and the name of the current working directory. All commands that deal with the file system will be relative to this working directory.

Menu Bar

Below the title bar is the menu bar (on Windows and most Linux distributions). This contains a list of many of the available features within the tool. Any functionality that is contained within the listed menus can be assigned a keyboard shortcut, configurable via the menu binding file (see the “Menu Binding” Chapter for more details on the structure of this file). From left to right, the main menus are as follows:

- File
- Edit
- Find
- View
- Tools

- Sessions
- Plugins
- Help

File Menu

The File Menu contains commands that are related to either the currently selected file (i.e., the file in view within the editor which has the keyboard focus) or all files. The following table describes the listed menu items and their associated functionality

Menu Item	Description
New Window	Launches a new TKE session window.
New File	Creates a new, unnamed file in a new tab.
New From Template...	Creates a new file based on a previously saved template file.
Open File...	Displays an open file dialog, allowing the user to select one or more files to open. Each file will be opened in a separate tab in the editor. Any directories containing these files that are not in the sidebar will be added to the sidebar.
Open Directory...	Displays an open directory dialog, allowing the user to select one or more directories to add to the sidebar.
Open Remote...	Displays the remote file/directory dialog window which will allow you to connect to a remote file server via FTP/SFTP and select either a directory or a file to open.
Open Recent	Displays a list of directories and files that have been recently opened. Click on a directory to add it to the sidebar. Click on a file to open it in a separate tab in the editor.
Open Favorite	Displays the list of favorited files/directories for quick opening in either the editor (file) or sidebar (directory).
Reopen File	Reopens the current file for editing, destroying any unsaved changes.
Change Working Directory	Changes the current working directory.
Show File Difference	Displays a new tab containing the current file in a difference view. From within the view, the user can view file differences between any two versions of the given file (if managed by a version system) or between it and another file (using the diff utility).
Save	Saves the contents of the current file to its original name. If an original name does not exist for the content, a “Save As” dialog will be displayed allowing the user to specify a file name.
Save As...	Displays a save file dialog window, allowing the user to save the current file contents to the

	given filename. The original filename of the content will be changed to this new name.
Save As Remote...	Displays the remote file/directory dialog window to allow you to select a server, directory and filename to save the contents of the current editing buffer to via FTP/SFTP/WebDAV.
Save As Template...	Displays an entry field at the bottom of the window, allowing the current file to be saved as a template file under the given name. If you specify an extension to the template name, any new files based on this template will use the template's extension for syntax highlighting purposes.
Save Selection As...	Saves only the currently selected text to a file without saving the current editing buffer.
Save All	Saves all files opened in the editor to their original file names. Any files which do not have original names, will have a save file dialog window shown, allowing the user to specify the name.
Export...	If the current editing buffer language is set to Markdown, this option will run the file contents through the Markdown parser, generating either HTML or XHTML output (the General/DefaultMarkdownExportExtension specifies the default extension to use for Markdown exporting but can be overridden in the save file dialog window). Additionally, the user can embed snippet text within <code><tke:ExportString></tke:ExportString></code> tags. Any snippets will be generated at the time of export, replacing the tags and snippet text with the generated text.
Line Ending / Windows	Changes the line ending to use for the current file to CRLF when the file is saved.
Line Ending / Unix	Changes the line ending to use for the current file to LF when the file is saved.
Line Ending / Classic Mac	Changes the line ending to use for the current file to CR when the file is saved.
Rename	Renames and/or moves the location of the current file in the file system.
Duplicate	Creates a copy of the current file and immediately opens the file for editing.
Delete	Deletes the current file from the file system and removes the tab from the editor. This option will be displayed if the "Use Move to Trash" general preference option is unset.
Move To Trash	Moves the current file to the trash. This option will be displayed if the "Use Move to Trash" general preference option is set.
Lock/Unlock	The "Lock" option will change the state of the editor to not allow text modifications to the window (content is effectively "Read Only"). A small lock icon will be displayed in the associated tab to indicate that the file content is currently "locked". The "Unlock" option will change the state of the editor back to the modifiable state.
Favorite/Unfavorite	Marks the current file as a favorite (with the "Favorite" command) or removes the file as a favorite (with the "Unfavorite" command). Favorited files can be opened quickly with the

	“Open Favorite” menu list or the command launcher. Additionally, favorited files/directories can be used in the “Find in File” feature.
Close	Closes the current tab. If the text content is in the modified state (as indicated by the “*” character in the tab), a prompt will be displayed asking the user if the content should be saved prior to closing.
Close All	Closes all tabs in the editor. If text content in a tab has been modified, a prompt will be displayed asking the user if the content should be saved prior to closing.
Quit	Exits the application. Any modified files in the editor will prompt the user if the content should be saved prior to exiting the application.

Edit Menu

The Edit menu contains menu items that affect the contents within the current file. The following table describes the items available within this menu.

Menu Item	Description
Undo	Undoes the last change made to the file content. Each file can have an unlimited number of items that can be undone. Saving a file clears the undo stack for that file.
Redo	Re-applies the last undone change made to the file content. Saving a file clears the redo stack for that file.
Cut	Deletes the selected text, copying the deleted content to the clipboard. If no text is currently selected, the current line is deleted and sent to the clipboard.
Copy	Copies the selected text to the clipboard. If no text is currently selected, the current line is copied to the clipboard.
Paste	Pastes the content in the clipboard, inserting the text before the insertion cursor. The content is copied “as is”.
Paste and Format	Pastes the content in the clipboard, inserting the text before the insertion cursor. The content is indented to fit into the current insertion point.
Select All	Selects all of the text in the current editor.
Vim Mode	When selected, changes the editing environment to use Vim-style interaction. When deselected, changes the editing environment back to “normal” editing mode.
Toggle Comment	Detects the comment state of the current selection. If the selected text is not commented out, places a line comment in front of any selected text in the current file. If the selected text is commented out, the comments are removed from the selected lines. If a selection does not exist, the current line (or lines, if multicursors are enabled) is commented/uncommented in a similar fashion.

Indentation / Indent	Indents the selected text by one level of indentation.
Indentation / Unindent	Unindents the selected text by one level of indentation.
Indentation / Format Text	Modifies either the selected text or the entire file content (depending on whether text is currently selected or not) to match the indentation in the current context.
Indentation / Indent Off	Turns indentation mode off for the current editor. Hitting the ENTER key in the editing window will place the cursor in the first column of the next row.
Indentation / Auto-Indent	Turns auto-indentation mode on for the current editor. Hitting the ENTER key in the editing window will place the cursor in the same column as the previous line's starting character.
Indentation / Smart Indent	Turns smart indentation mode on for the current editor. Hitting the ENTER key in the editing window will perform the proper indentation based on the current language and context. If a character sequence is entered that completes an indentation, the character sequence will be adjusted to the proper indentation level.
Cursor / Move to First Line	Moves the cursor to the start of the first line of the file and adjusts the view so the cursor is visible.
Cursor / Move to Last Line	Moves the cursor to the start of the last line of the file and adjusts the view so the cursor is visible.
Cursor / Move to Next Page	Moves the cursor down by a single page and adjusts the view so the cursor is visible.
Cursor / Move to Previous Page	Moves the cursor up by a single page and adjusts the view so the cursor is visible.
Cursor / Move to Screen Top	Moves the cursor to the start of the line at the top of the current screen.
Cursor / Move to Screen Middle	Moves the cursor to the start of the line in the middle of the current screen.
Cursor / Move to Screen	Moves the cursor to the start of the line at the bottom of the current screen.

Bottom	
Cursor / Move to Line Start	Moves the cursor to the start of the current line.
Cursor / Move to Line End	Moves the cursor to the end of the current line.
Cursor / Move to Next Word	Moves the cursor to the beginning of the next word.
Cursor / Move to Previous Word	Moves the cursor to the beginning of the previous word.
Cursor / Move Cursors Up	In multicursor mode, moves all of the cursors up by one line.
Cursor / Move Cursors Down	In multicursor mode, moves all of the cursors down by one line.
Cursor / Move Cursors Left	In multicursor mode, moves all of the cursors to the left by one character.
Cursor / Move Cursors Right	In multicursor mode, moves all of the cursors to the right by one character.
Cursor / Align Cursors	When multicursors are set in the current file, this command will adjust each line such that all cursors will be aligned to the same column. The cursors will be aligned to the highest column in the multicursor set.
Insert / Line Above Current	Inserts a blank line above the current line and places the cursor at the beginning of the blank line for editing.
Insert / Line Below Current	Inserts a blank line below the current line and places the cursor at the beginning of the blank line for editing.

Insert / File Contents	Prompts the user to select a file for insertion. If a file is selected, the entire contents of the file are inserted the line below the current line.
Insert / Command Result	Prompts the user to input a shell command. If a legal shell command is entered, the result of the command is inserted below the current line.
Insert / From Clipboard	Displays the command launcher in clipboard mode to allow the user to view and select one of the clipboard history elements to insert into the current editor.
Insert / Snippet	Displays the command launcher in snippet mode to allow the user to view and select one of the language-specific snippets to insert into the current editor.
Insert / Enumeration	When one or more multicursors are set, allows the user to insert ascending numerical values at each cursor insertion position.
Delete / Current Line	Deletes the current line and places the cursor at the beginning of the next line. The deleted line is placed into the clipboard.
Delete / Current Word	Deletes the current word and places the cursor at the beginning of the next word. The deleted word is placed into the clipboard.
Delete / Current Number	Deletes the current number and places the cursor just after the deleted text. The deleted number is placed into the clipboard.
Delete / Cursor to Line End	Deletes all characters between the current cursor and the end of the line, placing the cursor on the character previous to the current character.
Delete / Cursor to Line Start	Deletes all characters between the start of the current line and up to (but not including) the current cursor.
Delete / Whitespace Forward	Deletes all consecutive whitespace (i.e., space and tab) characters from the current cursor towards the end of the current line.
Delete / Whitespace Backward	Deletes all consecutive whitespace characters from the current cursor towards the start of the current line.
Delete / Text Between Character	Displays an input field allowing a single character to be entered. The character is searched for the first occurrence before the current cursor and the first occurrence after the current cursor. All characters between these two characters is deleted and placed in the clipboard.
Transform / Toggle Case	Toggles the case of the character at the current insertion cursor or of all selected characters.
Transform /	Sets the case of the character at the current cursor or all selected characters to lower case.

Lower Case	
Transform / Upper Case	Sets the case of the character at the current cursor or all selected characters to upper case.
Transform / Title Case	Sets the case of the character at the current cursor or all selected characters such that the first character of each word is capitalized while all other characters are placed into lower case.
Transform / Join Lines	If multiple lines are selected, joins all lines containing a selection are joined with a single space character into one line. If no lines are selected, the line below the current line is joined to the current line.
Transform / Bubble Up	If multiple lines are selected, all selected lines are moved up by one line (the line above will be moved below the bubbled line(s)); otherwise, the current line is bubbled up one line.
Transform / Bubble Down	If multiple lines are selected, all selected lines are moved down by one line (the line below will be moved above the bubbled line(s)); otherwise, the current line is bubbled down by one line.
Transform / Replace Line With Script	If the current line contains an executable shell command, the command is executed and the resulting output replaces the current line.
Preferences / Edit User - Global	Displays the user's global (cross-language) preferences in an editor tab. Saving changes made to this tab will immediately update the environment without restarting.
Preferences / Edit User - Language	Displays the user's current language preferences in an editor tab. Saving changes made to this tab will immediately update the environment without restarting.
Preferences / Edit Session - Global	Displays the current session's global (cross-language) preferences in an editor tab. This option will only be available if a named session is currently opened (see Session menu for details). Saving changes made to this tab will immediately update the environment without restarting.
Preferences / Edit Session - Language	Displays the current session's current language preferences in an editor tab. This option will only be available if a named session is currently opened. Saving changes made to this tab will immediately update the environment without restarting.
Snippets / Edit User	Adds the user's global snippet file into the editor.
Snippets / Edit Language	Adds the user's snippet file into the editor for the current language.
Snippets / Reload	Reloads the contents of the snippets for the current language and user. Useful if the snippet file contents are not usable within the editor.
Templates / Edit	Opens an existing named template for editing.

Templates / Delete	Deletes an existing named template.
Templates / Reload	Reloads the names of the existing templates.
Emmet / Expand Abbreviation	Expands the Emmet abbreviation syntax that is found to the left of the cursor (i.e., cursor must be placed on the right side of the abbreviation for proper expansion to occur).
Emmet / Edit Custom Abbreviations	Displays the custom Emmet abbreviation file in a new editing buffer allowing the user to change, remove or add custom Emmet syntax to their liking. Saving the editing buffer will cause the file changes to go into effect immediately.

Find Menu

The Find menu contains items for searching and, optionally, replacing text in the current file. It also contains items that can add search text to the current selection and items for finding text in a group of files (regardless if they are currently opened in the editor or not). The following table contains the items found in this menu along with the description of its functionality.

Menu Item	Description
Find	Searches the current file for a given regular expression. The displayed search bar also contains a checkbox for specifying whether a case sensitive search should be performed or not and a checkbox for saving the search input. Using the up/down keys while the input is in the entry field will allow you to traverse the find history and previously saved searches. Hitting the return key will cause all matches in the current file to be highlighted, the first match after the current cursor to be in view, and the cursor placed at the beginning of the match.
Find and Replace	Searches the current file for a given regular expression and replaces it with an associated string. The displayed search and replace bar also contains three checkboxes: one for specifying case sensitivity of the match, one for replacing the first match or all matches, and one for saving the search input. Using the up/down keys will traverse Find/Replace history and previously saved searches. Hitting the return key will perform the replacement.
Select Next Occurrence	Selects the next matched occurrence.
Select Previous Occurrence	Selects the previous matched occurrence.
Select All Occurrences	Selects all matched occurrences.
Append Next	

Occurrence	Adds the next matched occurrence to the selection.
Jump Backward	Jumps to the last cursor position that was more than 2 lines from the current cursor position. The number of minimum lines can be adjusted in the preferences file.
Jump Forward	Jumps to the next cursor position.
Jump To Line	Displays a user input interface that allows the user to specify a line number to jump to. Sets the cursor to the given line number and makes the insertion cursor visible.
Next Difference	If the current buffer is in difference mode, jumps to the next difference that is not currently in view. If no difference exists below the current view, jumps to the first difference in the file.
Previous Difference	If the current buffer is in difference mode, jumps to the previous difference that is not currently in view. If no difference exists above the current view, jumps to the last difference in the file.
Show Selected Line Change	If the current buffer is in difference mode and a line is currently selected, sets the first file version to the version that last modified the first line of the selection.
Markers / Create at Current Line	Sets a marker at the current insertion index.
Markers / Remove From Current Line	Clears the marker at the current insertion index if one exists.
Markers / Remove All Markers	Clears all markers in the current buffer.
Markers / <i>marker_name</i>	Jumps the cursor and file view to show the selected marker. The cursor will be placed at the beginning of the marked line.
Find Matching Bracket	Jumps the cursor and file view to show the parenthesis, bracket or quotation mark that matches the parenthesis, bracket or quotation mark under the current cursor. The cursor will be placed on the matched pair. If the cursor is currently not on a parenthesis, bracket or quotation mark, this option will set the cursor to the previous indentation character (if one exists for the current language).
Find Next Bracket Mismatch	If there is a bracket mismatch after the current insertion cursor, selecting this option will place the insertion cursor on the mismatching curly bracket, square bracket, parenthesis or angled bracket. To enable bracket mismatch highlighting, enable the Editor/HighlightMismatchingChars preference item.
	If there is a bracket mismatch before the current insertion cursor, selecting this option will place

Find Previous Bracket Mismatch	the insertion cursor on the mismatching curly bracket, square bracket, parenthesis or angled bracket. To enable bracket mismatch highlighting, enable the Editor/HighlightMismatchingChars preference item.
Find In Files	Performs a regular expression search in a specified list of files/directories. The resulting list of matches are displayed in a read-only editing buffer allowing you to jump to matches by clicking on a match result.

View Menu

The View menu allows the user to change the interface as desired. The following table lists the available menu items.

Menu Item	Description
Show/Hide Sidebar	Shows or hides the sidebar panel.
Show/Hide Console	For operating systems that allow a Tcl/Tk console to be viewed, shows/hides this console window from view. This menu item is only displayed if a console is available. The console is mostly useful for TKE development purposes only.
Show/Hide Tab Bar	Shows or hides the tab bar.
Show/Hide Status Bar	Shows or hides the status bar at the bottom of the window.
Show/Hide Line Numbers	Shows or hides the line numbers in the current buffer.
Line Numbering / Absolute	Displays line numbers starting at 1 and incrementing by one to the end of the file.
Line Numbering / Relative	Displays the current line number as 0 and counts up above and below the current line.
Show/Hide Marker Map	Shows or hides the marker map in the text scrollbar region.
Show/Hide Meta Characters	Shows or hides any characters in the current edit tab that are syntax highlighted as “meta” characters. Examples of meta characters would be formatting characters used in languages like Markdown.
Display Text	Displays the current line count and character count for the current file in the information bar.

Info	
Split View	When selected, creates a second view into the current file. Each view can be independently manipulated; however, any text modifications made in either window will be available in the other view. Deselecting this menu option will return the file to only showing a single view of the file in the editor.
Bird's Eye View	When selected, displays the bird's eye view within each opened tab. Deselecting this option will hide the bird's eye view in all opened tabs.
Move to Other Pane	Moves the current file to the other text pane. If only one text pane is currently viewable, a second pane will be displayed to the right of the current pane and the file will be moved to that pane. If a pane only contains the file that is being moved, that pane will be removed from view. This allows two files to be viewed "side by side".
Panes / Enable Synchronized Scrolling	When selected, synchronizes the scrolling of both panes to keep the displayed lines in alignment with one another when either pane is scrolled.
Panes / Align Panes	Causes the current line in both panes to align to each other horizontally.
Panes / Merge Panes	Merges all tabs in both panes into a single pane.
Tabs / Goto Next Tab	Changes the current file to be the file in the next tab in the current pane to the right of the current tab.
Tabs / Goto Previous Tab	Changes the current file to be the file in the next tab in the current pane to the left of the current tab.
Tabs / Goto Last Tab	Changes the current file to be the file in the last viewed tab in the current pane.
Tabs / Goto Other Pane	Changes the current keyboard focus to the current tab in the other pane. This menu item is only available if both panes in viewable.
Tabs / Sort Tabs	Alphabetically sorts the tabs in the current pane.
Tabs / Hide Current Tab	Hides the current tab from view.
Tabs / Hide All Tabs	Hides all of the opened tabs from view.
Tabs / Show All Tabs	Displays all hidden tabs in the tabbar.

Folding / Enable Code Folding	Enables/disables code folding in the current editing buffer. If this option is set to the enable state, the indentation mode dictates what type of code folding will be performed. (OFF = Manual, IND = Indentation-based, IND+ = Syntax-based (if syntax doesn't contain indent/unindent/reindent tokens, indentation-based code folding will be used)).
Folding / Create Fold From Selection	When manual code folding mode is enabled, creates a new fold such that the selected code will be folded.
Folding / Delete Current Fold	When manual code folding mode is enabled, removes the fold indicator at the current cursor's line. If the cursor is not on a fold indicator line, this command will be disabled.
Folding / Delete Selected Folds	When manual code folding mode is enabled, removes all opened/closed folds that are selected in the current editing buffer.
Folding / Delete All Folds	When manual code folding mode is enabled, deletes all of the folds in the current editing buffer.
Folding / Close Current Fold / One Level	Folds the code fold by one level on the cursor's current line.
Folding / Close Current Fold / All Levels	Folds all levels of the code fold located at the cursor's current line.
Folding / Close Selected Folds / One Level	Folds all selected folds by one level.
Folding / Close Selected Folds / All Levels	Folds all selected folds for all levels.
Folding / Close All Folds	Folds all of the code folds in the current editing buffer.

Folding / Open Current Fold / One Level	Unfolds the code fold by one level on the cursor's current line.
Folding / Open Current Fold / All Levels	Unfolds all levels of the code fold located at the cursor's current line.
Folding / Open Selected Folds / One Level	Unfolds all selected folds by one level.
Folding / Open Selected Folds / All Levels	Unfolds all selected folds for all levels.
Folding / Open All Folds	Unfolds all of the code folds in the current editing buffer.
Folding / Show Cursor	If the cursor is hidden inside of folded code, this command will unfold enough code folds to make the cursor visible.
Folding / Jump to Next Fold Mark	Moves the cursor to the next code fold indicator in the current editing buffer.
Folding / Jump to Previous Fold Mark	Moves the cursor to the previous code fold indicator in the current editing buffer.
Set Syntax	Changes the syntax highlighting and language-specific functionality to the specified language. By default, the language is determined by file extension. This menu allows the user to override the default behavior. To permanently add an extension to a language syntax handler, you will need to modify the associated syntax file. See the "Syntax Handling" chapter for more information about the structure of this file. Note: this menu option will not be shown on Mac OS X due to a system crash issue. Please change the menu using either the command launcher or the language selector on the bottom right corner of the editing buffer.
Set Theme	Changes the current syntax coloring scheme to one of the available themes. Setting the theme to this value will only be in effect while the application is running. If the application is quit and

	restarted, the default theme as specified in the preferences will be used.
--	--

Tools Menu

The Tools menu contains various miscellaneous functions that are available within the editor. The following table describes the items found in this menu.

Menu Items	Description
Launcher	Displays the command launcher interface, allowing the user to quickly access commands and other useful functionality as described in the “Command Launcher” chapter.
Theme Editor	Displays the TKE theme editor, which will automatically load the currently used theme for editing. See the Theme Editor chapter for more details about this tool and how to use it.
Start Profiling	Development tool only. Starts the UI profiling facility. This allows procedural performance evaluation.
Stop Profiling	Development tool only. Stops the current profiling run. After profiling information has been gathered, a profile report can be viewed within the editor.
Show Last Profiling Report	Development tool only. Displays the results of the last profiling run within the editor.
Show Diagnostic Logfile	Development tool only. Displays the diagnostic logfile for the current application. This file contains error information from the application and information useful for debugging transient tool issues.
Run BIST	Development tool only. Runs a built-in self test, displaying the resulting output to standard output.
Restart TKE	Development tool only. Allows the editor to be quit, restarted and returned to the current editing state. This is useful when TKE source code is modified and needs to be reloaded.

Sessions Menu

The Sessions menu contains items for saving, deleting, opening, closing and switching named sessions (see the Sessions chapter for more details about sessions and their usage). The following table describes the items in this menu.

Menu Items	Description
Switch To	Quickly closes the current session (named or unnamed) and opens the current window using the named session in the associated submenu.
Close Current	Closes the current named session and reverts the current window to the last state of the unnamed session.
Save Current	Saves the current state of the current named session.
Save As	Saves the current state of the editor as a given named session.
Delete	Permanently deletes a named session.

Plugins Menu

The Plugins menu contains items that allow third-party plugins to be installed, uninstalled and reloaded. Additionally, if TKE is run in developer mode, provides a facility for creating a new plugin quickly. The following table describes the items in this menu.

Menu Items	Description
Install...	Allows new third-party plugins to be installed. See the “Plugins” chapter for more information.
Uninstall...	Allows third-party plugins to be uninstalled. See the “Plugins” chapter for more information.
Show Installed...	Displays a list of the currently installed plugins which includes plugin information. If a plugin is selected and that plugin contains a README.md file associated with the plugin, the information contained in that file will be opened in a read-only editing buffer, allowing you to view usage information about the selected plugin.
Reload	Reloads all installed plugins. This is primarily useful when developing plugins. This menu option allows plugins to be quickly reloaded without requiring the application to be quit and relaunched.
Create...	Development tool only. Creates the template for a new plugin and displays the file in the editor. See the “Plugin Development” chapter for more details about how to create third-party plugins.

Help Menu

The Help menu provides instructional facilities to help you learn how to use this application and to describe the application version information. The following table describes the available items in this menu.

Menu Item	Description
User Guide	Displays this User's Guide in the default PDF or ePub viewer application in your environment. The format type to display can be controlled in your user preferences (see Help/UserGuideFormat).
Tips & Tricks	Displays the TKE Tips & Tricks blog in your browser. Information is added to this website on a weekly basis. Check it out to get the most out of using TKE.
Check for Update	Performs an in-app update. If an update is available, a window detailing the update information will be displayed. If the application is current with the latest available release, a window will be displayed indicating that this is the case. Upon successful completion, the application will be restarted into the new version.
Send Feedback	Creates an e-mail window populated with TKE's developer e-mail address and an appropriate subject line. Fill out the rest of the mail window and send it to provide tool feedback. Feedback is the best way to help us know how we can improve TKE to best meet the needs of its users.
Send Bug Report	Creates an e-mail window populated with TKE's developer e-mail address, an appropriate subject line and information from the application's diagnostic logfile. Add any helpful information about what you were doing at the time the problem occurred to help in understanding the problem. We appreciate good bug reports!
About TKE	Displays a window detailing the current application version and developer information. Clicking the email value will start a mail compose window addressed to TKE development. Clicking on the Twitter name will display the TKE Twitter page in a web browser. Clicking on the website URL will display the TKE website homepage in a web browser. Clicking on the license will display the license in a readonly editing tab. Clicking on the Credits button will change the About window display to show a list of credits to other projects that provided code for TKE (clicking the Back button in this view will display the main About window).

Sidebar

The sidebar is located on the left side of the window. It contains a tree-like view of one or more root directories (any directory in a file system can be a TKE root directory), subdirectories and files. By default, whenever a file is opened within TKE, the file's directory is automatically added to the sidebar. Additionally, the user can open a directory via the "File" menu which is added to the sidebar. This sidebar view allows the user to quickly open other files that are within the same directory without having to navigate through an open dialog box. You may also add a directory to the sidebar by dragging and dropping the directory onto the sidebar. When you have a valid directory dragged into the sidebar, the sidebar border color will turn green to indicate that the item may be dropped.

In addition to being able to quickly open files from the sidebar, several other functions are provided for each type of directory and file. The following subsections identify the different types and their associated functionalities. To access the menu of functionality for a given type, simply right-click on an item in the sidebar. This will display a contextual menu listing the available commands.

Whenever you open a remote file or directory (via the File / Open Remote...) menu option, the associated directory will be displayed in the sidebar alongside local directories. You may perform most of the same operations on remote files that you can with local files with the exception that you may not view difference information.

To hide or show a level of directory hierarchy, left-click on the disclosure triangle next to the directory to show/hide. Alternatively, you may use the space bar or return key to toggle the disclosure state. You may also use the left/right keys to specifically close or disclose directory information.

To operate on more than one file, you can select multiple files or directories by holding Control or Command while left-clicking and then right-click to display the contextual menu. Note that TKE will keep you from selecting both files and directories (whichever type the first selection is determines what will be allowed to be selected).

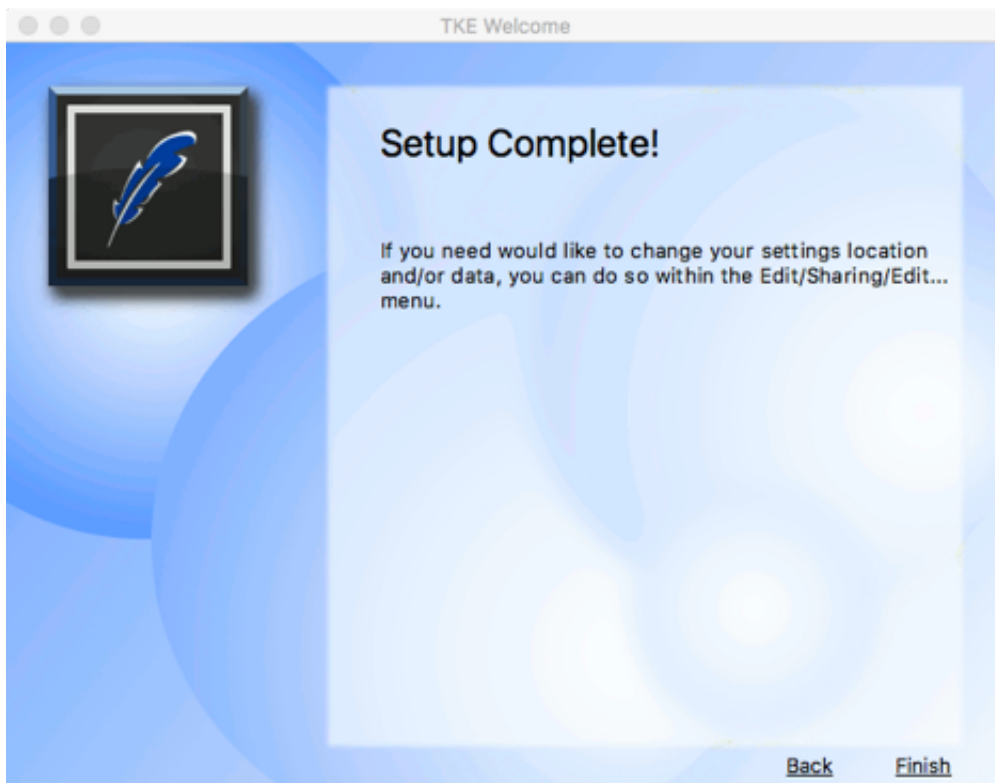
Files within the sidebar can be automatically filtered out of the sidebar via the “Sidebar/Ignore*” preference items. Any files that match any of these patterns will not be displayed in the sidebar. This is useful for de-cluttering the sidebar with files that cannot be edited within TKE (i.e. object files, image files, etc.)

Files and directories are added in alphabetical order. If you would prefer to have all folders be listed first, followed by all of the files within the same directory, you can specify that in the View preferences pane.

Root Directory

A root directory in the sidebar is any directory that doesn’t have a parent directory immediately shown in the sidebar. You may have more than root directory listed in the sidebar. To view the full pathname of a root directory, hover the cursor over the directory name until the tooltip appears.

The following image is a depiction of the sidebar with the root directory highlighted.



The following table lists the available contextual menu functions available for root directories.

Menu Item	Description
New File	Adds a new file to the root directory. If this menu item is selected, an entry field at the bottom of the window displayed, allowing the user to specify a filename for the new file. Entering a name and hitting the RETURN key will create the new file in the directory and open the file in the editor.
New File From Template	Opens a new file to the selected directory in an editing buffer. A prompt for a filename will be displayed at the bottom of the main window. After a name is entered and the RETURN key pressed, a list of available templates will be displayed. Selecting a template will create the new tab, insert the text, and perform any snippet substitutions.
New Directory	Adds a new directory to the root directory. If this menu item is selected, an entry field at the bottom of the window is displayed, allowing the user to specify a name for the directory. Entering a name and hitting the RETURN key will create the new directory.
Open Directory Files	Opens all shown files that are within the directory.
Close Directory Files	All open files in the editor that exist within the root directory and below it will be closed. Any files which require a save will prompt the user to save or discard the file modifications.
Disconnect From Server	This option will only be displayed for root directories that exist on a remote server. Selecting this item will automatically remove the associated directory hierarchy from the sidebar, close all files and disconnect the associated FTP/SFTP/WebDAV connection.
Hide Directory Files	Hides the tabs of any opened files in this directory in the tabbar. Useful for focused workflows.
Show Directory Files	Shows any hidden tabs of opened files within the selected directory.
Copy Pathname	Copies the pathname of the selected root directory to the clipboard.
Rename	Renames the root directory in the file system. The current full pathname will be specified in an entry field at the bottom of the application window. Once filename editing is complete, hit the RETURN key to cause the rename to occur. Hit the ESCAPE key to cancel the renaming operation.
Delete	Deletes the root directory from the filesystem and removes the directory from the sidebar. If this item is selected, an affirmation prompt will be displayed to confirm or cancel the deletion. This option will be displayed if the “Use Move to Trash” general preference option is unset.
Move to trash	Moves the selected directories to the file system’s trash. Directories will be moved without a user prompt. This option will be displayed if the “Use Move to Trash” general preference

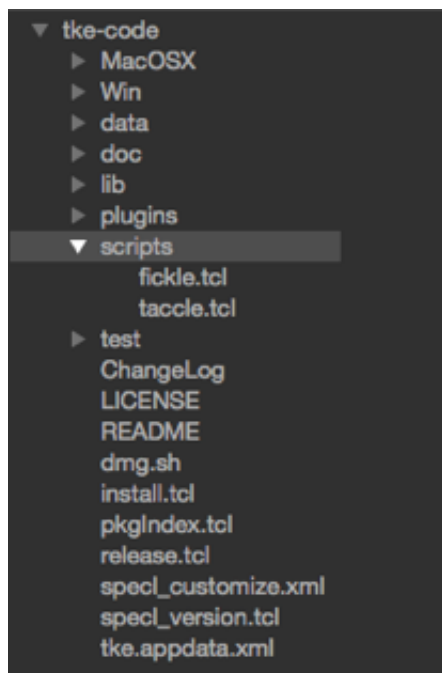
	option is set.
Favorite/Unfavorite	Marks the selected directory to be a favorite (if the Favorite command is selected) or removes it from the favorites list (if the Unfavorite command is selected). Favorited directories can be quickly added to the sidebar via the File / Open Favorite menu or the command launcher.
Remove from Sidebar	Removes the root directory from the sidebar (no modification to the file system will take place). If this item is selected, the entire root directory is removed from the sidebar.
Add Parent Directory	Adds the parent directory in the filesystem of the root directory. The current root directory will no longer be a root directory (replaced by the parent directory) but will become a standard directory underneath the parent.
Make Current Working Directory	Changes the current working directory to the root directory. Selecting this item will make all file operations within the editor relative to the selected directory. Additionally, the working directory information in the title bar will be updated to match this directory.
Refresh Directory Files	Updates the sidebar contents for the root directory.

In addition to these functions, plugins can also add functionality beneath these items in the menu. See the [Plugins](#) and [Plugin Development](#) chapters for more information.

Directory

A non-root directory is any directory in the sidebar which has a parent directory associated with it (i.e., any directory in the sidebar that is not a root directory).

The following image depicts the sidebar with a non-root directory highlighted.



The following table lists the available contextual menu functions available for non-root directories.

Menu Item	Description
New File	Adds a new file to the directory in both the sidebar and the file system. If this menu item is selected, an entry field at the bottom of the window displayed, allowing the user to specify a filename for the new file. Entering a name and hitting the RETURN key will create the new file in the directory and open the file in the editor.
New File From Template	Opens a new file to the selected directory in an editing buffer. A prompt for a filename will be displayed at the bottom of the main window. After a name is entered and the RETURN key pressed, a list of available templates will be displayed. Selecting a template will create the new tab, insert the text, and perform any snippet substitutions.
New Directory	Adds a new directory under the selected directory in both the sidebar and the file system. If this menu item is selected, an entry field at the bottom of the window is displayed, allowing the user to specify a name for the directory. Entering a name and hitting the RETURN key will create the new directory.
Open Directory Files	Opens all shown files that are within the directory.
Close Directory Files	All open files in the editor that exist within the directory and below it will be closed. Any files which require a save will prompt the user to save or discard the file modifications.
Hide Directory Files	Hides all opened file tabs within the selected directory in the tabbar. Useful for focused workflows.
Show Directory Files	Shows all hidden tabs in the tabbar whose associated files exist within the selected directory.
Copy Pathname	Copies the selected directory pathname to the clipboard.
Rename	Renames the directory in the file system. The current full pathname will be specified in an entry field at the bottom of the application window. Once filename editing is complete, hit the RETURN key to cause the rename to occur. Hit the ESCAPE key to cancel the renaming operation.
Delete	Deletes the directory from the filesystem and removes the directory from the sidebar. If this item is selected, an affirmation prompt will be displayed to confirm or cancel the deletion. This option will be displayed if the “Use Move to Trash” general preference option is unset.
Move to trash	Moves the selected directories to the trash. Directories will be moved without a user prompt. This option will be displayed if the “Use Move to Trash” general preference option is set.
Favorite/Unfavorite	Marks the selected directory to be a favorite (if the Favorite command is selected) or removes it from the favorites list (if the Unfavorite command is selected). Favorited directories can be quickly added to the sidebar via the File / Open Favorite menu or the

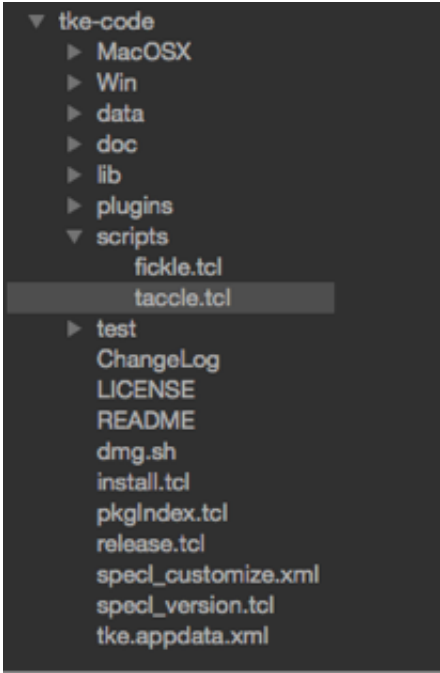
	command launcher.
Remove from Sidebar	Removes the directory from the sidebar (no modification to the file system will take place). If this item is selected, the entire directory is removed from the sidebar.
Remove Parent from Sidebar	Removes all parent directories of the selected directory from the the sidebar and makes the selected directory a root directory in the sidebar.
Make Current Working Directory	Changes the current working directory to the selected directory. Selecting this item will make all file operations within the editor relative to the selected directory. Additionally, the working directory information in the title bar will be updated to match this directory.
Refresh Directory Files	Updates the sidebar contents for the selected directory.

After these functions will be listed any directory popup menu items that are added via plugins. See the Plugins and Plugin Development chapters for how to create these plugin types.

Files

Files have different functions available to them than directories. Double-clicking any file will open the file in the editor.

The following image depicts the sidebar with a file highlighted.



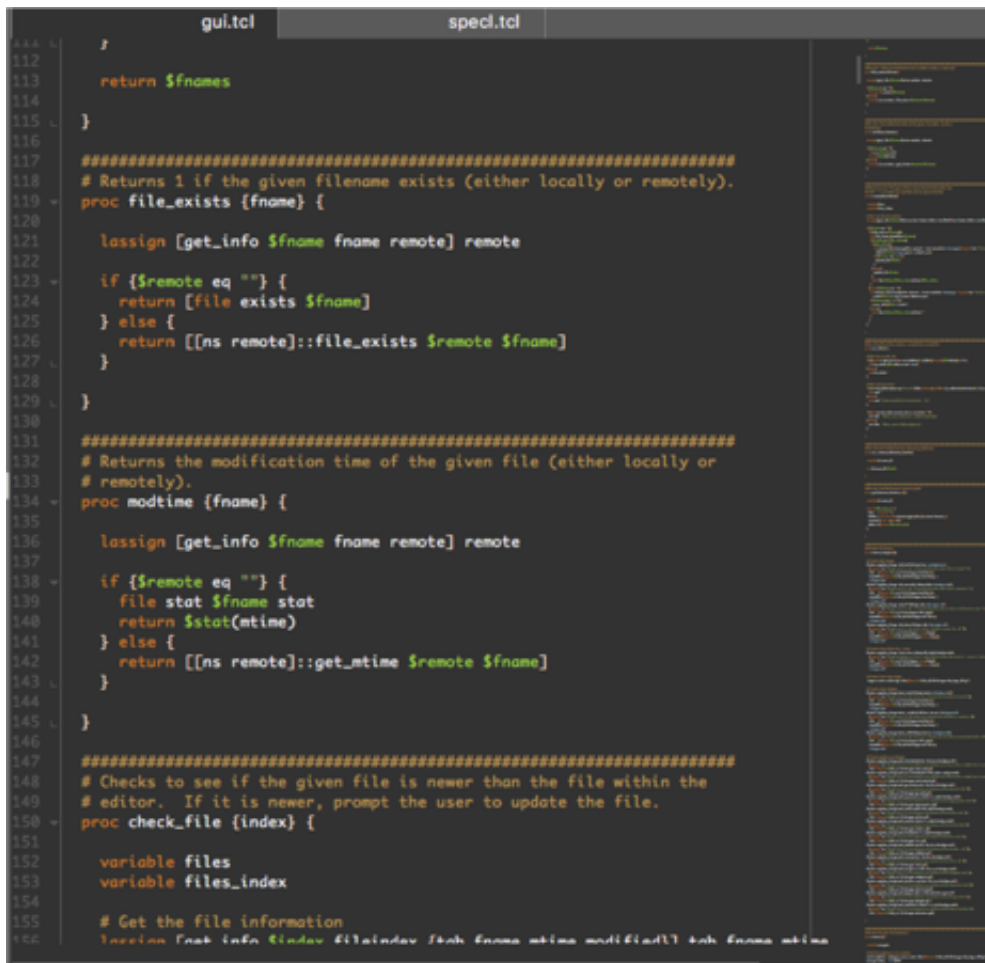
The following table lists the available functions for files.

Menu Item	Description
Open	Opens the selected file in the editor. An opened file will have a color applied to its background to make it easy to identify opened files in the sidebar.
Close	Closes the selected file in the editor. If the file has been modified, a prompt will be displayed asking if the changes should be saved or not.
Hide	Hides the tab of the selected file in the tabbar.
Show	Unhides the tab of the selected file in the tabbar.
Show Difference	Displays a difference view of the currently selected file.
Copy Pathname	Copies the selected file pathname to the clipboard.
Rename	Renames the selected file in the file system. The current full pathname will be specified in an entry field at the bottom of the application window. Once filename editing is complete, hit the RETURN key to cause the rename to occur. Hit the ESCAPE key to cancel the renaming operation.
Duplicate	Will create a duplicate file of the selected file in the same directory. The file will be named with a unique name and will be editable. Use the ‘Rename’ command to change the name, if necessary.
Delete	Deletes the file from the filesystem and removes the file from the sidebar. If this item is selected, an affirmation prompt will be displayed to confirm or cancel the deletion. This option will be displayed if the “Use Move to Trash” general preference option is unset.
Move to trash	Moves the selected files to the trash. No user prompt will be displayed to confirm the request. This option will only be displayed if the “Use Move to Trash” general preference option is set.
Favorite/Unfavorite	Marks the selected file to be a favorite (if the Favorite command is selected) or removes it from the favorites list (if the Unfavorite command is selected). Favorited files can be quickly opened in the editor via the File / Open Favorite menu or the command launcher.

After these functions will be listed any file popup menu items that are added via plugins. See the Plugins and Plugin Development chapters for how to create these plugin types.

Editing Pane

The editing pane comprises the majority of the application window, displaying all files that are open for reading/writing. The following image shows a representation of this pane.



The pane is made up of four (or optionally five) main areas:

1. Tab bar
2. Line number bar
3. Editing Buffer
4. Scrollbars
5. Bird's Eye view

Tab Bar

The Tab Bar sits at the top of the editing pane, allowing for more than one file to be edited at a time and quick switching between text windows. To switch to a different tab, simply left-click on any tab in the tab bar.

By default, tabs are added to the tab bar in the order they were opened; however, the user can change this order by clicking on a given tab and dragging the tab to a new position in the tab bar. Additionally, you can sort the tabs in alphabetical order by selecting the “View / Tabs / Sort Tabs” menu item. If you would always like tabs to be sorted in alphabetical order, you can change this behavior in the user preference file (see the Preferences chapter for information on how to edit this file).

When more files are opened than the tab bar has space to hold, the tab bar will adjust itself to show as many tabs as is comfortable to display and then display some tab shift buttons on each side of the tab bar (indicated with left and right arrows). To view tabs off-screen to the right, click on the right button. To view tabs off-screen to the left, click on the left

button. Alternatively, you can use your mouse’s scrolling capability with the mouse pointer within the tabbar to quickly scroll the list of tabs to the right or left. To see a list of all opened tabs, right-click on either the left or right arrow buttons and select a file from the displayed dropdown list. The resulting list will show an in-order list of tabs in the resulting menu with separators strategically placed, to quickly find which tabs are off-screen on each side of the currently displayed tab bar.

In addition to switching between different opened text windows, the tab bar also provides a number of other functions and visual indicators. When a file is locked, a locked icon will be displayed on the left side of the associated tab. If the tab contains a file in difference mode/view, a -/+ icon will be displayed on the left side of the associated tab. When a file has been modified, an asterisk will be displayed to the left of the file name, indicating that a save is needed to commit the changes to the file. The base name of the file is displayed in the tab along with file extension; however, if you would like to see the full pathname of the file, simply hover the mouse over any portion of the tab and a tooltip is displayed with this information. To close the tab, move the mouse cursor over the tab and a close icon is displayed on the right side of the tab, clicking on this button will close the associated tab (if the file is closed and the text has been modified, a prompt will be displayed indicating that the file has not been saved, allowing the user to save the file).

To help eliminate tab clutter, TKE now has a tab hiding feature built into the menu, the sidebar and the tab contextual menu that allows you to hide/show tabs within the tabbar. If a tab is hidden, its contents remain in tact (i.e., unsaved changes will exist) unlike if the tab is closed. This tab hiding feature can help you to create focused workflows while editing without losing the file state of the hidden tabs (ex., you can still see which files are opened but hidden in the sidebar by viewing the file’s icon in the sidebar).

In addition to these functions, you can also access a drop down menu of functionality by right-clicking on a tab. The following table summarizes this functionality.

Menu Item	Description
Close Tab	Closes the current tab. This is the exact same behavior as clicking on the close button within the tab.
Close Other Tabs	Closes all of the other tabs in the tab bar, leaving this tab as the only opened tab in the editing pane.
Close All Tabs	Closes all tabs in the tab bar.
Hide Tab	Hides the current tab from view. You can show the tab by either using the sidebar or by using the View / Tabs / Show All Tabs option.
Split View	Creates another editing buffer in the same tab (it is placed above the current editing buffer). This buffer is a second view into the same file. This allows you to view and edit two different views of the same file. Selecting this command again will remove the second editing buffer view.
Bird's Eye View	If checked, displays the bird's eye view of the current file within a narrow ribbon on the left side of the editing buffer. This option allows the user to override the view menu option and/or preference option that controls the display of the birds' eye view.
Locked	Indicator of the locked status of the file and allows the user to toggle the locked status of the file. If a checkmark is displayed next to this item, the file is locked and cannot be modified within TKE. If no checkmark exists, the file is modifiable.
Favorited	Indicator of the favorite status of the file and allows the user to toggle the favorite status of the file. Favorited files can be quickly opened via the File / Open Favorite menu or the command launcher.
Show in Sidebar	Causes the current file to be displayed in the sidebar (even if the directory is not disclosed).
Move to Other Pane	Moves the tab and associated text window to the other editing pane, allowing for side-by-side text editing. If the other editing pane does not exist, it will be created. If the current tab is the only tab in the current pane and it is moved, the current pane will disappear, leaving only a single pane displayed in the editing panel. This option will not be available if there is only one tab opened in the editing panel.

Line Number Bar

The line number bar is displayed on the left side of the editing panel. Two line numbering schemes are available: absolute and relative. In absolute line numbering, each number is associated with a corresponding line in the text window starting with line 1. In relative line numbering, the line containing the insertion cursor is numbered 0. Line numbers increase by one above the current line and below the current line. You can select between the line numbering schemes in the View / Line Numbering menu option.

In addition to showing line numbers, the line number bar also provides some useful selection functionality. If the left-button is clicked on a line number, the entire corresponding line is selected in the text window. If the left-button is held down while the mouse cursor is moved, the clicked line and all lines between that line and the current mouse cursor will be selected. If the SHIFT button is held when the left mouse button is clicked, all lines between the last left-clicked line and the current line are selected.

When code folding is enabled for the current editing buffer (via the View / Folding / Enable Code Folding menu option), the line number bar will contain graphical disclosure triangles for each foldable line in the editing buffer. Left-clicking on a disclosure triangle will toggle the code fold at that location, hiding one level of indentation. If the Shift key is held while left-clicking on a disclosure triangle, code folding will be toggled in a deep manner, affecting all levels of indentation at or below the selected code block. You can unfold all folded code or fold all foldable code using the View / Folding menu options.

Finally, the line number bar can be used to create markers. Markers are basically jump points in the text window. Any line can be marked by right-clicking on a line number. When this occurs, an entry field at the bottom of the window is displayed, allowing the user to provide a name for the marker. Giving a name to a marker is optional, but useful. To give the marker a name, enter a string and hit the RETURN key. To create an unnamed marker, simply hit the RETURN key in the name entry field. To cancel the creation of the marker, hit the ESCAPE key in the name entry field. After a marker has been created, the corresponding line number will be given an orange highlight in the line number bar.

To clear an existing marker, simply right-click on a line that has already been given a marker. The line highlight will be cleared to indicate that the marker no longer exists.

Editing Buffer

The editing buffer provides the main source of editing functionality. TKE supports two modes of editing functionality: Vim mode and standard mode. See the Vim chapter to see what functions are available when editing in this mode, the rest of this section will only mention functions available when the editor is not in Vim command mode (i.e., either Vim insert mode or standard mode).

Visually the editing buffer contains two basic UI elements: the text editor pane and the scrollbars.

The text editor pane allows text to be read and modified. The following table specifies the different key and mouse bindings on the editor that the user can take advantage of.

Key/Mouse Binding	Description
Left-mouse click	Sets insertion cursor just before the character underneath the mouse cursor. Clears any selections. If left-button is held while mouse is moved, selection is created between insertion cursor and under mouse cursor.
Left-mouse double click	Selects the word under the mouse and positions insertion cursor at the start of the word. Holding mouse button while dragging will select all words between insertion cursor and mouse cursor.
Left-mouse	Selects the entire line under the mouse. Holding mouse button while dragging will select all

Left-mouse triple click	Selects the entire line under the mouse. Holding mouse button while dragging will select all lines between insertion line and mouse cursor line.
Shift + Left-mouse click + drag	Adjusts the end of the selection nearest the mouse cursor when the left button is pressed.
Shift + Left-mouse double click + drag	Adjusts the end of the selection nearest the mouse cursor in whole word units.
Shift + Left-mouse triple click + drag	Adjusts the end of the selection nearest the mouse cursor in line units.
Control + left-mouse click	Repositions the cursor without affecting the selection.
Middle-mouse click	Selection is copied into the text at the position of the mouse cursor.
Middle-mouse click + drag	Moves the current view of the text window.
Insert key	Inserts the current selection at the position of the insertion cursor.
Left/Right key	Moves the insertion cursor one position to the left/right and clears the selection.
Shift + left/right key	Moves the insertion cursor one position to the left/right and adds the character to the selection.
Control + left/right key	Moves the insertion cursor to the left/right by one word.
Shift + Control + left/right key	Moves the insertion cursor to the left/right by one word and adds the word to the selection.
Up/Down key	Moves the insertion cursor one line up/down and clears the selection.
Shift + up/down key	Moves the insertion cursor one line up/down, extending the selection.
Control + up/down key	Moves the insertion cursor by paragraphs (groups of lines separated by blank lines).
Shift + Control + up/down key	Moves the insertion cursor by paragraphs, extending the selection.
Next/Prior key	Moves insertion cursor forward/backward by one screenful of text and clears the selection.
Shift + next/prior key	Moves insertion cursor forward/backward by one screenful of text, extending the selection.

Next/prior key	
Control + next/prior key	Moves screen forward/backward by one screenful of text without affecting insertion cursor or selection.
Home key OR Control + 'a' key	Moves the insertion cursor to the beginning of its current line and clears any selection.
Shift + Home key	Moves the insertion cursor to the beginning of the line, extending the selection to that point.
Control + Home key	Moves the insertion cursor to the beginning of the text and clears the selection.
Shift + Control + Home key	Moves the insertion cursor to the beginning of the text, extending the selection.
End key OR Control + 'e' key	Moves the insertion cursor to the end of its current line and clears the selection.
Shift + End key	Moves the insertion cursor to the end of its current line, extending the selection to that point.
Control + End key	Moves the insertion cursor to the end of the text and clears the selection.
Shift + Control + End key	Moves the insertion cursor to the end of the text, extending the selection.
Control + '/' key	Selects all of the text.
Control + '\' key	Clears the selection.
Delete key	Deletes the selection (if one exists) or deletes the character to the right of the insertion cursor.
Backspace key	Deletes the selection (if one exists) or deletes the character to the left of the insertion cursor.
Control + 'k' key	Deletes from the insertion cursor to the end of the line. If the insertion cursor is already at the end of the line, the newline character is deleted.
Control + 'o' key	Opens a new line by inserting a newline character in front of the insertion cursor without moving the insertion cursor.
Control + '+' key	Increases the font size of the editor text by a size of one.
Control + '-' key	Decreases the font size of the editor text by a size of one.

Control + - key	Decreases the font size of the editor text by a size of one.
File drag-drop	Inserts the contents of the file at the location of the cursor (which will follow the mouse cursor).
Text drag-drop	Inserts the selected text associated with the drag-drop operation at the location of the cursor (which will follow the mouse cursor).
Alt + Left mouse click	Adds a cursor to the multicursor list at the character under the mouse cursor. Also makes the current cursor the anchor cursor.
Alt + Right mouse click	Adds one or more cursors between the anchor cursor and the current cursor such that one cursor will be placed on each line at the same column location as the anchor cursor.
Shift + Alt + Left mouse click + drag	Selects a column of text with the upper left corner of the selection starting at the button press position and the lower right corner ending at the button release position.

On MacOS and Windows, you can drag a file into the text editing area to insert the files contents into the editing buffer starting at the insertion cursor. You can also drag and drop text into the editing buffer to just insert that text into the editing buffer.

Scrollbars

The scrollbars allow you to change the text view displayed in the editing area. By default, both scrollbars display a minimalist, thin scrollbar. Using a mouse wheel gesture will also move the viewing area up and down (the slider in the vertical and horizontal scrollbars will give an indication as to what portion of the file is currently in view). If the text area is not scrollable, the associated scrollbars will be hidden from view. Moving the mouse cursor into the scrollbar area will make the slider larger so that it is easier to grab, if necessary.

Besides using mouse scrolling to change the view, you can grab the scroll slider and move it to any spot in the file. Left clicking in the scrollbar area will cause the view to jump to the selected point of the file. Right clicking in the scrollbar area will adjust the view by a single page (if the area above the slider is right-clicked, the view will move up by a page; if the area below the slider is right-clicked, the view will move down by a page). Right-clicking on the slider will not change the view.

If TKE is operating in multi-pane mode (i.e., two editing buffers are displayed side-by-side) and the View / Panes / Enable Synchronized Scrolling option is set, scrolling either editing buffer will automatically scroll the other pane's editing buffer in lock step. This is useful if you are manually viewing differences between two different files. When sychronized scrolling mode is enabled, the scrollbar sliders will change to a different color (red in most cases) to indicate the current mode. To assist in aligning the panes, you can set the current cursor in both panes to the lines that you would like to see aligned and choose the View / Panes / Align Panes menu option which will cause both panes to align their views such that the current lines in each pane will be vertically aligned. Once that has been set, enabling synchronized scrolling will keep those lines in vertical alignment.

In addition to changing the view, the vertical scrollbar on the left side of the editing buffer can also display a simple file map. If markers are set in the file, they will be displayed in the scrollbar area with a horizontal line using the same color of the marker in the line number area. For difference views, the difference lines will also be drawn in the scrollbar area.

The top of the file scrollbar area will display map information for the first line in the file while the bottom of the file scrollbar area will display map information for the last line in the file.

Bird's Eye View

The bird's eye view displays the current file in a read-only fashion using a small font, allowing a large portion of the file to be viewed to help provide context and for visually searching text. When it is displayed, placing the cursor into any portion of it will highlight the currently displayed text within the bird's eye view. Clicking on the highlighted text in the bird's eye viewer and dragging up/down will adjust the view in the associated editing buffer. Clicking on any non-selected portion of the bird's eye view will jump the editing buffer view to center on the clicked line of text. You can adjust the bird's eye view by using the scroll wheel function of your mouse while the cursor is within the bird's eye view.

Displaying the view can be handled at three different points:

1. View preference panel
 - Controls default bird's eye view when the application is started or when new files are opened.
2. View menu
 - Changes the show/hide status of the bird's eye view for all opened files and will control the status of the bird's eye view on any newly opened file.
 - Overrides the preference value as long as the application remains open.
3. Tab contextual menu
 - Allows the show/hide status for the current tab.
 - Overrides the previous controls.

In terms of preference control over the bird's eye view, you can set the character width of the view and the size of the font.

Find Highlighting

TKE supports searching within a text window via the "Find" menu. When a string is searched within the text window, all matching text will be highlighted and the insertion cursor will be placed at the beginning of the first matched text. This allows you to quickly see all matches within the text window.

When either the "Find" or "Find and Replace" functions are invoked, if text is currently selected in the text window, that text will be automatically placed in the search field.

Find in Files

When the user performs a "Find in Files", a special tab will be added to the editing pane. This file will contain snippets of lines of text from all files that have matches to the search text. Within the window, all matching text will be highlighted the same yellow that is used for normal searching. If the user left clicks or hits the space bar when the insertion cursor is within on any matching text within this tab, the corresponding file will automatically be added to the editing pane as a new tab and the insertion cursor will be placed at the beginning of the matching text in the file. This allows you to quickly find and get to the matches within the editor.

Status Bar

The status bar is the area located at the bottom of the application window. It's function is to display the following information to the user.

- Vim mode (if the editor is currently in Vim mode).
- Vim macro recording mode (only displayed when recording and includes the name of the buffer being recorded to).
- Current row and column position of cursor within the current editor.
- Informational, temporal messages provided by the application.
- Current mode of auto-insert for the current editor (includes ability to change the indentation mode for the current editor).
- Display current syntax applied to current editor (and ability to change that language).

The following image is a representation of the status bar.



Vim Mode

If the current editor is in Vim mode, the right-most indicator will specify the current mode that Vim is in.

Vim Macro Recording Status

If the current editor is in Vim mode, the next indicator on the left of the status bar will display this state along with the name of the buffer being recorded into. When recording mode is completed within the current editor, this status indicator will clear from the information bar.

Cursor Position Status

The cursor position status information is located on the left side of the status bar. If a file is currently being edited, the current row and column position of the cursor is displayed. Additionally, if the editor is running in Vim mode, the current Vim mode is displayed just to the right of the position information.

Message Display

To the right of the position status information is a typically blank area which can be used to display temporary informational messages to the user from the application. If a message is displayed in this area, it will appear and then disappear after several seconds (keeping the message area blank again).

Auto-Indent Indicator

Near the right side of the status bar is an indicator of the auto-indent mode. A value of "OFF" indicates that indentation mode is off. A value of "IND" indicates that auto-insertion mode is enabled. A value of "IND+" indicates that smart indentation mode is enabled. To change the indentation mode, you may either change it via the "Edit/Insert Mode" menu

or simply click on the indentation indicator to display a popup menu where you can change the mode.

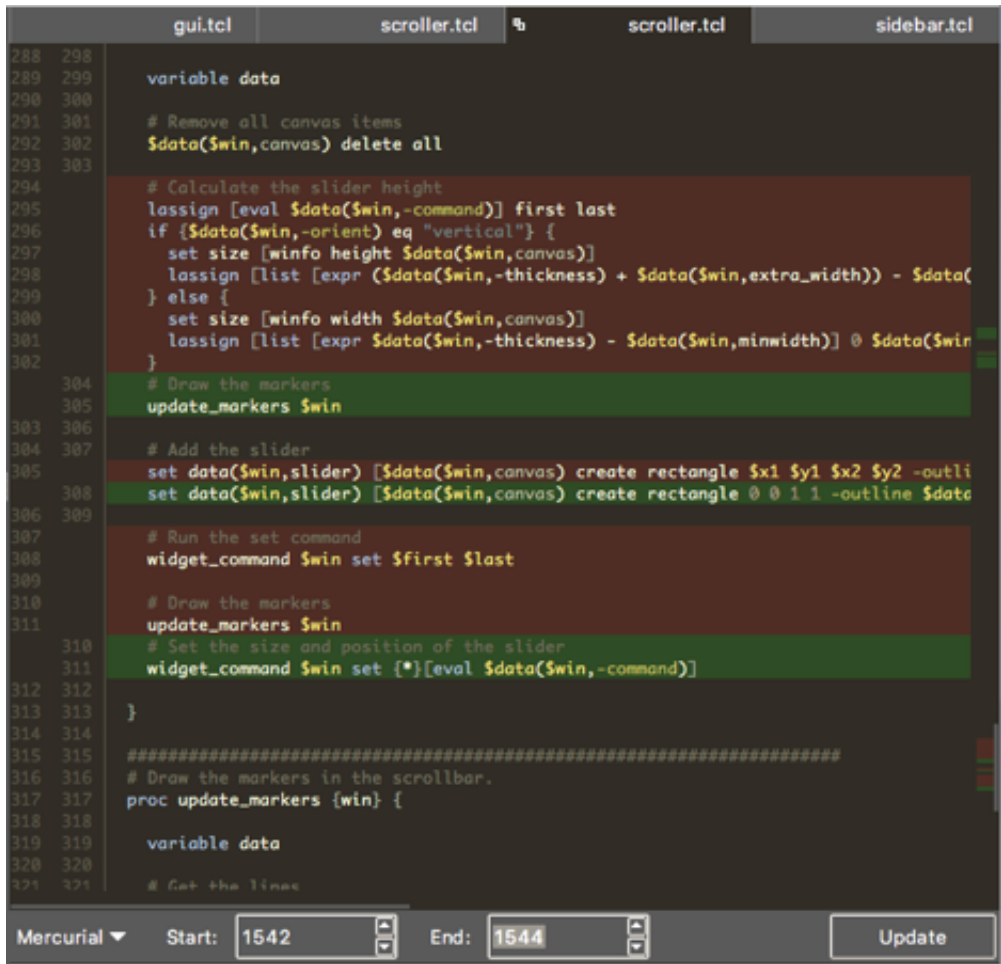
Syntax Display

On the right side of the status bar is the syntax display area. If a file is currently being edited, the current syntax highlighting language is displayed. To change the current language, simply left-click on the language name and select a different language from the list. If the current file cannot be automatically discerned by TKE, a value of “<None>” will be displayed in the status bar.

Difference Viewer

When a file is displayed in the editor, the “Show File Difference” option in the File menu will create a new tab in the editor with the file shown in “Difference View”. This view is somewhat different than when a file is in edit mode. The purpose of this view is to allow the user to visually see file differences between two versions of the same file (available from a version control system like Mercurial, Git, Bazaar, Subversion or Perforce), between two different files (using the Posix standard diff utility), or using output from a customized command that produces unified difference output.

The following image shows what a difference view looks like.



The window is comprised of 4 main parts:

- Line number bar (left)
- Main file viewing area (middle)
- Difference map (right)
- Control panel (bottom)

Line Number Bar

The line number displays two sets of line numbers, corresponding to the associated line numbers of each file that is part of the difference. The numbers on the left side correspond to the first file while the numbers on the right side correspond to the second file. Gaps in line numbers represent differences that the associated file has with the other file.

Like the line number bar in normal editing view, you can click on a set of line numbers to create a marker and you can click and drag in this bar to select all associated lines of text for the purposes of copying.

Main File Viewing Area

The main viewing area displays both of the files that are a part of the difference. Lines that are in common are displayed as a file normally when being edited. Lines that are a part of the first file, but not the second are displayed in a reddish color. Lines that are a part of the second file, but not the first are displayed in a green color. Text that is displayed in the main file viewing area are not editable; however, text may be selected and copied to the clipboard. You can also save the contents of the viewer under a supplied name (i.e., saving the buffer contents will not automatically overwrite the original file).

Difference Map

The difference map on the right-hand side of the window is displayed in lieu of the standard scrollbar, functioning in the same way but providing a quick view of the difference information in the entire file. Click on an area in the scrollbar to display that portion of the file in the file viewing area. Drag the scrollbar to any position in the difference map to also view that portion of the file. The inside of the slider is empty, allowing the user to easily see the difference information in the area that the slider resides.

Control Panel

The control panel is displayed just below the main file viewing area. It provides the user a simple method of changing which versions of the given file are differenced in the main file viewing area.

On the left side of the control panel is the version system selection menu. By default, TKE will attempt to automatically determine which version system is managing the file. The following values are currently supported:

System	Description
Mercurial	Uses the Mercurial version control system. Select the first and second versions using the selectors in the control panel to change which versions are differenced.
Bazaar	Uses the Bazaar version control system. Select the first and second versions using the selectors in the control panel to change which versions are differenced.
Git	Uses the Git version control system. Select the first and second version using the selectors in the control panel to change which versions are differenced. Versions are represented by their shortened SHA-1 values.
Perforce	Uses the Perforce version control system. Select the first and second versions using the selectors in the control panel to change which versions are differenced.
Subversion	Uses the Subversion version control system. Select the first and second versions using the selectors in the control panel to change which versions are differenced.
diff	Allows the user to perform a Unix diff of the current file and another file in the file system. Simply enter the pathname of the file to compare to the file loaded in the main file viewing area to perform the difference.
custom	Allows the user to enter in a specific difference command to execute in the shell. The output of the difference command must be in unified difference output format.

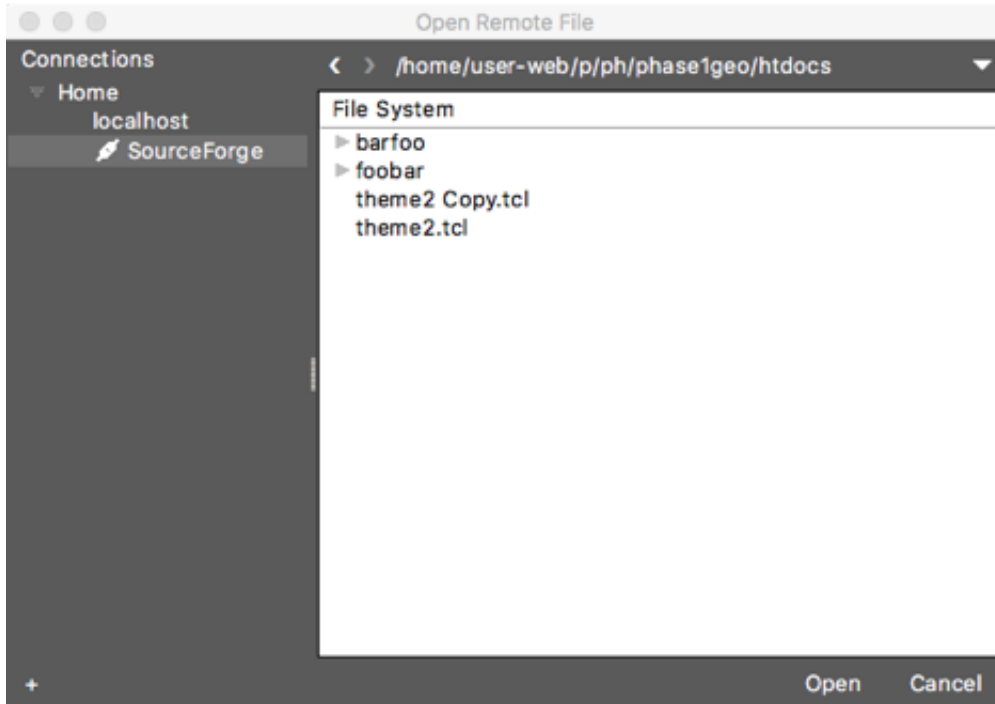
Note: If you have one of the above tools installed on your system, but TKE fails to identify which tool is to be used for the file, most likely the problem is that the tool is not in your environment's binary path. To fix this you can either add the directory to your PATH environment variable path (if you are starting TKE from the command-line) or use the preference General panel, select the Variables tab and set the PATH environment variable within the Variables table. Once you have properly setup the PATH variable within the preferences panel, TKE should be able to (without a restart) find the proper tool that is managing the file.

To the right of the version system menu is either a group of version selectors (if the menu displays a file version system), a file entry box (if the menu displays the “diff” option), or a command entry box (if the menu displays the “custom” option). Use these widgets to quickly display the desired difference. By default, if TKE was able to automatically determine the version system being used, TKE will setup the selector widgets such that the first version is the last committed version and the second version is the current working copy of the file. The main file viewing area will automatically display the difference information. Whenever the user clicks on one of these widgets or changes the displayed version, an information window will be displayed just above the control panel displaying the logfile information of the current version in the widget. A preference value exists that can allow the user to show/hide this information when versions are changed. Changing the mouse/keyboard focus to another widget will hide this informational display from view.

To change the file versions, simply adjust the selector widgets to match the desired versions and click on the “Update” button that will be displayed on the right side of the control panel. The file viewing area will only be modified after the “Update” button is clicked. If the “Update” button is not available, it indicates that the output data in the main file viewer matches the current selections in the control panel.

Remote File Editing

In addition to the ability to view/edit files on the local file system and across an NFS mounted drive, TKE also has the ability to view/edit files natively on remote systems using FTP/SFTP/WebDAV. To edit remote files, use the File / Open Remote... menu option to display the remote file dialog window. A depiction of this window is displayed below.



The window is comprised of three areas:

1. Connection bookmark sidebar (left)
2. Remote file viewer (right)
3. Button bar (bottom)

Connection Bookmark Sidebar

On the left side of the window is the connection bookmark sidebar. This is the area which shows all available connections that have been previously setup and saved. The sidebar is organized by groups. By default, a single group called "Group" will occupy the sidebar. This group may be renamed and/or deleted (if another group exists in the sidebar — at least one group must exist in the sidebar). To create a new group, click the "+" button in the button bar and select the "New Group" option. This will display a small popup window allowing you to specify the new group name. Entering a unique name and clicking the "Create" button will create the new group in the sidebar. Groups may be moved around in the sidebar by selecting and dragging them to a new position in the sidebar.

To rename a group name, right click the group name and select the "Rename Group" menu option. This will display a renaming popup. Entering a name and selecting the "Rename" button will change the group name in the sidebar. Note that groups may only be renamed if they do not contain any active connections (i.e., all connections within the group are in the disconnected state — noted by the lack of a connection icon next to the connection name).

To delete a group name, right click the group name and select the “Delete Group” menu option. A confirmation popup will be displayed to confirm the group deletion (which will delete all connections within the group as well). Like the “Rename Group” option, this option will only be available if all of the group connections are in the disconnected state.

To create a new remote connection, either select a group name, right-click and select the “New Connection” menu option or select a group and click on the “+” button in the lower left corner of the window and select the “New Connection” option. This will make display the connection editor pane (described below). Clicking on the “Create” button after filling in all of the non-optional fields will add the connection to the sidebar under the group specified with the “Group” menu in the connection editor. You may move a disconnected connection to a different group or change its position within the group by dragging the connection to a new position within the sidebar.

To edit an existing connection, right click the connection and select the “Edit Connection” option. To delete the connection, right click the connection and select the “Delete Connection” option. A confirmation popup will be displayed prior to deleting the connection. To test a connection for connectivity, right click the connection and select the “Test Connection”, after a moment a popup will be displayed indicating the availability of the connection. To start a connection, either double-click the connection or right-click the connection and select the “Open Connection” option. After the remote connection has been made, the connection’s directory contents will be displayed in the remote file viewer area on the right of the window. To disconnect an open connection, right-click the connection and select the “Close Connection” option. Note that when a connection is currently in the opened state (as indicated in the sidebar with the connected icon next to the connection name), only the “Close Connection” option will be enabled.

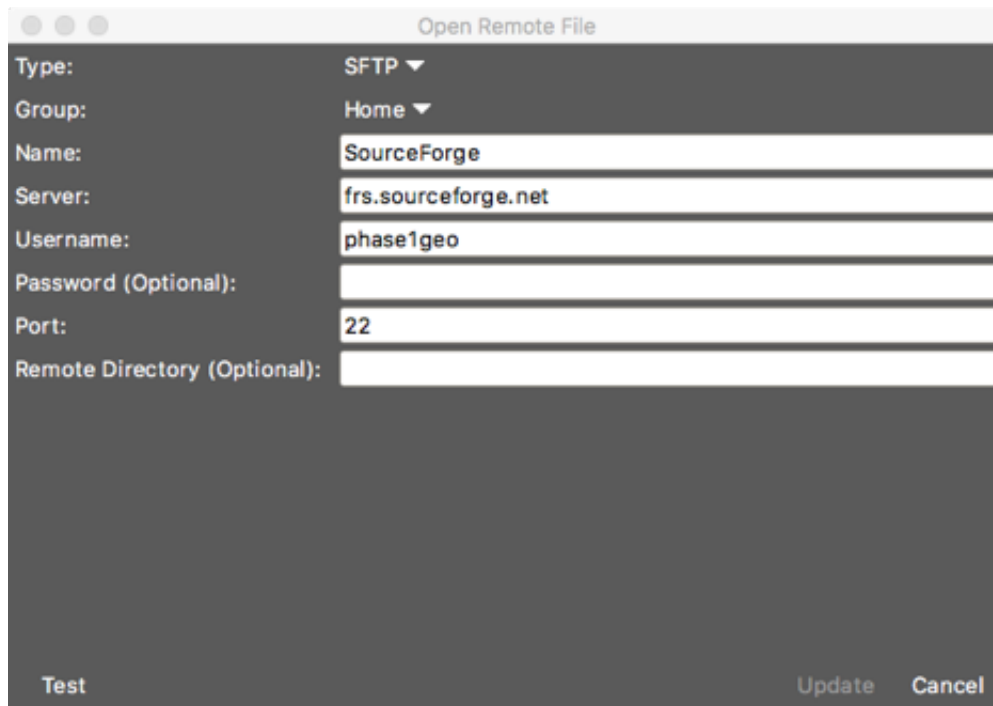
Remote File Viewer

The remote file viewer area of the window displays the currently selected and opened connection directory contents. To open a remote directory or file, navigate to the file/directory, select it and click on the “Open” button in the toolbar. If a directory is selected, the directory will be added to the sidebar in the TKE main window where you can interact with it just like a local filesystem. If a file is selected, an editing buffer will be created displaying the current contents of the remote file and the file’s directory will be loaded into the sidebar of the main TKE window. Any modifications and consequent saves made to this file will automatically be pushed back to the remote server.

To navigate the remote file viewer, double-click on any directory to view the contents of that directory in the viewer. To move up the file system, use the menu button at the top of the viewer area and select an ancestor directory to view. To move backward and forward through the directory history, use the back/forward buttons at the top left of the file viewer.

Connection Editor

The connection editor pane is displayed when adding a new connection or editing an existing connection. The following is a depiction of this pane.



Open Remote File

Type: SFTP ▼

Group: Home ▼

Name: SourceForge

Server: frs.sourceforge.net

Username: phase1geo

Password (Optional):

Port: 22

Remote Directory (Optional):

Test Update Cancel

The “Type” field allows you to specify the protocol to use to connect to the remote server. At present you can connect via FTP, SFTP or WebDAV (if your system supports it — see the installation chapter for details).

The “Group” menu contains the list of created connection groups. Use this to select which group the connection should be associated with.

The “Name” field is what you will refer to the connection as. The combination of the group and name fields must be unique within the TKE session, so you can name the connection anything that you would like. The field is required.

The “Server” field is either the URL server name or the TCP/IP identification of the server to connect to. This field is required.

The “Username” field is the name of the user to use for logging into the server. This field is required.

The “Password” field is the password to use along with the username for logging into the server. This field is optional to specify in the connection. If a password is entered here, opening the connection will not prompt for a password. If a password is not specified, opening the connection will prompt for the password. It is important to note that if a password is entered, it is saved in an encrypted manner such that it is not immediately readable; however, it can be decoded with the proper tools, which is even more important to know if the remote connection file is stored on a service such as Dropbox or iCloud Drive.

The “Port” field specifies the server port that where the connection request will be sent to. The port value will automatically be set to the standard number whenever the “Type” field is changed; however, if the the server uses a different port, you can change that value in this field. This field is required.

The “Remote Directory” field specifies the initial directory that will be displayed after successfully connecting to the server. This field is optional. If not specified, the server will automatically choose the directory to display.

Once all required fields contain a value, you can use the “Test” button to check if the settings allow a successful connection or not. The pass or fail status will be displayed in a popup window. If you get a passing test status, click on the “Create”/“Update” button in the editor to save the connection settings and return the view to the main remote file dialog window. Click on the “Cancel” button to skip saving the settings changes and return to the main remote file dialog window.

Remote Save File Dialog

If you want to save an existing editing buffer to a remote server, select the File / Save Remote As... menu option. This will display the remote save file dialog window. This window will look almost identical to the remote open file dialog window with two exceptions. In the button bar, a “New Folder” button will be available. Clicking on this button when a directory is displayed in the remote file viewer will display a new folder popup window allowing you to specify the name of the new folder. The new folder will be added to the current remote directory.

Additionally, an entry field is displayed allowing you to specify the filename to save to. Clicking the “Save” button will save the specified filename into the currently selected directory. This will also cause the remote directory to be added to the main window sidebar (if it does not already exist there). Once the file has been saved to the remote server, you may continue to edit the file and save it — the saves will be pushed to the remote file server automatically.

Main Sidebar

Once a remote directory is specified in the main window sidebar, you can perform most of the same operations to that remote directory and its related files as you can any local directory. You can open files directly, close files from editing, rename directories/files, duplicate files, delete directories/files, change the directory display, etc. The only operations that you cannot perform is a file difference (since running the file difference would require interacting with a file to perform the difference on the remote server, which is not supported via FTP/SFTP/WebDAV) and making a remote file/directory be a favorite.

In addition, TKE will not save remote directories/files states in-between TKE sessions. This means that if you have opened remote connections and quit TKE, restarting TKE will not automatically reconnect and display previous sidebar and file state.

Sessions

As mentioned in various sections in this guide, TKE has support for named sessions. A session is simply defined as a single TKE window with the following attributes:

- Window geometry and location on the screen
- Fullscreen and zoomed status of the window
- Current working directory
- Command launcher position
- Set of opened directories in the sidebar
- Set of opened files, including information about their pane location, tab location, tab state (i.e., locked, readonly, buffer, language, indent mode, etc.), cursor and yview position
- Markers
- Difference view information including version system and first and second version values
- Global preferences
- Language-specific preferences
- Find, Find/Replace and Find in Files saved search input

Using sessions, you can quickly and efficiently create multiple named sessions and switch between them as the complete saved state of the session is remembered. This means no fiddling with session setups when working on more than one project or area of a project. Less friction with environments means more focus on the work.

By default, TKE will start in an unnamed session; however, at any time the user can save the current setup of TKE (whether in a named or unnamed session) as a new named session using the “Sessions / Save As...” option. This will display an input field at the bottom of the window, allowing the user to specify a name to call the session. Named sessions are persistent on disk, meaning that they will remain available for switching/opening after TKE has been quit and restarted.

Once the session has been saved under a given name, the title bar of the window will include the name of the session. A named session can be modified and resaved using the “Sessions / Save Current” menu option.

If you want to exit a given session, at any time you can use the “Sessions / Close Current” menu option. This will revert the current session back to the last state of the unnamed session.

If you are working in either a named or unnamed session and wish to change the current window to a different session setup, use “Sessions / Switch To” menu option, select one of the available named sessions. Doing so will change the current window to display the last saved state of the named session.

If you are in a Windows or Linux environment, you can open a new window using a given named session using the “Sessions / Open” menu option and select a previously saved name session.

Finally, if you are done using a named session and would like to remove it from disk, simply select the “Sessions / Delete” menu option and select the named session to delete. After selecting the “Yes” option, the named session will be permanently removed.

If you are using TKE from the command-line, you can start TKE in a named session by using the '-s' command-line option. The value passed to '-s' is a name of a session.

Command Launcher

The command launcher provides access to all of the available functionality from anywhere within the application. To call up the launcher, simply hit the key combination (by default, the key combination is Control-Space but this can be changed within the menu bindings file). The resulting widget is a simple entry field displayed in the upper center portion of the window. The cursor will be placed within the entry field for immediate command entry.

To perform a command, simply begin typing the name of the command that you wish to perform. As you enter characters, the command list will be immediately updated with the best matches. The command launcher uses a fuzzy search algorithm for matching that remembers the most used commands based on the input string, allowing you to quickly perform most commands with only a few typed characters.

If one or more matches are found, the top-most entry will be the best match. The best match will also be selected. To execute the best match, simply enter the RETURN key. To change the selection to another displayed match in the list, simply use the up/down arrow keys until the desired command is selected and hit the RETURN key.

The following table describes the types of commands that can be executed within the command launcher along with any special characters that call up specific functionality.

Command Type	Description	Character Sequence
Menu commands	Any menu item commands can be executed from within the launcher.	(Enter any portion of the menu command string)
Clipboard History	Inserts any of the items stored in the clipboard history into the current editor and/or copies the text into the clipboard.	#...
Snippet insertion	Inserts any of the language-specific snippets available for the current editor.	;...
Symbol Jumping	Jump to any supported language symbol (i.e., procedure, function, etc.) in the current editor.	@...
Marker Jumping	Jump to any marker in the current editor.	,...
Sidebar File Open	Open any shown file in the sidebar for editing.	>...
Calculator	Perform numerical calculator expressions (any valid numerical Tcl expression is allowed). Selected result is copied to the clipboard.	(Enter any valid Tcl calculation)
URL launcher	Open a specified URL in the local web browser or recall a previously used URL from history and open that location.	(Enter any valid URL)
URI launcher	Executes the given URI and stores the URI in its history for quickly performing the same function later on. (Ex. <code>dash://tcl.text</code> — opens the Dash application (if installed) and displays the documentation for the Tcl/Tk text widget).	(Enter any valid URI that is supported on your system).
Plugin installation	Displays all available plugins that can be installed. Selecting a plugin in the resulting list installs the plugin.	install
Plugin uninstallation	Displays all available plugins that can be uninstalled. Selecting a plugin in the resulting list uninstalls a plugin.	uninstall
Syntax modification	Changes the syntax highlighting rules for the current editor.	Enter a name of any supported language or <code>Syntax:</code> for a full list of all available languages.
Theme modification	Changes the syntax highlighting color scheme for all editors	Enter a name of any installed theme or enter <code>Theme:</code> for a full list of all available themes.

In addition to the normal command launcher UI (entry field with a list of matching commands listed below), the command launcher also has a preview window that is available for a subset of functionality. The preview window will be displayed below the entry field and to the right of the command list. Highlighting a command in the command list will update the preview window. The preview window is available for the following command launcher functions.

Function	Displayed in Preview
Snippets	Raw snippet content from the snippet file.
Clipboard history	Full content for a paste item.
Plugin installations	Revision and description of the selected plugin.

Additionally, you may move the location of the command launcher widget by grabbing any edge of the launcher and drag it to a new location. If the associated preference value is set, the launcher will display in the new location each time that is invoked. If the preference value is cleared, the launcher widget will display in the default location the next time it is invoked.

Vim Commands

The editor supports a subset of the classic Vim functionality as well as some useful extensions while operating in Vim mode. To edit documentation in Vim mode, go to the “Tools” menu and click on the “Vim mode” option. When the mode is selected in the menu, the editor will respond to Vim input. To place the editor back into standard editing mode, click on the “Vim mode” menu option again. To set the default editing mode, go to the preferences file and set the “Editing/VimMode” value to a value of 1 (for Vim mode) or 0 (for standard mode).

Standard Vim Commands

The following table describes the available standard Vim functionality.

Any characters in bold (ex. b) represent the actual character. Any characters in all caps (ex. ESC or CONTROL-) represent its associated key on the keyboard (these are unprintable characters). Any characters in grey italics (ex. num) font represent variables whose value is described in the description field.

Line Numbers

Command or KEY	Description
.	Specifies current line.
^	Specifies the first line in the file.
\$	Specifies last line in the file.
<i>number</i>	Specifies the line at line number number.
<i>marker_name</i>	Specifies the line marked by <i>marker_name</i> .

Undoing/Cancelling commands

Command or KEY	Description
ESC	Cancels unexecuted command or if in editing mode, ends editing mode to return to command mode. If the current mode is the command mode, any selections or search highlighting is cleared from the current editor.
u	Counteracts last command that changed the buffer.

Repeating a command

Command or KEY	Description
.	Repeats the last command that changed the buffer.
q <i>a-z</i>	Starts recording the following keystrokes to the specified buffer labeled <i>a</i> through <i>z</i> . To stop recording to this buffer, enter q when in command mode. When you are recording, the information bar will display this state information and specify the buffer label storing the keystrokes. If q is entered immediately following the buffer label, it will effectively delete the buffer contents.
@ <i>a-z</i>	Replays the stored keystrokes in the specified buffer.

Editing a file

Command or KEY	Description
:n	Edits the next file in the editor tab order.
:e <i>filename</i>	Edits the specified file (if the file is not already opened, opens the file in a new tab).
:e <i>\#</i>	Edits the previously edited file.
:r <i>filename</i>	Places a copy of the specified file below the current line.
:r <i>!command</i>	Executes the provided shell command and inserts its standard output on the line below the cursor.
gf	Edits the file whose name is under or after the cursor. The file will be added to the sidebar including the contents of its directory.
CONTROL- g	Displays number of lines and characters in the current file in the status bar.

Saving/Closing a file

Command or KEY	Description
:w	Writes file under the original name. If an original name has not been specified, a “Save As” window will be displayed.
ZZ or :wq or :wq!	Writes the file under the original name and closes the current tab. If an original name has not been specified, a “Save As” window will be displayed.
:wq filename or :wq! filename	Writes the file under the given filename and closes the current tab.
:q	Closes the current tab. If the text has been modified since the last save, a prompt will be displayed asking if you would like to save before closing.
:q! or :cq or ZQ	Closes the current tab regardless of the modification status. Changes will not be saved and a prompt will not be displayed.
:w filename	Writes the current file under the specified filename.
:x,yw filename	Writes the specified range of lines to the given filename.
:x,yw! filename	Writes the specified range of lines to the given filename overwriting the contents of the file.

Searching/Replacing

Command or KEY	Description
<i>/string</i>	Finds all occurrences of the given string, jumping the cursor to the first occurrence below the current line.
<i>?string</i>	Finds all occurrences of the given string, jumping the cursor to the first occurrence above the current line.
n	Repeats the last '/' or '?' operation.
<i>numn</i>	Repeats the last '/' or '?' operation <i>num</i> times.
N	Repeats the last '/' or '?' operation in the opposite direction.
<i>numN</i>	Repeats the last '/' or '?' operation in the opposite directory <i>num</i> times.
?	Jumps to the previous occurrence of the previous search.
<i>:x,ys/oldstring/newstring/flags</i>	Finds and replaces one or more occurrences of <i>oldstring</i> with <i>newstring</i> where <i>oldstring</i> can be any Tcl regular expression. The <i>flags</i> value if empty, causes only the first match to be replaced in the given range. The following flags are valid: g = all matches are replaced in the given range, i = ignores case in matching, I = case sensitive matching.
*	Searches the text for the next occurrence of the current word.

Inserting/Replacing text

Command or KEY	Description
i	Inserts before the current character.
a	Inserts after the current character.
A	Inserts at the end of the current line.
I	Inserts at the beginning of the current line.
o	Inserts below the current line (opens new line).
O	Inserts above the current line (opens new line).
r	Replaces the current character (no ESC necessary).
R	Replaces from current cursor position to end of line; does not change characters not typed over.
cw	Replaces the current word.
c <i>char</i>	Replaces all text contained within the pair of char characters before and after the current insertion cursor.
cc	Replaces the current line.
C	Replaces all text from the current insertion cursor to the end of the current line.

Joining text

Command or KEY	Description
J	Joins the current line and the line below it.
<i>num</i> J	Joins <i>num</i> lines, starting with the current line.

Changing case

Command or KEY	Description
~	Changes case of the current character. If text is currently selected, all selected characters will have their case changed.
num~	Changes case of the next <i>num</i> characters starting with the current character.
g~motion	Transforms all text between the cursor and the indicated motion direction to invert the current case.
gumotion	Transforms all text between the cursor and the indicated motion direction to lower case.
gUmotion	Transforms all text between the cursor and the indicated motion direction to upper case.

Moving around in a file

Command or KEY	Description
h	Moves left one character
j	Moves down one line
k	Moves up one line
l	Moves right one character
w	Moves the insertion cursor to the beginning of the next word.
b	Moves the insertion cursor to the beginning of the previous word.
0 or ^	Moves to the beginning of current line
\$	Moves to the end of the current line
:num	Moves to line <i>num</i> (note: <i>num</i> value of 0 or 1 takes you to the first line).
gg	Moves to the beginning of the file.
G	Moves to the end of the file.
numG	Moves to the line <i>num</i> .
RETURN	Moves the insertion cursor to the first non-whitespace character in the line after the current line.
SPACE	Moves the insertion cursor one character to the right, moving to the next line below the current line if the cursor is at the end of the line.
BACKSPACE	Moves the insertion cursor one character to the left, moving to the next line above the current line if the cursor is at the end of the line.

	line if the cursor is at the end of the line.
-	Moves the insertion cursor to the first non-whitespace character in the line before the current line.
H	Moves the insertion cursor to the first line on the screen.
M	Moves the insertion cursor to the middle line on the screen.
L	Moves the insertion cursor to the last line on the screen.
<i>numl</i>	Moves the insertion cursor to the specified column in the current line.
CONTROL-f	Scrolls forward one screen.
CONTROL-b	Scrolls backward one screen.
UP	Moves the cursor one line up.
DOWN	Moves the cursor one line down.
LEFT	Moves the cursor one character to the left.
RIGHT	Moves the cursor one character to the right.

Deleting text

Command or KEY	Description
x or DELETE	Deletes the current character.
<i>numx</i>	Deletes <i>num</i> characters, starting with current character.
X	Deletes the character before the current character.
<i>numX</i>	Deletes <i>num</i> characters before the current character.
dw	Deletes current word.
<i>numdw</i>	Deletes <i>num</i> words, starting with the current word.
dd	Deletes current line (deleted contents are placed in clipboard).
<i>numdd</i>	Deletes <i>num</i> lines, starting with the current line (deleted contents are placed in clipboard).
D	Deletes from current cursor position to the end of the line.
d0	Deletes from current cursor position to the beginning of the current line.
d\$	Deletes from the current cursor position to the end of the current line.
:x,yd	Deletes lines x through y (deleted contents are placed in clipboard).

Copying text

Command or KEY	Description
y	Yanks the current character (yanked contents are placed in clipboard).
numy	Yanks <i>num</i> characters, starting with the current character (yanked contents are placed in clipboard).
yw	Yanks the current word.
numyw	Yanks <i>num</i> words, starting with the current word.
yy	Yanks the current line (yanked contents are placed in clipboard).
numyy	Yanks <i>num</i> lines, starting with the current line (yanked contents are placed in clipboard).
:x,yy	Yanks lines <i>x</i> through <i>y</i> (yanked contents are placed in clipboard).

Pasting text

Command or KEY	Description
p	Places contents in the clipboard below the current line.
P	Places contents in the clipboard above the current line.

Indentation

Command or KEY	Description
>>	Indents the current line by one shift.
<i>num</i> >>	Indents the number of lines (starting at the current line) by one shift.
>i{ or >i}	Indents all lines between the surrounding curly brackets.
>i(or >i)	Indents all lines between the surrounding parenthesis.
>i[or >i]	Indents all lines between the surrounding square brackets.
>i< or >i>	Indents all lines between the surrounding angled brackets.
<<	Unindents the current line by one shift.
<i>num</i> <<	Unindents the number of lines (starting at the current line) by one shift.
<i{ or <i}	Unindents all lines between the surrounding curly brackets.
<i(or <i)	Unindents all lines between the surrounding parenthesis.
<i[or <i]	Unindents all lines between the surrounding square brackets.
<i< or <i>	Unindents all lines between the surrounding angled brackets.
==	Applies automatic indentation formatting to the current line.
=G	Applies automatic indentation formatting from the beginning of the current line to the end of the file.
:set shiftwidth= <i>num</i>	Sets the number of spaces that shifting will use to shift once.
=i{ or =i}	Formats all text between the surrounding curly bracket pairs.
=i(or =i)	Formats all text between the surrounding parenthesis.
=i[or =i]	Formats all text between the surrounding square brackets.
=i< or =i>	Formats all text between the surrounding angled brackets.

Visual (Selection) mode

Command or KEY	Description
v	Changes the mode to visual mode. Using the navigation commands during visual mode will change the current selection.
V	Changes the mode to visual line mode. Using the navigation commands during visual line mode will change the current selection by lines.
vi{ or vi}	Selects all characters between the surrounding curly bracket pairs.
vi(or vi)	Selects all characters between the surrounding parenthesis.
vi[or vi]	Selects all characters between the surrounding square brackets.
vi< or vi>	Selects all characters between the surrounding angled brackets.

Code Folding

Command or KEY	Description
za	Toggles the fold that is at the current line. If no fold marker exists on this line, no action will be taken.
zA	Toggles all folds to the same state.
zc	Closes one level of folding at the current insertion cursor.
zC	Closes all levels of folding at the current insertion cursor.
zo	Opens one level of folding at the current insertion cursor.
zO	Opens all levels of folding at the current insertion cursor.
zR	Unfolds all folded code in the current editing buffer.
zM	Folds all foldable code in the current editing buffer.
zf	When text is selected and we are in manual folding mode, causes the selected text to be folded.
zfnunj	When we are in manual folding mode, causes the following <i>num</i> lines to be folded.
zfnunk	When we are in manual folding mode, causes the previous <i>num</i> lines to be folded.
zd	When we are in manual folding mode and the cursor is in a line that contains a fold indicator, the fold indicator will be removed.
zE	When we are in manual folding mode, all fold indicators are removed.
zi	Toggles all folds by one level to the same state.
zj	Jumps the cursor to the next folded line indicator.
zk	Jumps the cursor to the previous folded line indicator.
zv	If the cursor is within folded code such that it is not currently visible, this command will open enough folds to make the cursor viewable.
:x,yfold	Specifies a range of lines to fold. This is only valid when the foldmethod is set to manual.
:x,yfoldclose[!]	Specifies a range of lines where any opened folds will be closed. If the ! character is specified, all fold levels will be closed; otherwise, one level of folding will be closed.
:x,yfoldopen[!]	Specifies a range of lines where any closed folds will be opened. If the ! character is specified, all fold levels will be opened; otherwise, one level of folding will be opened.

Miscellaneous

Command or KEY	Description
%	Moves to matching (,), { , }, [,], > , < , “ or ‘ character.

Extended Vim Commands

To provide additional functionality to the user, Vim command extensions have been added to the standard list. The following table specifies these Vim command extensions.

Tab or Text Pane Traversal

Command or KEY	Description
:N	Changes to the previous tab.
:p	Changes focus to the tab in the other opened text pane (only available when the other pane exists).

Marker (Bookmark) Creation

Command or KEY	Description
:m	Creates a marker (bookmark) for the current line. This marker can be named in the subsequent entry field that is displayed. Hitting return in the marker entry field will create a named marker (or if no text was typed, an unnamed marker). Hitting the ESC key in the entry field will cancel the marker creation process.
:m <i>marker</i>	Creates a marker (bookmark) for the current line using the provided name.
:cd <i>directory</i>	Changes the current working directory (as displayed in the title bar) to the specified directory.

Multicursor Functionality

Command or KEY	Description
s	Sets a multicursor cursor on the current character. Also makes this character the anchor for any multiline cursor sets.
S	Sets multicursors for every line between the current line and the last multicursor anchor, inclusive. Each multicursor will match the column of the anchor multicursor.
J	When one or more multicursors are set, moves all of the cursors down one line.
K	When one or more multicursors are set, moves all of the cursors up one line.
H	When one or more multicursors are set, moves all of the cursors to the left by one character.
L	When one or more multicursors are set, moves all of the cursors to the right by one character.
\#	When one or more multicursors are set, allows the user to insert an ascending numerical values at each cursor input. See below for details on its usage.

Multicursor Enumeration

When the '#' Vim command is entered after multiple cursors have been set, an entry field called "Starting number:" will be displayed, allowing you to specify the numerical value to insert at the first cursor position. TKE will parse that number and use it for calculating all subsequent cursor positions, incrementing the value by one for each cursor. The following are valid starting number representations:

- *prefix* [0–9]+
 - Inserts prefix followed by a decimal value.
- *prefixd* [0–9]+
 - Inserts prefix followed by a decimal value preceded by "d"
- *prefixb* [0–1]+
 - Inserts prefix followed by a binary value preceded by "b"
- *prefixo* [0–7]+
 - Inserts prefix followed by an octal value preceded by "o"
- *prefix[xh]* [0-9a-fA-F]+
 - Inserts prefix followed by a hexadecimal value preceded by either "x" or "h".

Note: If a value is not specified, a value of zero is assumed.

String/Bracket Insertion

Command or KEY	Description
c'	If a selection exists, all selected code will be encapsulated in single quotes. If no selection exists and current insertion cursor is within a single quote quotation, the right single quote is moved one word to the right. If none of the above is true, the current word is encapsulated in single quotes.
c"	If a selection exists, all selected code will be encapsulated in double quotes. If no selection exists and current insertion cursor is within a double quote quotation, the right double quote is moved one word to the right. If none of the above is true, the current word is encapsulated in double quotes.
c{	If a selection exists, all selected code will be encapsulated in curly brackets. If no selection exists and current insertion cursor is within a curly bracketed code block, the right curly bracket is moved one word to the right. If none of the above is true, the current word is encapsulated in curly brackets.
c[If a selection exists, all selected code will be encapsulated in square brackets. If no selection exists and current insertion cursor is within a square bracketed code block, the right square bracket is moved one word to the right. If none of the above is true, the current word is encapsulated in square brackets.
c(If a selection exists, all selected code will be encapsulated in parenthesis. If no selection exists and current insertion cursor is within a parenthetical code block, the right parenthesis is moved one word to the right. If none of the above is true, the current word is encapsulated in parenthesis.
c<	If a selection exists, all selected code will be encapsulated in angled brackets. If no selection exists and current insertion cursor is within a angle bracketed code block, the right angle bracket is moved one word to the right. If none of the above is true, the current word is encapsulated in angled brackets.

Line Bubbling

Command or KEY	Description
CONTROL-j	Moves the current line down one line, moving the line below the current line above it. If lines are selected, this command moves all of the selected lines down by one line.
CONTROL-k	Moves the current line up one line, moving the line above the current line below it. If lines are selected, this command moves all of the selected lines up by one line.

Deletion

Command or KEY	Description
dn	Deletes all subsequent characters that are numbers.
dN	Deletes all preceding characters that are numbers.
ds	Deletes all subsequent space and tab characters.
dS	Deletes all preceding space and tab characters.

Vim Options

Vim options are settings that apply to either the local editing buffer or all editing buffers and are accessed using the the following command:

```
:set option?=value? ?option=value...?
```

Where the value of option (and optionally value) corresponds to any of the following values.

Option	Values	Default	Scope	Description
autochdir/acd, noautochdir/noacd	None	off	Global	When set, the current working directory will automatically change to be the directory containing the currently active file and will change whenever the user makes a new file the active file.
autoindent/ai,noautoindent/noai	None	off	Local	When set, the indentation mode of the current editing buffer will be set to auto-indent (IND) mode. When unset, the indentation mode of the current editing buffer will be set OFF.
browsedir/bsdir	last, buffer, current, directory	last	Global	When the open file/directory dialog box is displayed, this value dictates the starting directory that will be displayed. (last = Last used directory; buffer = Directory containing the current file in the editing buffer; current = Current working directory; directory = Uses the specified pathname as the starting directory location) The default value for this option can be changed using the

				General/DefaultFileBrowserDirectory preference option.
expandtab/et, noexpandtab/noet	None	on	Local	When set, forces the use spaces instead of tabs when the TAB key is pressed. The number of spaces is determined by the value of the tabstop option (if specified) or the Editor/SpacesPerTab preference value. When unset, forces the use of tabs when the TAB key is pressed.
fileformat/ff	dos, unix, mac	auto determined	Local	Overrides the end-of-line character that is used when saving an editing buffer. By default, this value is determined by Editor/EndOfLineTranslation preference setting.
foldenable/fen, nofoldenable/nofen	None	None	Local	This is option is only valid when the foldmethod is set to manual. If enabled, all existing folds are closed; otherwise, if unset, all existing folds are opened.
matchpairs/mps	{:}, (:), [:], <:>	determined by language	Local	Specifies character pairs that specify auto-completion characters. (Ex: <code>set mps+=<:></code> to add angled brackets; <code>set mps--=(:),[:]</code> to remove parenthesis and square brackets); <code>set mps={:}</code> to use only curly brackets).
modeline/ml, nomodeline/noml	None	on	Local	When set, TKE will use any Vim modelines specified at the top of the file. When unset, TKE will ignore Vim modeline syntax.
modelines/mls	Num	determined by preference value	Global	Specifies the number of lines starting at the top of the file that TKE will search for Vim modeline syntax. This value overrides the default value from the Editor/VimModelines preference value.
modifiable/ma, nomodifiable/noma	None	on	Local	When set, sets the file lock status to locked. When unset, sets the file local status to unlocked.

modified/mod, nomodified/nomod	None	off	Local	When set, causes the status of the editing buffer to indicate that it is currently modified. When unset, clears the modified state of the editing buffer.
number/nu, nonumber/nonu	None	on	Local	When set, displays line numbers. When unset, hide the line numbers from view.
numberwidth/nuw	Num	4	Global	Specifies the minimum width of the line number gutter in characters.
relativenumber/rnu, norelativenumber/nornu	None	off	Local	When set, displays the line numbers in relative numbering format. When unset, displays the line numbers in absolute numbering format.
shiftwidth/sw	Num	determined by preference value	Local	Specifies the number of spaces to use when a left or right shift operation or an indentation/unindentation occurs. This overrides the default value specified with the Editor/IndentSpaces preference value.
showmatch/sm, noshowmatch/nosm	None	on	Global	Specifies whether a matching bracket/quote character will be automatically highlighted when the cursor is on the associated bracket/quote character.
smartindent/si, nosmartindent/nosi	None	on	Local	When set, the indentation mode of the current editing buffer will be set to smart-indent (IND+) mode. When unset, the indentation mode of the current editing buffer will be set to OFF.
splitbelow/sb, nosplitbelow/nosb	None	off	Local	When set, splits the current editing buffer to provide two views of the same file. When unset, removes split view from the current editing buffer.
		auto determined		Overrides the default language syntax highlighting to apply to the

syntax/syn	Lang	by file extension	Local	current editing buffer with the given language.
tabstop/ts	Num	determined by preference value	Local	Specifies the number of spaces that a TAB in the file counts for.

Vim Modelines

By default, TKE will parse the first few lines of each opened file for Vim modeline syntax. If a valid modeline is found, the recognized Vim options within the modeline are parsed and applied. A valid modeline is one of the following formats.

Mode Line Syntax
vi:set <i>opts</i> :
vim:set <i>opts</i> :
vim <i>version</i> : set <i>opts</i> :
vim < <i>version</i> : set <i>opts</i> :
vim = <i>version</i> : set <i>opts</i> :
vim > <i>version</i> : set <i>opts</i> :
ex:set <i>opts</i> :
vi: <i>opts</i>
vim: <i>opts</i>
vim <i>version</i> : <i>opts</i>
vim < <i>version</i> : <i>opts</i>
vim = <i>version</i> : <i>opts</i>
vim > <i>version</i> : <i>opts</i>
ex: <i>opts</i>

The value of *opts* is a list of Vim options separated by colon or space characters. The spaces listed above are required, including the space prior to the beginning of the Vim options. The *version* value is disregarded though the syntax is parsed.

As an example of a valid Vim modeline, consider the following line which sets the tabstop value to 4, adds an angled bracket match pair and sets line numbering to relative.


```
// vim:ts=4 mps+=<\:> rnu
```

Only local options will be used in the Vim modeline. Global options will be ignored without error.

Snippets, Emmet and Templates

Snippets allow the user to enter a short bit of text (herein called the *abbreviation*) which will be replaced by a larger piece of text (called the *snippet*) when a whitespace character (selectable in the preference file) is entered. For example, suppose we have defined an abbreviation called “hw” which is assigned the snippet text “Hello, world!”. If we enter the following string in an editor:

```
cout << "hw
```

and follow it with hitting either the SPACE, RETURN or TAB key, the editor will replace the abbreviation to look like the following:

```
cout << "Hello World!
```

In addition to simple ascii text, the snippet text can contain various styles of variables. For example, suppose we are editing a file called “foobar.cc” and have defined an abbreviation called “cf” which is assigned the snippet text “\$FILENAME”. If we enter the following string in an editor:

```
File: cf
```

and follow it with hitting either the SPACE, RETURN or TAB key, the editor will replace the abbreviation to look like the following:

```
File: foobar.cc
```

Snippet Variables

The following table represents the various variables that can be used within snippet text. Note that all variables are expanded at the time the snippet replacement occurs. Additionally, the DOLLARSIGN (\$) and BACKTICK (`) characters are special characters. If you require these characters to be treated as literal characters in your snippet, you will need to escape these characters by placing a BACKSLASH (\) character just before it.

Variable	Description
\$SELECTED_TEXT	Inserts the currently selected text at this variable's location. If no text is currently selected, an empty string is inserted in its place.
\$CLIPBOARD	Places the contents that are currently in the clipboard at this variable's location.
\$CURRENT_LINE	Places the current line contents (minus the abbreviation) at this variable's location.
\$CURRENT_WORD	Places the current word at this variable's location.
\$DIRECTORY	Places the current directory at this variable's location.
\$FILEPATH	Places the current file pathname at this variable's location.
\$FILENAME	Places the root file name at this variable's location.
\$LINE_INDEX	Places the position of the current insertion cursor (specified as <i>line.column</i>) at this variable's location.
\$LINE_NUMBER	Places the line position of the current insertion cursor at this variable's location.
\$CURRENT_DATE	Places the current date at this variable's location. The date is specified as MM/DD/YYYY.
\$CURRENT_TIME	Places the current time at this variable's location. The time is specified as HH:MM AM/PM.
\$CURRENT_MON	Shortened name of the current month (ex., Jan, Feb).
\$CURRENT_MONTH	Long name of the current month (ex., January, February).
\$CURRENT_MON1	Numerical value for the current month expressed as either a one or two digit value.
\$CURRENT_MON2	Numerical value for the current month expressed as a two digit value where the first digit will be a zero, if needed.
\$CURRENT_DAYN	Shortened name of the current day of the week (ex., Mon, Tue).
\$CURRENT_DAYNAME	Long name of the current day of the week (ex., Monday, Tuesday).
\$CURRENT_DAY1	Numerical day of the current month expressed as either a one or two digit number.
\$CURRENT_DAY2	Numerical day of the current month expressed as a two digit number where the first digit will be zero, if needed.
\$CURRENT_YEAR2	Two digit representation of the current year where the first digit will be zero, if needed.
\$CURRENT_YEAR	Four digit representation of the current year.

Variable	Description
DOLLARSIGN0	Places the cursor at this variable's location after the entire snippet has been expanded.
<i>\$number</i>	Places the cursor at this variable's location in the order of number. Hitting the TAB key will jump the cursor to the next cursor stop. For example, if a snippet uses the variables "\$1 ... \$2", the cursor will first be placed at location "\$1" and when the TAB key is pressed, the cursor will jump to the location of "\$2". If more than one "\$1" is use within the same snippet, the text that is entered in the first occurrence of this variable will also be entered in all other places within the snippet that share the same number.
<i>\${number:value}</i>	Places the cursor at this variable's location in the order of number, placing the string value at the cursor's location. The string value can be used as a placeholder to remind the user what information to insert at that location. The value string will be automatically selected when the snippet is inserted so that immediately typing text will delete the value string with the user's entered string.
<i>`shell_command`</i>	Executes the specified command between the back tick characters.
<i>\${number/pattern/format/opts}</i>	The value of number must be a tabstop previously specified in the snippet. Its value is run through a regular expression match against pattern (the specified options are passed to the regular expression parser (values of g, i and l are supported; see the Vim command for search/replace for a description of these flags). The resulting matches are used with format and the resulting value is inserted. See the following table for a description of format strings.
<i>\${variable/pattern/format/opts}</i>	The value of variable must be one of the above variables (minus the starting dollar sign). Its value is run through a regular expression match against pattern (the specified options are passed to the regular expression parser (values of g, i and l are supported — see the Vim command for search/replace for a description of these flags). The resulting matches are used with format and the resulting value is inserted. See the following table for a description of format strings.

Transform Format

A variable or mirror value text transformation is possible using the last two documented commands in the prior table. The value of format contains the resulting text that will replace the entire transformation string within the snippet. The following table describes valid syntax that can be used in this field.

Any numbers represented in the format text (preceded by a '\$' character when used on its own) refer to matched values in the transformation match pattern (represented by "(...)" regular expression syntax). Each match will be assigned to a corresponding match variable which can be referenced using '\$' followed by its match number. Any match variables will be substituted with their matched value (or the empty string if the match variable was not assigned).

Syntax	Description
<i>text</i>	Any normal text can be specified. The following characters are special and must be escaped with a BACKSLASH character if the literal value is required: ‘(, ’ and “.
l	The case of the character immediately following this character sequence will be changed to lower case.
u	The case of the character immediately following this character sequence will be changed to upper case.
L...E	The case of all characters between these character sequences will be changed to lower case.
U...E	The case of all characters between these character sequences will be changed to upper case.
(?number:...)	If the corresponding match variable was assigned a value, substitutes this syntax with the format text found to the right of the ‘:’ (colon) character.
(?number:...:...)	If the corresponding match variable was assigned a value, substitutes this syntax with the format text found to the right of the first ‘:’ (colon) character; otherwise, substitutes this syntax with the format text found to the right of the second ‘:’ character.

Creating Snippets

Snippets are maintained in individual files according to the syntax language of the buffer that uses them. Therefore, all Tcl snippets will be placed in a Tcl.snippets file within your ~/.tke/snippets directory while C++ snippets will be placed in a C++.snippets file in the same directory. In addition, each user also has a global snippets file which is available in all buffers. These snippets are stored in ~/.tke/snippets/user.snippets.

To create or edit a snippet for a specific language, make sure that the current editor language is set to the language of the snippet being created or modified and select the “Edit / Snippets / Edit Language” menu command. This will add the language-specific snippets file from your ~/.tke/snippets directory via the preferences window. If the file doesn’t yet exist, TKE will automatically create it for you.

The following image shows what the snippets editing pane looks like.

Search Snippets		Language Tcl ▼
Keyword	Snippet	
pplace	::tk::PlaceWindow \$1 widget \${2:}	
hheader	# TKE - Advanced Programmer's Editor # Copyright (C) 2014-2016 Trevor Williams (phase1geo@gmail.com) # ...	
proccom	#####...	
ccenter	set screenwidth [wininfo screenwidth .swizwin] set screenheight [wininfo screenheight .swizwin] set width [wininfo width .swizwin] ...	

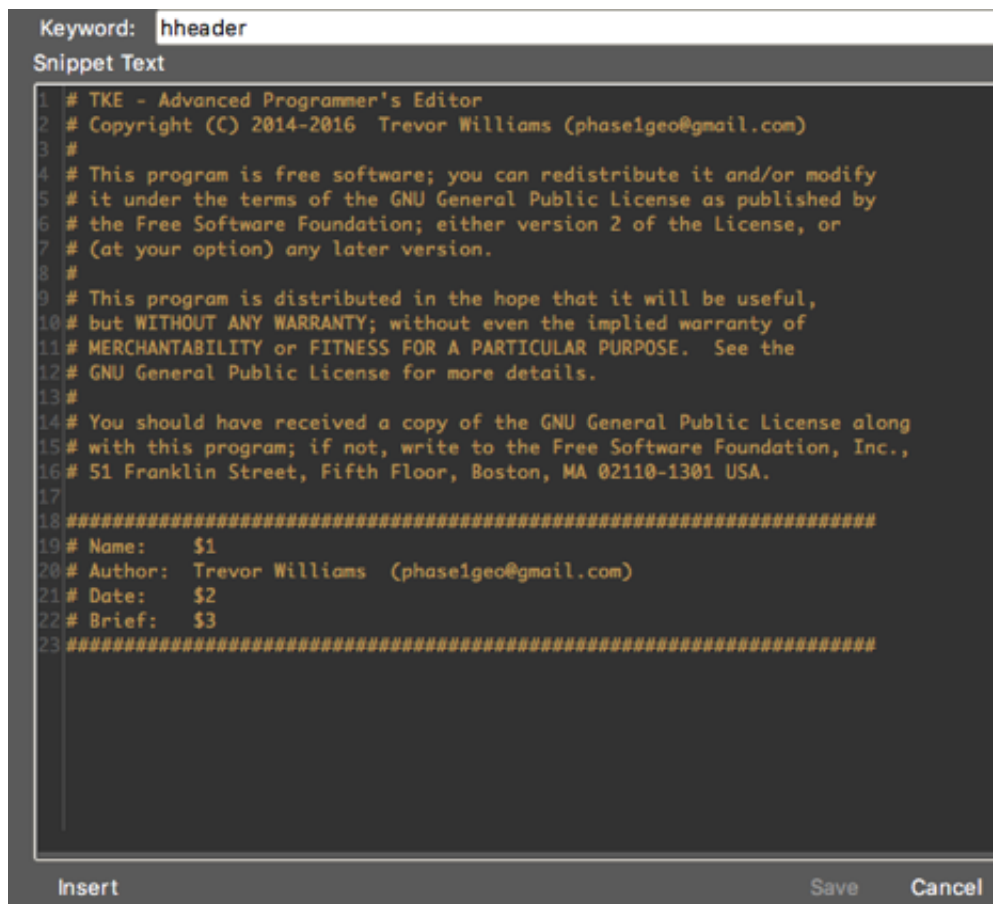
Add Delete

At the top left is a search bar that will display all snippet text within the table that matches the entered text. The search results are updated as you type. To view all items in the table, clear the search text.

At the top right is the language selection menu. To view/edit snippets in a different language, select the menu button and select a new language from the available list. If you would like to create a language agnostic snippet (i.e., a snippet that is available from any language), select the language menu button and choose the <All> language option.

The main table displays the available snippets in the given language along with their associated keyword and the first four lines of the snippet text. To view/edit the snippet, double-click the snippet in the table. This will display the snippet editor pane within the preferences UI. To add a new snippet, click the “Add” button (which will also display the snippet editor pane). To delete a snippet, select the snippet to delete and click the “Delete” button. A confirmation will be displayed to confirm the deletion.

The snippet editor pane is depicted as shown.



The top entry allows you to associate a snippet keyword. Whenever this keyword is entered in an editing buffer, the keyword will be replaced with the associated snippet text. Make sure that the keyword that you use is meaningful but not a string that you normally enter when editing files.

The main editing area allows you to enter the snippet text. All code entered in the text field will be syntax highlighted in the current language to help with readability. In addition, auto-indentation will be applied to the file using the syntax rules and preference settings. It is important to note that when the snippet is inserted in an editing buffer that the inserted text will be automatically indented to match the current indentation.

At the bottom of the snippet editor pane is the button bar. The “Insert” button will display a menu containing a series of snippet syntax that can be inserted into the text field. Selecting an item from this list will insert the item at the current insertion cursor position. If current snippet has both a valid keyword and a non-empty snippet string entered, the “Save” button will be enabled. Clicking this button will save the snippet and return you to the snippet table view. To cancel adding/editing the current snippet, click on the “Cancel” button. This will also return you back to the snippet table view.

As soon as a snippet has been edited and saved, it will be immediately ready to be used within the application.

Emmet

Emmet is primarily a syntax that allows for the creation of HTML/XML syntax as well as CSS syntax. It's minimalistic nature allows for quick generation of lots of code with a minimal number of input characters. This document will not attempt to describe the Emmet syntax (which can be found at [<http://docs.emmet.io>](http://docs.emmet.io) other than to state that TKE has full, built-in support for the Emmet abbreviation syntax for HTML/XML, Ipsum Lorem text insertion, and full support for CSS syntax.

Once an Emmet abbreviation has been entered in an editing buffer, make sure that the cursor is located at the right-hand side of the text and use the Edit / Emmet / Expand Abbreviation file option to expand the syntax (or use Control-E which is the default key binding for this option). If there is an error in the syntax, no expansion will be performed; otherwise, the abbreviation will be removed and its generated content will be inserted in its place (if the generated results span multiple lines, those lines will be preceded by the proper amount of whitespace). Additionally, any tabstop points in the generated text will cause the insertion cursor to be placed at the first tabstop and hitting the TAB key will jump the cursor to the next tabstop until all tabstops have been traversed.

Additionally, you can create your own Emmet abbreviations using the Emmet / Edit Custom Abbreviations menu item. This will display the custom abbreviation file in an editing buffer. The contents of this file are self-documented. Saving the editing buffer will immediately update the available Emmet abbreviations such that restarting the application will not be necessary.

Templates

Templates are essentially files containing valid snippet code. When a template file is created, a new file can be created using this template file as a starting point and variable substitutions and other information in the file can be preset using snippet variables and other snippet syntax. Template files are specially managed by TKE and are saved in the user's ~/.tke/templates directory (though you should not have to deal with this directory).

To create a new template file, simply create a new file with the needed text or use an existing file and use the "File / Save As Template..." menu option. This will display an input text field allowing for a name to be used for the template file. Any name can be used, but if you add a valid file extension that TKE recognizes for syntax highlighting, when the file is created based on the template, that extension's syntax highlighter will automatically be used on the new file (even though the name of the file will be set to "Untitled" until it is saved).

To create a new file based on a template, choose the "File / New From Template..." menu option. This will display the template chooser. A list of available templates are displayed and the currently selected file is displayed in the viewer panel of the window. Left-clicking or selecting and hitting the RETURN key will add a new file editing buffer, insert the template information in the new file, perform any snippet variable substitutions, and position the cursor in the first input area. Fill in the file just as you would in a snippet.

To edit the contents of a snippet, use the "Edit / Templates / Edit" menu. This will display the template chooser window. Select any one of the available templates using the left mouse button or by hitting the RETURN key on a selected template name. This will add the template to the editing buffer where you can edit the template file as you would any other file.

To delete an existing template, use the “Edit / Templates / Delete” menu. This will display the template chooser window. Select any one of the available templates using the left mouse button or by hitting the RETURN key on a selected template name. A confirmation window will be displayed to confirm the deletion. Clicking the “Yes” button will permanently delete the template.

Preferences

Preferences allow you to customize your experience with TKE by modifying various behaviors and/or appearances within the tool. TKE preferences are handled by four types of files:

1. Base preference file (located in the TKE installation directory — in data/preferences.tkedat)
2. User preference file (located at ~/.tke/preferences.tkedat)
3. User language preference files (loaded in the ~/.tke directory)
4. Session files (preference data is stored inside the session file itself)

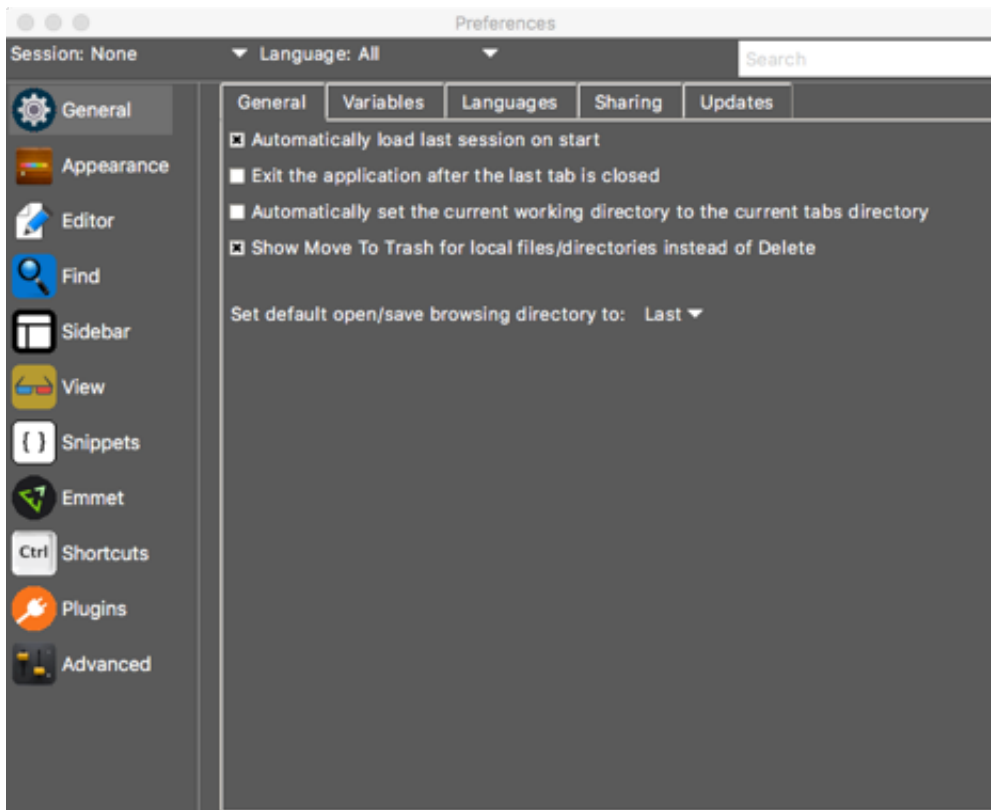
The preferences files are read and handled at two event times: when TKE is started and when the user preference file is written/saved. If the user preference file does not exist, the base preference file is copied and to the user's ~/.tke directory and the resulting file is read and its values used. If the user preference file already exists, its modification timestamp is compared to the timestamp of the base preference file — if the base file is newer than the user preference file, the base preference file is read in, the user preference file values are used in place of the base preference file and the resulting content is written back out to the user preference file. This makes sure that the user's preference file is always up-to-date with the preferences currently available in the tool. If the user preference file's modification time is the same as or newer than the base preference file, the user preference contents are used to configure the tool.

When a file is being edited that contains a previously created language preference file, that set of preferences override the user preference values for all tabs that use the given language.

If a session is currently being used, the preferences associated with the session will be used instead of the user preference file. If file with a saved language preference file is being edited within the session, the language preferences within the session will override the session preference file.

Editing Preference Information

The GUI preference window contains three main panes depicted in the following image.



At the top of the preference window is an area which allows you to select which preference types to modify. The **Session** menu button allows you to select whether you are editing the general user preferences (using the value of “None”) or preferences for a specific existing session. To change the main preference setting, simply click on the menu button and select the desired option. The **Language** menu button allows you to specify the preferences for any tabs opened with a specific language. Selecting a value of “All” will edit preferences that will be true for any language; however, selecting a language will display and save preference settings for the specific language chosen. Note that the session and language menu buttons are used in concert with each other to set the specified setting.

On the right side of the top panel is the search entry field. Use this field to quickly find a given preference value within any valid category pane in the GUI. The search value can be any set of printable characters where a “*” can serve as a wildcard character or set of characters. All matching search results will be immediately displayed and updated while typing just below the search field. In addition, the category tabs in the left sidebar will be highlighted showing which tabs contain matches to the current search string. Selecting a match from the match list will immediately change the preference GUI to display the pane containing the preference item.

On the left side of the preference GUI is the preference category sidebar. This sidebar contains the list of available preferences for the currently selected session/language preference item displayed at the top of the window. Click on a category to display the associated preference pane in the area to the right of the sidebar.

The main area to the right of the sidebar is the preference pane area. In this area, all available preference items associated with the currently selected preference category will be displayed. In some categories, the preference pane will be further broken down into subcategories using a simple tabbed interface towards the top of the preference pane. Clicking on a tab will display the preference items associated with the subcategory.

It is important to note that depending on the session/language setting, the available preference categories may change. Generally, the following are true:

- All available categories will be displayed when the session is set to “None” and the language is set to “All”. This is the general user settings.
- If the session is set to a non-“None” value and the language is set to “All”, the “General” and “Advanced” categories will not be available for modification.
- If the language is set to a non-“All” value, only the “Editor” category will be displayed in the UI. This is accomplished by hiding the sidebar and using the entire lower portion of the window for editor preference display.

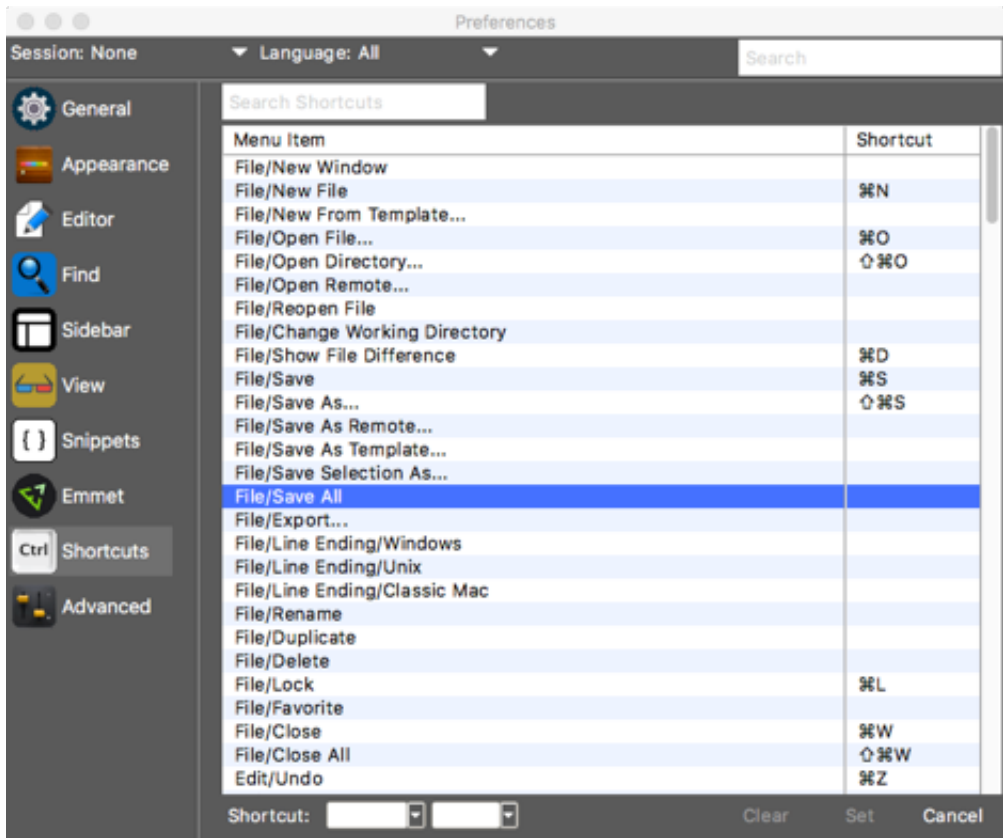
Any change made within the preferences window will be immediately applied/saved. When you are finished making preference changes, close the window using the window manager close option.

Menu Binding

The menu binding capability within TKE simply allows any user to customize the keyboard shortcuts to launch any menu command. By default, TKE contains a minimal set of menu bindings; however, any of the menu items can be overridden.

The default (global) menu binding file is located in the TKE installation directory (in `data/bindings/menu_bindings.windowingsystem.tkedat`). In addition to the global file, each user will have their own menu binding file which overrides the global file settings. This file is located at `~/.tke/menu_bindings.windowingsystem.tkedat`.

You can change the menu shortcuts by either starting the Preferences via the “Edit / Preferences / Edit User - Global” menu option and selecting the Shortcuts panel or via the “Edit / Menu Bindings / Edit User” menu option which will display the preferences GUI with the Shortcuts panel immediately displayed. A representation of the Sharing panel is displayed below.



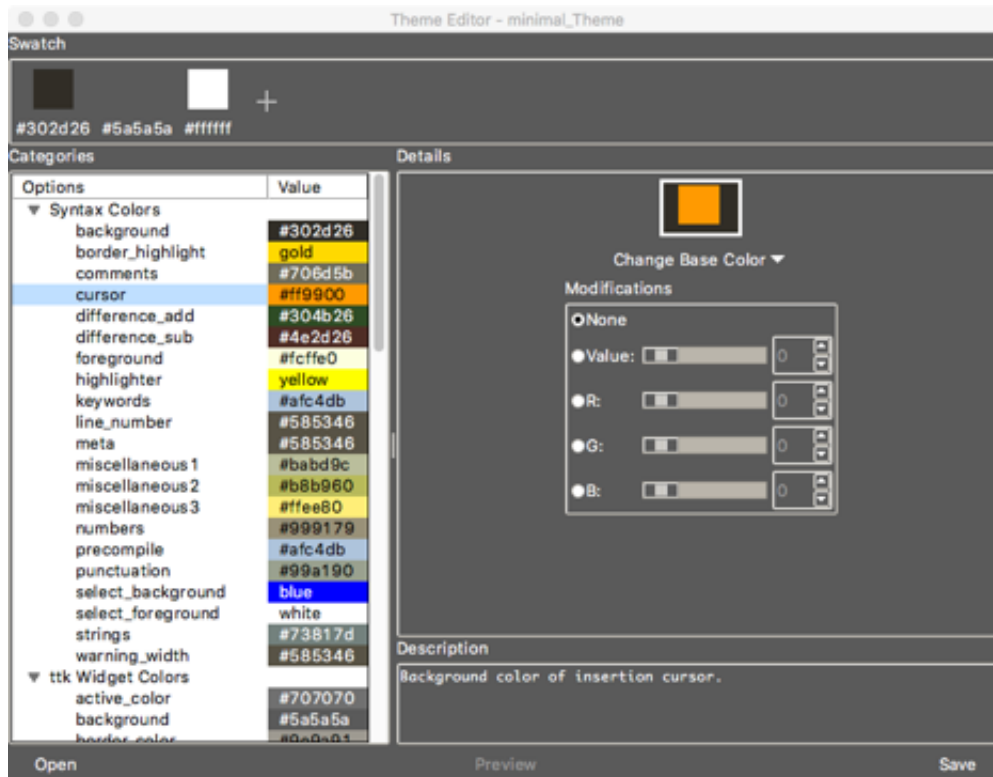
To the upper left, there is a search entry field which allows you to show only menu items in the table that match the given search criteria. The table will be updated as you type. To show all elements in the table, clear the search text.

To add or change a shortcut value, select a menu item in the shortcut table. This will display the shortcut editor below the table. Simply select a modifier combination and key via the two dropdown lists (the values available in the list boxes are automatically updated to guarantee that any available values will result in a unique key combination for the shortcut) and click on the “Set” button. To remove an existing shortcut, select it in the table and click on the “Clear” button. Click on the “Cancel” button or another table item to remove the menu item from being edited.

To quickly edit the shortcut on Linux systems, you can hold down the Control button while selecting any menu item in the main menu. This will automatically display the preference GUI and select the given menu item for editing purposes in the Shortcuts panel.

Theme Editor

The Theme Editor is a GUI interface for creating, editing, importing and/or exporting themes within TKE. To access the theme editor, select the “Tools / Theme Editor” menu command. The window displays a preview of the syntax highlighting scheme in a sample window. The following image is a representation of this window.



Interface

The theme editor interface contains 5 basic UI elements:

- Title bar
- Color swatch editor
- Category table
- Option detail pane
- Option description pane
- Button action bar

Title Bar

The title bar displays the name of the theme that is currently being edited. If the current theme was previously imported and contains attribution information (i.e., the name of the creator and/or creator’s website) the attribution information will also be displayed in the title bar.

In addition to this theme information, the title bar also displays the modified status of the theme. Whenever you make a savable change to the theme, the character between “Theme Editor” and the name of the theme will change to an asterisk (*). When the theme is saved, this character will be changed to a dash (-) character indicating that the theme contains no unsaved changes.

Finally, to exit the theme editor, use the title bar window quit button. If the theme has unsaved changes, a prompt will be displayed allowing you to save the changes prior to exiting the window (or you can cancel the quit operation through this prompt as well).

Color Swatch Editor

To help with color editing, the theme editor provides an area at the top of the window which can optionally contain one or more colors that are deemed to be important colors in the theme. These colors are saved as part of the theme file such that they will be displayed in the swatch area whenever the theme is viewed in the theme editor. Swatch colors can be used directly or indirectly in the color assignment of various UI elements. This means that all of these UI elements that are dependent on a swatch color will be changed whenever the swatch color is changed.



Within the swatch editing area, each swatch is displayed by color and by color name. Within various theme editor menus, color swatches are only displayed by color name. The swatch area helps map the name to the color.

To add a swatch color simply click on the “+” button in the swatch area. This will display a color picker window. Select the desired color using this window and click the “OK” button to add the swatch color. Alternately, you can make any category color a swatch color by selecting the color row in the category table and click the “+” swatch button. This will also display the color picker window but the default color will match the selected color in the table, allowing you to save an exact color more quickly. You can add up to 8 swatch colors per theme.

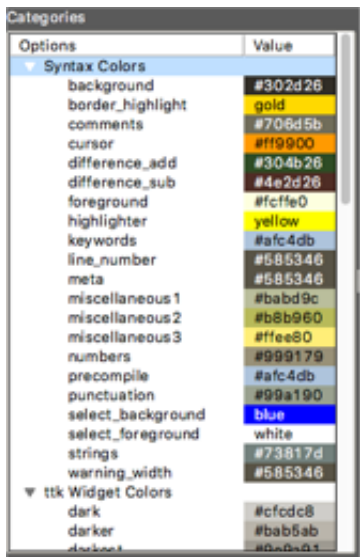
To edit a swatch color, simply left-click on the swatch color. This will display a color picker window where the default color will be current swatch color. Choosing a different color and clicking the “OK” button will change the swatch color and any other theme colors that are dependent on the swatch color.

Finally, to delete an existing swatch color, right-click on the swatch color. A confirmation window will be displayed. Clicking the “Yes” button will confirm the deletion. If there were any colors that were dependent of the color, their colors will not be altered but they will no longer be dependent on the given swatch color.

Category Table

The category table is displayed on the left side of the window and displays all of the UI elements that can be customized as part of the theme. They are organized in the table by category. Left-clicking on the disclosure triangle of the category will show/hide all options within the category. Left-clicking on a category option row will select the row and display the

associated option information in the option detail pane which allows the options value to be changed. Each category option displays the name of the option as well as displays the current value (and potential representation of the value, if possible).



Options	Value
▼ Syntax Colors	
background	#302d26
border_highlight	gold
comments	#706d5b
cursor	#ff9900
difference_add	#304b26
difference_sub	#4e2d26
foreground	#fcffe0
highlighter	yellow
keywords	#afc4db
line_number	#585346
meta	#585346
miscellaneous1	#babd9c
miscellaneous2	#b8b960
miscellaneous3	#ffee80
numbers	#999179
precompile	#afc4db
punctuation	#99a190
select_background	blue
select_foreground	white
strings	#73817d
warning_width	#585346
▼ ttk Widget Colors	
dark	#cfdcd8
darker	#bab5ab
darkest	#808080

There are a few operations that can be performed on the category table which are accessible by left-clicking on the header of the table. This will display a menu containing the following features:

Category Table Search

To help find a specific category option, the category table provides a simple search mechanism. Clicking on the **Table Search** option will display the search entry field just above the category table. Entering text in this field will modify the displayed category options that match the current search string. If you hit the **Return** key while entering text, the current text will be selected. If text is selected, entering an additional character will delete the search string and replace it with the entered character. If you hit the **Escape** key, the search field will be closed and all options in the category table will be displayed. You can also select the **Table Search** option in the table menu a second time to hide the search field as well.

Category Table Filtering

To help make viewing or searching for items in the category table simpler, it also provides for some basic filtering functionality. The filter submenu contains the following filtering capabilities:

Menu Item	Description
Show All	Displays all items in the table. This filtering option is useful when the table is hiding information from a previously applied filter.
Show Category	Displays a submenu containing a list of all theme categories. Selecting a category in this list will hide all other categories and display only the selected category.
Show Color	Displays a submenu containing a list of all colors assigned to UI elements within the theme. The first colors displayed will be the swatch colors, followed by a menu separator, followed by all other colors. Selecting a color will display only those options whose color matches the chosen color.
Show Selected	Value This option will only be valid when an option row is selected in the category table. All options in the table that have a value that match the selected row's value will be displayed.
Show Selected	Option This option will only be valid when an option row is selected in the category table. All options in the table that have the same option name as the selected row will be displayed.

Category Table Copy/Paste

To help make creating new themes easier, the theme editor also allows you to copy multiple theme options from one or more themes (all options are appended to the theme copy buffer) and consequently paste those options into a new theme.

To copy items, you will need to put the category table into copy mode by selecting the category table menu's **Enable Copy Mode** option. Selecting this option will display two buttons below the category table: **Copy** and **Close**. Once the table is in copy mode, you can select/deselect options by left-clicking on the option. Additionally, you can select/deselect an entire category's options by left-clicking on the category row in the table. Once you have the options selected that you want to copy, click on the **Copy** button to add these to the theme copy buffer (this buffer is separate from the clipboard). You can continue to select and copy in the same manner. When you are done copying items in the category table, click on the **Close** button to exit copy mode and return the table to normal operation.

To paste the current theme copy buffer to a theme, open the theme and choose the **Paste Theme Items** option in the category menu. This will update the current theme options and empty the theme copy buffer. You can also empty the theme copy buffer by closing the theme editor and redisplaying it.

Option Detail Pane

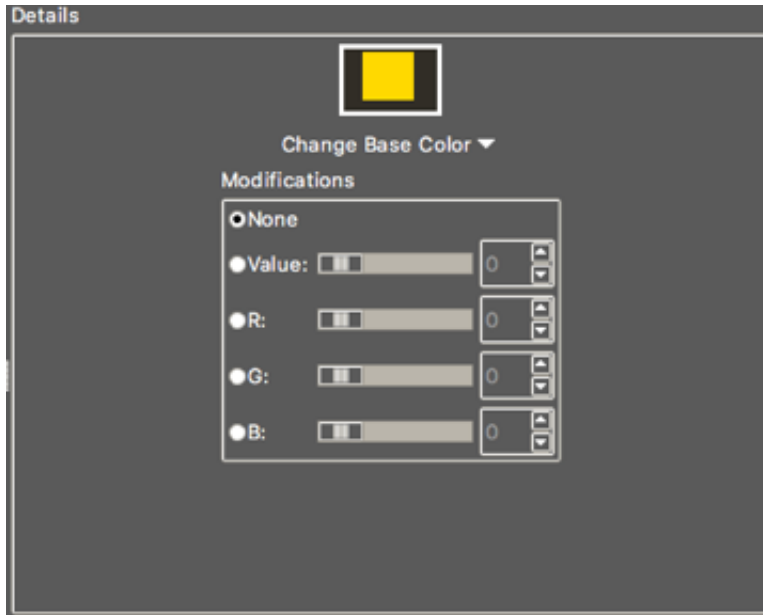
The option detail pane displays the option editor UI for the selected row in the category table. Any changes made within this pane will be immediately reflected in the category table. The detail pane will change its UI dependent upon the type of option being edited. Currently, there are 4 types of options can be changed:

1. Color
2. Number
3. Relief
4. Image

Each detail pane type is described in the following subsections.

Color

The color pane allows for editing a single color by either selecting a color through a standard color chooser, selecting a swatch color and, optionally, modifying the selected color with some simple modifications. A representation of the color pane is shown below.



The top image shows two colors. The background color is the base color that the given color will be surrounded by. Showing the base color along with the color being edited should help in choosing an appropriate color. The foreground color is the color being selected.

Below the color preview is the main color chooser. Left-clicking the button will allow you to either select a color with a standard color picker or select one of the swatch colors.

Below the color picker is the color modification panel. Choosing the “None” option will use the color as is. Choosing the “Value” option will change the current color’s HSV value by the given number. You can change this value by either using the slider, editing the value in the entry field or using the up/down buttons. As you change this value, the foreground preview color will display two colors. The color on the left is the selected color, the color on the right is the color with the modification applied (the actual color that will be used). The “R”, “G”, or “B” options, if selected, will change the color by the specified color value. You can only use one modification on the selected color. You are not permitted to combine modifications.

Any color changes made in this pane will be immediately reflected in the category option table.

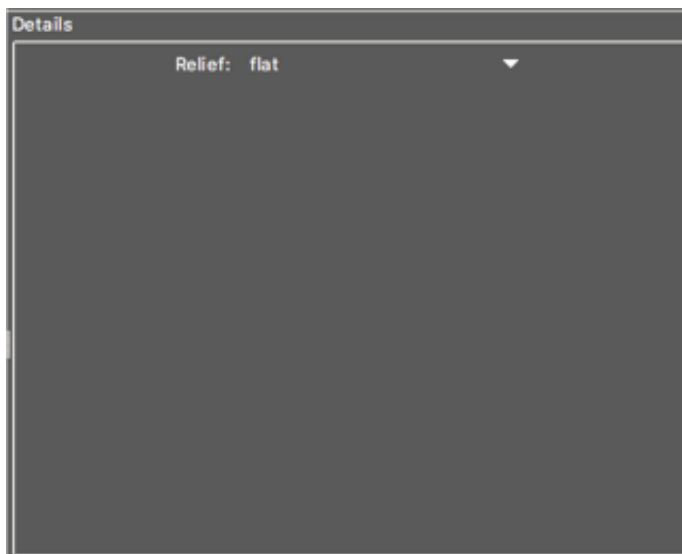
Number

The number pane allows you to modify the value of an option that has a numerical value. You can modify the value using the entry field and/or up/down arrows. Any change made in this pane will be immediately reflected in the category option table. A representation of the number pane is displayed below.



Relief

The relief pane displays a drop-down list allowing you to select from the available relief values. Any changes made will be immediately reflected in the category option table. A representation of this window is shown below.



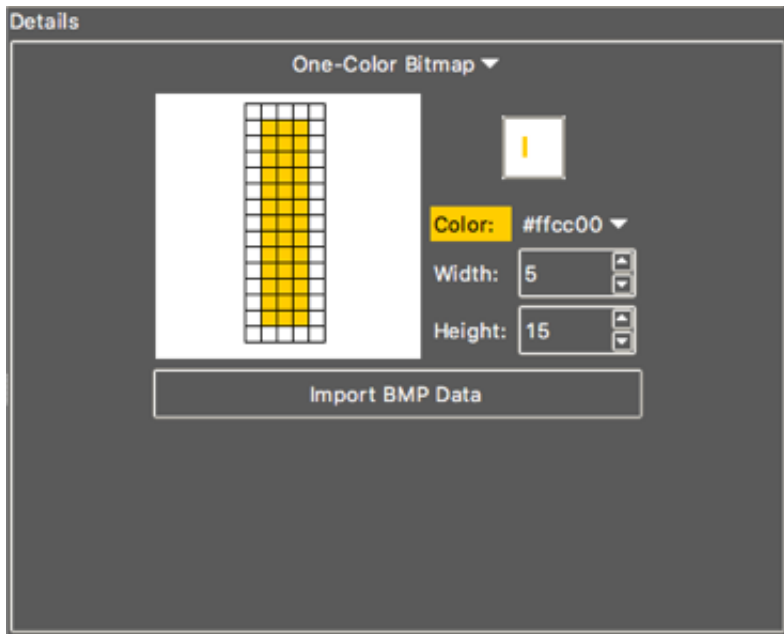
Image

The image pane is a powerful way to create and edit one or two-color bitmaps, or select a GIF image to use for a given UI image. The top-most selection menu allows you to switch between these three image editors.

Single-Color Bitmap Editor

The one-color bitmap editor creates a simple bitmap image that can contain one color. Any uncolored pixels will be transparent, taking on the background color of the window behind the image. To help represent what the image will look like, the background color of the editor and image preview area match the actual background color of the window. The following is a representation of the single-color bitmap editor.

On the left side of the pane is the bitmap editor. The grid of squares represents the pixels in the bitmap image. To set a pixel, simply left-click on the given pixel. To make the pixel transparent, simply left-click on a given pixel that is set.



On the right top is a preview of the image shown in actual size. Below the preview is the color chooser which allows you to change the color used in the image. You can select a color via a standard color chooser or you can select a color swatch value. Below the color picker are selection tools for setting the pixel width and height of the image.

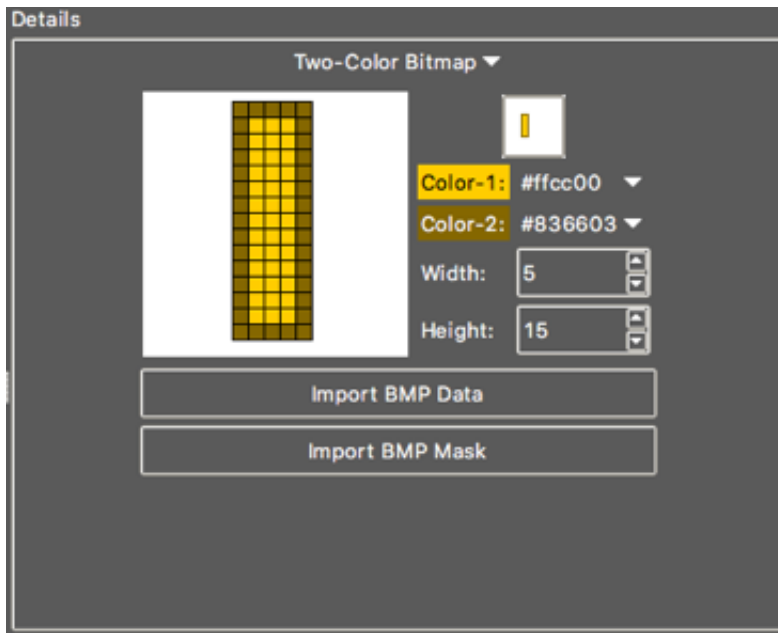
If you already have a BMP file containing the image, you can click the “Import BMP Data” button, select the file in the file chooser, and the bitmap image will be displayed in the selected color.

Two-Color Bitmap Editor

The two-color bitmap editor is much like the single-color bitmap editor except that the bitmap image is made up of two colors with any uncolored pixels being transparent. The following is a representation of this window.

On the left side, is the image editor. To set a pixel to a given color (or make it transparent), left or right-click on a pixel to change the pixel’s display.

The image preview is shown in the upper right-hand corner in actual size. Below this preview are two color choosers to select the first and second colors of the image. With the color choosers, you can select a color via a standard color picker or use one of the swatch colors. Beneath the color selectors, are widgets to change the height and width of the image.



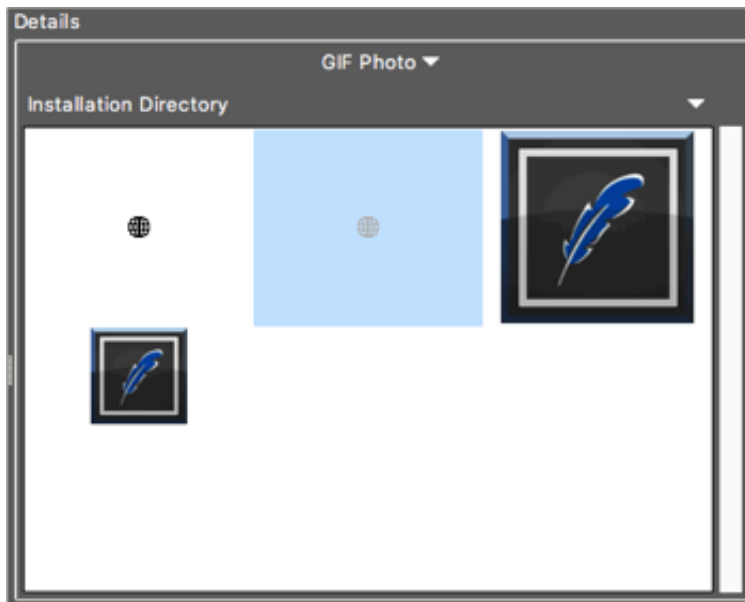
If you already have a BMP image in a file, you can click on the “Import BMP Data” or “Import BMP Mask” buttons will set the image’s bitmap data or bitmap mask fields to the given value. The mask defines which pixels are drawn and which are transparent. The data defines which pixels are drawn in the first (pixel set) and second (pixel clear) colors.

GIF Photo

The GIF photo browser allows the user to choose an existing GIF photo image from either the installed image directory, the user’s theme directory or an existing file on the file system. In the latter case, the image will be copied to the user’s theme directory when the theme is saved so that the image can be used even if the original copy is moved or discarded. The following image is a representation of this window.

At the top of the pane is a directory chooser for selecting an image. Three options are provided: installation directory (the image directory in the installed version of TKE), the user directory (user-owned directory in ~/.tke/themes) that contains the current theme data and related images, or choose a directory from anywhere in the filesystem. When an option is selected, any files with a .gif extension will be displayed in the grid view below.

The grid view shows previews of the available images. The background color of the grid will match the background color that the image will be displayed onto. To select an image to use, simply select it with the mouse. This will immediately update the image in the category option table.



Option Description Pane

This pane is located just below the option detail pane. Its contents simply help describe what the currently selected option in the category table controls.

Button Action Bar

At the bottom of the theme editor window is the button action bar. This area displays the UI buttons and other widgets used for opening themes for edit, saving theme changes, creating new themes, importing themes into TKE, exporting themes from TKE (for the purposes of sharing themes with other users), and previewing theme changes within the current TKE session.

The displayed content in this area is contextually aware. By default, the main open/preview/save button bar is displayed.

To open another theme for editing or import a theme into TKE, click on the “Open” button. This will change the button bar to the open/import button bar.

To import a .tkethemz theme file (TKE theme file package) or a .tmtheme (TextMate theme file), click on the “Import” button. This will display a file picker window. Find the desired theme file and click on the “Open” button. This will install the theme into the user’s theme directory (located in ~/.tke/themes) and immediately display the theme in the theme editor. Doing this will also redisplay the open/preview/save button bar.

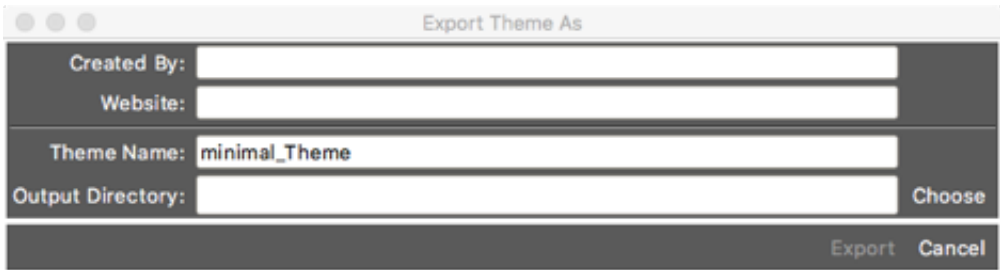
To open a different theme for editing, click on the “Choose Theme” button. This will display a list of installed themes. Selecting a theme from this list will display the theme contents in the theme editor and automatically preview them in the current TKE session. The import/open button bar will remain visible, allowing you to quickly open theme after theme without extra clicks. When you are done using the import/open button bar, you can either click the “Done” button or simply make a change to any part of the opened theme. Both actions will dismiss the import/open button bar and display the open/preview/save button bar.

When you want to see a live preview of the current theme changes in the current TKE session, click on the “Preview” button in the open/preview/save button bar. After a second or two, the current TKE session will be updated with the theme edits made in the theme editor (whether the changes have been saved or not).

If you want to save the current theme changes under a new theme name, left-click on the “Save” button. This will display the export/save button bar. You can change the name of the theme in the name entry field. If you are a TKE developer, a directory selection button will allow you to specify whether the theme should be saved in the TKE installation directory or the user’s theme directory. If you are not a developer, all new themes will be saved in the user’s theme directory. Clicking on the “Save” button will save the changes to the new theme and redisplay the open/preview/save button bar. Clicking on the “Cancel” button in the export/save button bar will cancel the theme creation and redisplay the open/preview/save button bar.

If you want to save the changes back to the current theme, either left-click on the “Save” button in the open/preview/save button bar and then click the “Save” button in the export/save button bar or, even more simply, right-click on the “Save” button in the open/preview/save button bar which will immediately save the changes to the current theme file without changing the button bar state.

To export the current theme to a theme package which can be shared with and imported by other TKE users, click the “Save” button in the open/preview/save button bar and then click the “Export” button. This will display a popup window which is represented in the following image.



Export Theme As	
Created By:	<input type="text"/>
Website:	<input type="text"/>
Theme Name:	<input type="text" value="minimal_theme"/>
Output Directory:	<input type="text"/> Choose
Export Cancel	

The first two entries allow the user to enter attribution information. This information will be displayed in the title bar of the theme editor when a user imports the theme and views it within the theme editor. Both fields are optional, but it is encouraged that at least the “Created By” field is filled in.

The last two fields contain the name of the theme being exported. By default, this field will be filled in with the current theme name being edited in the theme editor; however, you can change this name to be anything you want though short, memorable, unique names are preferable. The theme name is not optional. If no theme name is specified, the “Export” button at the bottom of the window will be disabled. The “Output Directory” field specified the name of the directory that will contain the exported theme package. The field is not editable; however, you can select a directory by clicking on the “Choose” button to the right of the display field. This field is also mandatory. Once the contents of this window have been properly filled out such that the “Export” button is clickable, clicking this button will dismiss the window and create the theme package file in the specified directory. The theme package file is a zipped file that can be shared in any method necessary with other users. You can install a TKE theme package by using the import functionality of the theme editor as described above, or, on Mac OS X, by dragging and dropping the file on the TKE application icon.

Plugins

In addition to all of the built-in functionality that comes standard, TKE also provides a plugin API which can allow development of new functionality and tools without needing to modify the source code. TKE ships with a small set of these plugins which are located in the TKE installation directory under the “plugins” directory.

Out of the box, plugins are not installed and available from within the tool; however, any plugin can be installed, uninstalled or reloaded within TKE (no restart is required). This will save those plugin settings to the user’s “plugin.dat” file in their ~/.tke directory. When TKE is exited and restarted, any previously installed plugins will be installed on application start.

Plugins can interface to TKE in a variety of ways. Which interfaces are used is entirely up to the developer of the plugin. Each plugin can create multiple interfaces into the tool to accomplish its purposes. The following table lists the various ways that plugins can interface into TKE.

Interface Type	Description
menu	Plugins can create an entire subdirectory structure under the “Plugins” menu.
tab	popups Create menu items within a tab’s popup menu.
sidebar	popups Create menu items within any of the sidebar’s popup menus.
application events	Plugins can be run at certain application events (i.e., on start, opening a file, closing a file, saving a file, receiving editor focus, on exit, etc.)
syntax highlighting descriptions	Provide a file containing a syntax highlighting description which is added to the built-in list of syntax highlighting schemes.

Installing a Plugin

Installing a new plugin is accomplished from the “Plugins / Install...” menu command. Doing so will display the command launcher in plugin installation mode. Any uninstalled plugins will be displayed in the launcher. To install a plugin from the list, simply select a plugin name from the list or begin typing a portion of the plugin name in the entry field until it is selected, and hit the RETURN key.

This will cause the plugin to be immediately installed. No application restart is required.

Uninstalling a Plugin

Uninstalling a plugin is similar to the process of installing a plugin. Select the “Plugins / Uninstall...” menu command. This will display the command launcher in plugin uninstallation mode. Any installed plugins will be displayed in the launcher. To uninstall a plugin from the list, simply select a plugin name from the list or begin typing a portion of the plugin name in the entry field until it is selected, and hit the RETURN key.

This will cause the plugin to be immediately uninstalled. No application restart is required.

Showing Installed Plugin Usage Information

You can view the list of currently installed plugins using the “Plugins / Show Installed...” menu command. Doing so will display the command launcher with the list of currently installed plugins. To the right of the list the currently selected plugin’s information will be displayed in the preview window. If a plugin is selected in the list and the plugin contains usage information, it will be displayed in a read-only editing buffer.

Reloading Plugins

Reloading plugins is only necessary in two cases. In the first case, the user installs a new plugin in the TKE installation’s “plugins” directory. In this case, the plugins will need to be reloaded so that the new plugin name will be viewable in the plugin install list without requiring an application restart. The second case where plugin reloading is needed is in plugin development (more on this process is described in the “Plugin Development” chapter within this document).

To reload plugins, simply select the “Plugins / Reload” menu command. This will cause all plugins to be immediately reloaded. No application restart is necessary.