Erbele

by Andreas Bentele

https://github.com/abentele/Erbele

Based on the discontinued project **Fraise 3.7.3** (https://github.com/jfmoy/Fraise).

Fraise originally was forked from **Smultron 3.5.1** (https://sourceforge.net/projects/smultron/), by Peter Borg.

Introduction

Erbele is a free text editor for Mac macOS Sierra 10.12 and later, which is both easy to use and powerful. It is designed to neither confuse newcomers nor disappoint advanced users. It should work perfectly for a whole variety of needs; like web programming, script editing, making a to do list and so on.

Erbele has all open documents in a list with beautiful Quick Look icons to your left just like e.g. iTunes so you can easily switch between many documents - you can also choose to display them as tabs if you prefer it that way.

It is easy to program with Erbele as it colours the content in different colours depending on what the code does. And you also have many ways to search for words and line numbers to help finding the code you are looking for. You can also split the window in two to display two parts of the same document or to compare two different documents side by side.

You can also preview HTML files directly in Erbele and save snippets of text and insert them simply with a shortcut. And if you don't want to be disturbed by other applications or the desktop you can let Erbele cover the whole screen to let you concentrate on your work.

For the more advanced users Erbele can find all those system files that are normally hidden and it has authenticated open and saves for them. Erbele can also use regular expressions and it can run commands and scripts.

For all international users there is full support for all encodings and it is translated into many languages so you can use it in your own native tongue.

Split window

You can choose to work with two different documents or two parts of the same document by splitting the window. By default it splits the window horizontally but by holding Option you split it vertically. You can only split the window so that you see one document over the other. You choose the document that should appear in the lower/right part by choosing it in the lower part of the document.

View in separate window

If you want to view the document in a separate window you can do so by double-clicking it in the documents list or choose the command in the **View** menu.

Quick look icons list

In the document list on the left hand side Erbele uses the Quick Look icons so it possible to get a glance of the contents directly.

View in full screen

You can edit the document in full screen (nothing else is visible, not even the menu bar) by choosing the command in the View menu. You return to the normal view by pressing the *Escape* button on the keyboard.

Functions

You can see a list of all the functions in your document in the **Tools** menu or with the toolbar item.

Tabs

You can see all documents in the documents as tabs by choosing **Show Tab Bar** in the **View** menu.

Encodings

Only a small selection of the available encodings are shown to begin with. You can choose to see more under **Advanced** section in Preferences. If you think that the characters are wrong you can test to reload the text with another encoding in the Text menu (this option is only available if there are no edits made to the document).

Snippets

Snippets are pieces of text that you use often or want easy access to. You create a snippet by using the command in the **Tools** menu or by dragging it into the Snippets

window. You can sort the snippets by dragging them into collections. You can also assign a shortcut key to the snippets so that you can insert the snippets by just using the keyboard - you do this by just placing the cursor in the *Shortcut* column and just press the shortcut you want. All snippets are listed in the **Tools** menu at the bottom. If you include one or several *%%s* in the text, they will be replaced by the current selection, if some text is selected when the snippet is inserted. And if you include a *%%c* the cursor will be placed at that position after the snippet has been inserted.

Preview

With this you can preview the HTML or Markdown in the current document. If e.g. you are using a FTP- program to edit an external file you set the base URL of that file so that all relative links are loaded, e.g. for images and css-files. If you are editing the file *index.html* for the site www.apple.com you can put http://www.apple.com/ as the base URL for all the images to load correctly (as long as you are connected to the internet of course). You can choose the preview parser from the drop down menu present in the Preview window, (each document you preview will retain its parser preferences).

Commands

You can run commands from within Erbele if you e.g. want to compile some code or make a file executable, or if you want e.g. to validate an HTML document or insert e.g. the current date. You can also assign a shortcut to the command which can be ran by just using the keyboard - you do this by just placing the cursor in the Shortcut column and just press the shortcut you want. If there is a result from the command it can be shown in the current text if you have chosen it to be inline or otherwise in a separate window. You can include %%p or %%d in your command to get the full path or directory of your current document. This means that you don't have write the path of the document but you can e.g. compile the current document easily. If you include *%%s* the selected text will be saved into a temporary file and *%%s* exchanged with the path to that file. You can also select some text and run that text as a command inline. E.g. if you write "*date*" (without the quotes) and select it and choose **Run Selection Inline** from the **Tools** menu it will be replaced by the current date.

Run text

When you choose **Run Text** in the **Tools** menu the text is saved to a temporary file and then ran with the command chosen in Preferences, normally */bin/sh*. Any result is displayed in the *Commands* window.

AppleScript

Limited AppleScript support is included in Erbele. You can retrieve some information and change the text in the current document. What is available can be seen in this example script:

```
tell application "Erbele"
    set theName to name as Unicode text
    set thePath to path as Unicode text
    if edited is false then
        display dialog "No need to save"
    end if
    set theContent to content as Unicode text
    set content to "This text will replace all text in the current document"
end tell
```

Live find

If you quickly want to find something you can use the *Live Find* toolbar item. Press *Command-Option-F* to place the cursor in the search field and write what you want to search for. If it can be found it will be selected and if you want to search for the next occurrence press *Tab* to get to the text and then press *Command-G*.

Erbele as an external editor

Erbele has support for the ODB Editor Suite so any program that implements that protocol (e.g. FTP programs) can use Erbele to edit its files.

Auto-completion

You can set in the Preferences whether Erbele should suggest the completed word automatically. Otherwise you can press *F5* or *Option-Esc* to get the list. If you don't find a match in what's been suggested you can press *Esc* or just continue writing.

Comment or uncomment selection

Inserts whatever the syntax definition has defined as single-line comment for the current document, e.g. // for a PHP-document, at the beginning of the line if the first line of the selection isn't already commented, otherwise it removes the comments. The menu item is only available if the used syntax definition has a defined single-line comment.

Command-line utility

Install it through the **Help** menu (note that if you choose to install it directly you have to authenticate with an administrators username and password twice; once for the utility and once for its man page). With Erbele (the name of the utility) you can start the application Erbele by just writing "*erbele*" in the Terminal with no options. If you write something after *erbele* it considers that to be the path to the file or, if you write more than one path, the paths to the files. You can use the standard wildcards e.g. * and ~ for your home folder. If you don't specify a full path it adds the current working directory to what you write. If it can't find any of the specified files, it first creates them and then opens them. (Note that if you use a wildcard and it can't match a file to that it creates a file including the wildcard character as a character in the path, e.g. if you write `erbele ~/`

`Desktop/*.jsp` and there aren't any .jsp-files on the desktop it creates a file called *.jsp on the desktop.) For more information write "*man erbele*" in the Terminal.

Services

If you choose **Services** from "application" menu you can make Erbele open the currently selected file, insert the selection into the current document or create a new document with the current selection. (Note that you need to log out and then log in at least once before this feature works.)

Advanced find

Multi-document search and replace with the option to use regular expressions. The regular expression engine is ICU. You can find more information about it here: http://www.icu-project.org/userguide/regexp.html

Show invisible characters

Shows a symbol for characters that are usually not seen, i.e. space, tab and new-line.

Drag-and-drop of files into the documents list

You can drag-and-drop one file or several files (or a folder) into the document list and Erbele opens them. You can also drag documents between projects, but if you drag from a tab bar the destination must also be a tab bar and likewise if you drag from the document list you must drag to a document list.

Grab text to scroll

If you press the *option* and *command* keys and drag with the mouse you scroll the text.

Print settings

When you choose **Print...** you can choose **Erbele** in the pop-up menu and choose which font to use. You can also choose if you want to print the header and if you only want to print the current selection. It is also possible to set the smallest margin you want to use and if you want to print the syntax colours.

Open hidden...

With this you can see all the files even the ones that the system usually hides.

Save a copy as...

Save a (backup) copy at a location without affecting your current file.

Remove needless whitespace

Removes any possible spaces or tabs after the text and before the line break.

Entab & detab

Replace tabs in the selected text with spaces or vice-versa.

Shift left & right

Remove tabs and spaces to shift to selected text to the left or insert tabs to move it to the right.

Navigate between different documents

Press *command* and *option* and *up-arrow* (or *left-arrow*) to select the document above in the list, and *command* and *option* and *down-arrow* (or *right-arrow*) to select the one below. You can also select the first ten documents in the list by pressing *command* and a number from 1 to 0.

Line numbers

If you don't want to show the line numbers choose **Hide Line Numbers** in the **View** menu. Or you can set the Preferences to not show line numbers for new documents.

Syntax colours

If you don't want to show the syntax colours choose **Hide Syntax Colours** in the **View** menu. Or you can set the Preferences to not show syntax colours for new documents.

Line wrap

If you don't want the lines to wrap at the end of the window choose **Don't Line Wrap** in the **View** menu. Or you can set the Preferences to not line wrap new documents.

Status bar

If you don't want to show the Status Bar choose **Hide Status Bar** in the **View** menu or if you just want to not show some particular information set the Preferences to your wish.

Close tag

This tries to find the correct tag to close by calculating the number of tags that has been closed and opened (skipping tags like *
* and *<hr />*) so it will find the last tag that was opened but not closed. After it has found the correct one it insets the closing tag adding a / to the opening tag e.g. *</head>*. Note that this feature only works with tags surrounded by *<* and *>*.

Prepare for xml

This changes all occurrences of **&**, **<** and **>** in the selection into **&**, **<** and **>** so that the text can be used in a XML-file.

Dark Mode

Erbele supports macOS Dark Mode. In *Preferences -> Appearance* preference pane you can set different colour patterns for Light and Dark modes.

Preferences

create new document

Whether you want a new empty document or nothing at all when you start the program. This option has no effect if you start with a file e.g. by double clicking on a file.

syntax colour

Whether new documents should by default get syntax colours according to the settings in the *Syntax Colouring* tab.

line wrap

Whether new documents should wrap the lines by default so you can see the whole line even if it is long.

show invisible characters

Whether to show invisible characters as default for new documents.

spell check as you type

Turns on spell checking automatically.

show line gutter for new documents

Whether to show the line numbers field for new documents. The width is in pixels and the line numbers field automatically resizes if it is too small. Set the width to no less than 20 and there really should be no need to have it larger than 50 but you can if you want to.

show matching braces

When you write a), } or] Erbele briefly highlights the corresponding (, { or [if it can find it, otherwise it beeps.

indent new lines the same as the line above

After you press *Enter* places the cursor on the new line where text starts on the preceding line.

treat { and } intelligently

Indents one step on the next line if the last character on the preceding line is a { and shifts the characters on the line to the left when inserting a } if there are no letters before.

suggest automatically, after delay

Whether the list of matching word-completions should appear automatically and how long (in seconds) after the last pressed key that the list of matching words should appear.

include standard words

Whether the common English words should also be included in the auto-completion instead of just the words from current syntax definition.

number of items in recent menu

How many past files to display in the **File** menu under **Recent Items**.

Appearance

Text font

Set the font and size for the text of the document and also for its line numbers.

Text colour

Set the colour of the text of the document.

Background colour

Set the colour of the background of the document but not for its line numbers.

Gutter text colour

Set the colour of the text of the line numbers field.

Gutter background colour

Set the colour of the background of the line numbers field.

Tab width

This sets how wide a tab should be; it calculates the width by checking the width of the number of spaces you state here in the current font.

Indent width

This sets how wide an indent should be; it calculates the width by checking the width of the number of spaces you state here in the current font.

Show full path of document

If you want the full path of the document to appear in the documents list.

Size of text

Set the size of the system font to either 11 or 13 where the open documents are listed.

Show full path in window title

If you want to display the full path to the current document e.g. "*CocoaIsWonderful.html - /Users/steve/Desktop*" instead of just "*CocoaIsWonderful.html*" in the title of the window.

Show page guide, at column

If you want to display a faint line after e.g. 80 characters. This only works correctly when you use a monospaced font.

Status bar: last saved, in format

The date and time of the last time the document was saved and the format it should be displayed in. See *Expert Preferences* for instructions on how to change the format yourself.

Status bar: length of document

The number of characters in the document

Status bar: length of selection

The number of characters in the current selection, if less than two characters are selected this won't show in the status bar

Status bar: position

Shows first the location of the cursor in the line and then the location of the cursor in the whole document.

Status bar: text encoding

Show which encoding the current document has; you can change it through **Text Encoding** in the **Text** menu and also set the Preference as you like in the **Open & Save** tab.

Status bar: syntax definition

Show which syntax definition is set for this document; you can change it through **Syntax Definition** in the **View** menu and also set the Preference as you like in the **Syntax Colouring** tab.

Open

Whether when you choose **Open** to start at the folder where you last opened a document from, or start at the same folder as the current document or whether to always start at a particular folder.

Save as

Same as for **Open** but for when you choose **Save As...**

Open all documents I had open when I last quit

This saves a list of all the documents (except for new documents that has never been saved) that are open (and not in a saved project) when Erbele is closed and opens them automatically the next time Erbele starts. If there are documents in more than one unsaved project all those are collected into one project the next time Erbele starts.

Assign document to Erbele

When a Mac OS X application saves a file it sometimes sets the Creator and Type of the document so that whenever you double-click on that document it starts with that particular application. If you choose this option Erbele will change that information so that the document will always be opened by Erbele. New documents are always saved so that they are opened by Erbele.

Append name in "save as", with

If you e.g. always want to add the extension .txt to all filenames, check this box and Erbele will automatically append that to the name when the "Save As" sheet comes up.

Line endings (when saving)

Different operating systems uses different invisible characters to denote that you want a new line. This option gets you to choose to which line ending you want when you save your document. If you don't have a particular need you can usually set it to "Leave unchanged" which leaves the line endings as it is for documents that you have opened. New documents uses the Unix standard.

Advanced - Syntax Colours

Active - colour

Here you can turn on syntax colouring for each part and decide which colour Erbele should use. You can use a different set of colours for Dark Mode. See *Syntax Colouring* for a description of what the different parts are.

Only colour till the end of the line if it can't find closing tag

Only continue to colour till the end of the line instead of the rest of the document if just the start tag is found and not the closing tag in e.g. a multiline comment

Colour multi-line strings

Colour a string even if it has line-breaks in it.

Advanced - Syntax Definitions

Use definition

Here you choose whether the Erbele should guess from the extension which syntax definition to use (e.g. if the extension of the document is .html it will use HTML) or if it should always use a particular syntax definition). If the document does not have an extension it tries to guess by checking the first line of the document for e.g. #!/usr/bin/perl and then maps it to the syntax definition; if it can't guess, it will use the Standard one.

The table

The table lists all the extensions that are mapped to each language. If you add or change an extension just double-click in the Extensions-column and change the extension (the extensions are space separated and there should be no dots). Erbele uses the first language (alphabetically) that has the extension so if many languages share the same extension make sure that no language above also has the extension.

Advanced - Encodings

Choose whether to always use a particular text encoding when you open a file or whether to guess the encoding of the document. Erbele guesses by trying to see if any information about the encoding is saved along with the file otherwise it checks for "charset=" or if it can't find that, search for "encoding=" and map the encoding to what follows after that. If it can't find the encoding this way it checks for either EFBBBF, FEFF or FFFE at the beginning of the file and, if found, set the encoding to UTF-8 or UTF-16. If it can't guess by any of these means it sets it to the chosen encoding. Erbele checks to see if the document has been saved with a particular encoding (it is saved as a piece of meta-data) and, if so, it always uses that one.

indent with spaces, not tabs

Writes spaces and not a tab when the tab key is pressed. How many spaces are determined by *Tab width* in *Appearance*.

tab stops

Whether you want to use tab stops or not, i.e. if it should calculate how many spaces are needed to get to the next stop, which is defined by "*Tab Width*", instead it will insert a fixed number of spaces when pressing tab (if one has chosen to insert spaces instead of a tab when indenting).

enable smart insert and delete

With this option Erbele can keep track of spaces when you cut, copy and space so that you never get e.g. two spaces before a word.

check if document has been updated by another application

If another application opens the current document and saves it so that the document you opened is not the same as it is now you get the choice with this option if you want to update your document or whether to ignore the changes made to the saved document.

use rgb rather than hex when inserting colour values

When choosing a colour with **Insert Colour...** in the **Action** menu you can insert the colour value like this `rgb(255,255,255)` rather than like this `#FFFFFF`.

auto-insert a closing)

Automatically inserts a) when you write a (.

auto-insert a closing }

Automatically inserts a } when you write a {.

highlight current line

Changes the background colour of the current line so it is more easily seen.

colour for invisible characters when shown

The colour that the usually invisible characters have when they are shown.

opening: if I choose to open a folder, open all files within it

Opens all files within the chosen folder.

opening: recursively

Whether to open files in subfolders as well.

opening: filter out documents with these extensions

If you e.g. open a folder with some images you can filter out those extensions here so those images will not be opened.

preview parser

When the preview is shown this setting decides what default parser is used to interpret the code into the seen preview.

run text

When you choose **Run Text** in the **Action** menu the text is saved to a temporary file and then ran with this command.

Expert Preferences

Quit Erbele and run these commands in Terminal (which you can find in */Applications/ Utilities*) and then start Erbele again.

date format

You can change the last date format in the list under Appearance, you do it by writing:

```
defaults write org.erbele.Erbele UserDateFormat "mmmm"
```

The formats that you can insert instead of mmmm can be found here: http:// icu.sourceforge.net/ userguide/formatDateTime.html

update documents automatically without warning

By writing:

```
defaults write org.erbele.Erbele UpdateDocumentAutomaticallyWithoutWarning -bool YES
```

documents are reloaded automatically without a confirmation dialogue when setting Erbele to check if another application has updated the document. Exchange YES with NO if you want to turn off this feature again.

time between document update checks

By writing:

```
defaults write org.erbele.Erbele TimeBetweenDocumentUpdateChecks 10
```

you check if another application has updated any of the open documents every 10 seconds instead of the default 15 seconds. You can give it a value down to 1 second.

end file with line feed

By writing:

```
defaults write org.erbele.Erbele AlwaysEndFileWithLineFeed -bool YES
```

Erbele will check to see if the document ends with a line feed character (\n) and if not, it will add one. Exchange YES with NO if you want to return to the default state.

suppress replace warning

By writing:

```
defaults write org.erbele.Erbele SuppressReplaceWarning -bool YES
```

Erbele will not ask if you really want to change all occurrences in Advanced Find but just change them directly. Exchange YES with NO if you want to return to the default state.

delay before preview should update live automatically

By writing:

```
defaults write org.erbele.Erbele LiveUpdatePreviewDelay 5
```

you will change how long the delay will be after the last key is pressed before the preview should automatically update if you have checked the "live Update" box. Default value is 1.

quotes around directory when using %%d

By writing:

```
defaults write org.erbele.Erbele PutQuotesAroundDirectory -bool NO
```

Erbele won't put quotes around the path to the directory referenced by %%d. Exchange NO with YES if you want to return to the default state.

where focus should be when selecting results in Advanced Find

By writing:

```
defaults write org.erbele.Erbele FocusOnTextInAdvancedFind -bool YES
```

Erbele will change the focus when you click on a result in Advanced Find to the text so that you can directly change it. Exchange YES with NO if you want to return to the default state so that Erbele will keep the focus in the result list so that you can e.g. swtich between results with arrow-up and down keys.

keep empty window open

By writing:

```
defaults write org.erbele.Erbele KeepEmptyWindowOpen -bool YES
```

Erbele will keep the project window open after the last document within has been closed. Exchange YES with NO if you want to return to the default state.

grammar check

By writing:

```
defaults write org.erbele.Erbele AutoGrammarCheck -bool YES
```

Erbele will automatically check the grammar. Exchange YES with NO if you want to return to the default state.

automatic quote substitution

By writing:

```
defaults write org.erbele.Erbele AutomaticQuoteSubstitution -bool YES
```

Erbele will automatically substitute quotes from e.g. an inch character to a "proper" quote. Exchange YES with NO if you want to return to the default state.

automatically detect links

By writing:

```
defaults write org.erbele.Erbele AutomaticLinkDetection -bool NO
```

Erbele will not automatically detect links to e.g. web sites in the text and make them clickable. Exchange NO with YES if you want to return to the default state.

quicklook icons

By writing:

```
defaults write org.erbele.Erbele UseQuickLookIcon -bool NO
```

Erbele will use the standard icon and not the Quick Look icon in the document list. Exchange NO with YES if you want to return to the default state.

updating icons

By writing:

```
defaults write org.erbele.Erbele UpdateIconForEverySave -bool NO
```

Erbele will not update the icon every time you save and you will save resources but won't be able to see updates to the icon. Exchange NO with YES if you want to return to the default state.

Syntax Colouring

Erbele syntax colours your document if you have chosen so with the syntax definition of your choice.

the colours

The easiest way to learn which colours to choose in Preferences is to test on a document which you know well. But here to the right is an image of a php-code example which might help you.

```
/*This is a comment*/
<head>
<title>Erbele</title>
<?php
echo("This is a string");
$variable = "Erbele";
```

"<?php" and everything in grey until the closing tag (not shown here) are Instructions. "<head><title>" are Commands. "echo" is a Keyword.

change a syntax definition

The syntax definitions are defined in plist files in the Application bundle (ctrl-click on Erbele and choose *Show Package Contents -> Contents -> Resources*) e.g. html.plist for HTML documents. If you want to change anything or add a keyword you can just change the file and save it. (Remember to treat special characters in the standard xml-way e.g. < instead of < and if you don't want leave a string empty please do it like this <string></string>.)

The keys are: (I use php as an example within parentheses):

`beginCommand`
(<) to show html commands

`endCommand`
(>)

`beginInstruction`
(<?) to start a block of code

`endInstruction`
(?>)

`beginVariable`
($) the character (or any of the characters) that signals that it is a variable

`endVariable`
(:;(){}=[]+-/.,*!?%&><\) anyone of these characters ends the variable name

firstString
(") only one character is recognised for the strings

secondString
(')

firstSingleLineComment
(//) a comment till the end of the line

secondSingleLineComment
(#) another comment till the end of the line

beginFirstMultiLineComment
(/*) the start of a multiline comment

endFirstMultiLineComment
(*/)

beginSecondMultiLineComment
(<!--) to allow e.g. html comments in php-code

endSecondMultiLineComment
(-->)

keywordsCaseSensitive
(<false/>) set either to </false> or </true> if you want the keywords to be case sensitive

recolourKeywordIfAlreadyColoured
(<false/>) if you want a keyword to be coloured as a keyword even if its already been coloured as a command

keywords
(<string>and</string>) a list of keywords

autocompleteWords
(<string>auto-complete this word</string>) a list of words to add to the auto-completion

functionDefinition
(<string>a regular expression</string>) a regular expression that finds the beginnings of functions in the document

removeFromFunction
(<string>*a string*</string>) a string that is removed from the function name, e.g. if the function is defined by having the word function in front of it, it can be removed from the list to make it easier to read

```
attributes
```
you cannot define this yourself but it defined as the word before a "=" in a command

optional values

You can also add any of four optional strings which adds or removes characters from the character sets that decide where a keyword can start or end. The four keys are:

```
excludeFromKeywordStartCharacterSet
excludeFromKeywordEndCharacterSet
includeInKeywordStartCharacterSet
includeInKeywordEndCharacterSet
```
just add all characters you want, to the string to these keys.

add new definition

If you want to add a new definition, you must create a new file named languagename.plist and fill it with information as per above (it is easiest to copy an a plist file and only change what you need). Then you add a (or open the existing) SyntaxDefinitions.plist file in the /*Users/<your home folder>/Library/Application Support/Erbele* folder and add a dictionary into the file. The best way to do this is to make a copy of the SyntaxDefinitions.plist that's in the application bundle and start from that one. An example is below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//
EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<array>
<dict>
    <key>name</key>
    <string>PHP</string>
    <key>file</key>
    <string>php</string>
    <key>extensions</key>
    <string>php php3 phtml phtm</string>
</dict>
</array>
</plist>
```

The name-key is what shows up in the View menu and the file-key is what you name your file minus the .plist extension. The extensions-key lists the various extensions your language can have, space separated and without dots.

what is Erbele?

Erbele is a text editor which is both easy to use and powerful and it is written in Cocoa for Mac OS X.

requirements?

macOS Sierra 10.12 or later

the name?

Why "Erbele"? That's simple: Fraise was forked from Smultron in 2010. "Smultron" is the Swedish word for "wild strawberry", and "Fraise" is the French word for "Strawberry". As I'm a German software developer, why not using the same word in German? But in German it would be "Walderdbeere", which is a little bulky. This is the reason why I used the Swabian (Swabian is a German dialect) word for it, which is "Erbele". I think most people in Germany today don't know this term (even the Swabian people), and there's no other brand using this term.

why you've written it?

I've always thought that there has been a need for an easy, free text editor which won't confuse newcomers nor disappoint advanced users.

does it work with my ftp program?

Yes. You can usually decide which program should be an external editor in FTP programs and if you choose Erbele and then choose to open the file in the external editor it opens in Erbele, and then when you save the file in Erbele it is automatically uploaded to the FTP site.

how to delete snippets, commands etc.?

Select it and press *Backspace* or *Delete* on the keyboard.

how to see the current function?

Either in the *Info* window or, if you choose the **Functions** menu (in **Tools**) or the toolbar icon the current function is marked.

how to see all the spotlight information in the Info window?

Place the cursor in the information and just go down or right with the cursor keys and the text field will scroll to show all information.

where are things saved?

Snippets, Commands and Projects are saved in *<your home folder>/Library/Application Support/Erbele/Erbele3.erbele*. Preferences and open files are saved in *<your home folder>/ Library/Preferences/org.erbele.Erbele.plist*.

how to uninstall it?

Drag Erbele.app to the trash, and you can find the preference file at *<your home folder>/ Library/Preferences/org.erbele.Erbele.plist*. If you've installed the command-line utility you can find "erbele" in */usr/bin* and its man file "erbele.1" in */usr/share/man/man1*. You also need to remove the folder *<your home folder>/Library/Application Support/Erbele*.

how to translate Erbele into my own language

You need first of all to install the Developer Tools. Then download the source code from the site. Choose English.lproj from the folder and copy it onto the desktop. Rename it to your language (the easiest way to know exactly what name it should have is to look inside another application that you know already has support for your language).

Then open all *.strings* files in a text editor (and make sure that it opens with text encoding Unicode UTF-16) and translate the right side of every string (you'll see things like %2$@ and you need to keep them in the string as that is the code to insert e.g. a filename).

You don't need to translate *InfoPlist.strings* or Erbele Help.

You can send the finished translation to me (the email address can be found in **About Erbele**) and I'll include it with the next release.

where is the code?

https://github.com/abentele/Erbele

where is the latest version of Erbele?

https://github.com/abentele/Erbele

Troubleshooting

First make sure that you are running the latest version. Go to https://github.com/abentele/ Erbele and check to see if you have the latest version. Then you can check to see if you have any Input Managers installed. Check the following directories:

- */Library/Application Support/SIMBL/Plugins/*
- */Library/InputManagers*
- *<your home folder>/Library/InputManagers*

and try to disable any there and restart Erbele and see if the problem persists.

If that doesn't help you can try to move aside (e.g. move temporarily to the desktop) the Erbele preference file so every setting will be reset to the default:

- *<your home folder>/Library/Preferences/org.erbele.Erbele.plist*

and the data file:

- *<your home folder>/Library/Application Support/Erbele/Erbele3.erbele.*

If nothing works you can try to download a fresh version of Erbele from https:// github.com/abentele/Erbele to make sure that the application hasn't been damaged in any way.

You can also check to see if it's a known bug at https://github.com/abentele/Erbele/ issues and if I'm working on a fix. And, if not, you can email me (the address is in **About Erbele**) or you can file a new bug at that page.

History

Current versions

Changelog entries for current versions, starting with version 3.7.4, can be found here:
https://github.com/abentele/Erbele/releases

Earlier versions

3.7.3 2010-03-29
Lastest Fraise release by before project discontinuation.

3.7.2 Fraise
Name changed for **Fraise** (to avoid confusion with Smultron, following a request of Peter Borg)
New update system using Sparkle framework
French, German, Japanese, Spanish, Swedish translations updated. (Italian coming)
Bug Fix: Doesn't switch to Fraise's space when opening a document in a different one.

3.7.1
String search in all files contained in the parent directory of the current document.
Duplicate Line
Auto Update
Correction of the French translation of SMLMainMenu.xib
Correction of the pointing URL in the About window.

3.7.0
Smultron has been discontinued. This is the new fork by Jean-Francois Moy:
Full 64 bits support (Snow Leopard)
Import of all localizations

3.5.1 2009-01-31
Added Catalan and Korean localisations
Added syntax definitions for Cobol, Csound, Erlang, GEDCOM, JavaFX, MEL, NEURON and SuperCollider
Improved the preview when printing

3.5 2008-04-16
Fixed an issue with the Italian localisation
Fixed an issue with the complete-menu
Added ASM-MIPS syntax definition
New toolbar icons

3.4 2008-04-04
Added Greek localisation
Faster
Reduced memory usage
Added MySQL syntax definition
Added Metaslang syntax definition

Added VHDL syntax definition
Various other small improvements and bug fixes

3.3 *2007-11-27*
Updated appearance
Now uses Quick Look instead of standard icon in documents list
More animation
Some small improvements and bug fixes

3.2.1 *2007-11-15*
Added Ox syntax definition
Some bug fixes

3.2 *2007-10-28*
Only for Leopard
Added Chinese (Traditional) localisation
Added IDL syntax definition
Added Logtalk syntax definition
Added Udo syntax definition
Some other improvements and bug fixes

3.1.2 *2007-07-16*
Added Scala syntax definition
Some other small changes

3.1.1 *2007-06-21*
Added Chinese (Simplified) and Japanese localisation
Added Verilog syntax definition
Added "hidden" preference to keep project window open after last document is closed
Fixed possible issue when using multiple single document windows
Some other small improvements and bug fixes

3.1 *2007-05-19*
Changed default encoding to Unicode (UTF-8)
Added way to automatically open previously opened documents/projects
Added Dutch localisation
Added ASM-x86 syntax definition
Fixed an issue when changing font size with the keyboard
Fixed a possible issue with the Spotlight importer
Made a small improvement when working as an external editor
Some other small improvements and bug fixes

3.0.7 *2007-05-14*
Added convenient way to change font size (Command Plus and Minus)
Fixed an issue when opening a saved project
Fixed a possible issue with Advanced Find
Fixed an issue with Services
Fixed an issue with setting shortcuts for commands
Fixed an issue with the colouring of CSS
Improved the German localisation
Some other small improvements

3.0.6 *2007-05-10*
Fixed a couple of possible crashes
A couple of minor improvements

3.0.5 *2007-05-08*
Added Italian, French and German localisation
Added eZ Publish syntax definitions
Improved the collaboration when working as an external editor
Fixed a couple of possible issues with Advanced Find
Some other improvements
Some other bug fixes

3.0.4 *2007-04-30*
Fixed a couple of possible crashes
Fixed a possible problem when working as external editor
Improved the highlighting of the current line
A few other small improvements

3.0.3 *2007-04-28*
Added Hungarian and Spanish localisation
Fixed possible issue when working as external editor
Fixed possible problem with regular expressions and replace
Fixed an issue when saving projects
Fixed bug when checking for updates
Various other small fixes and improvements

3.0.2 *2007-04-23*
Added Czech and Finnish localisation
Fixed possible problem with showing the wrong syntax colours
Fixed problem with sometimes not showing Advanced in Preferences
Fixed possible crash when opening a document with authentication
Various other small improvements

3.0.1 *2007-04-21*
Fixed some possible crashes

3.0 *2007-04-21*
Handling of projects is much improved
Running of commands is much improved
Updated apperance
New icons
Reduced memory usage
Improved and faster syntax colouring
Improved printing speed
Added vertical split (hold option)
Added FreeFem++ syntax definition
Bug fixes galore
Many, many other improvements

2.2.7 *2007-03-21*
Added Serbian (Latin) localisation
Added AMPL and TorqueScript syntax definitions

2.2.6 *2006-12-27*
Added syntax definitions for Lisp, Lilypond, MetaPost and LSL
Fixed a possible problem with large Markdown files
Updated Finnish localisation
Reduced memory usage under certain circumstances
Some small improvements and bugfixes

2.2.5 *2006-11-19*
Added Chinese (Simplified) localisation

2.2.4 *2006-11-14*
Can now preview Markdown (see Hidden Preferences)
Some new icons
Some small improvements to Shift Left and Right
Some other small improvements and bugfixes

2.2.3 *2006-11-10*
Fixed possible problem with Shift Left
Added possibility to have separate indent and tab widths (see Hidden Preferences)
Changed shortcut of Shift Left and Right to standard (Command-[and])

2.2.1 *2006-11-08*
Added Korean localisation
Added Nemerle syntax definition
Now uses tab stops per default when using spaces instead of tabs for indenting
Can now make sure file ends with line feed (\n), see Hidden Preferences
Updated Czech and French localisations
Some other improvements and bugfixes

2.2 *2006-10-21*
Added option to have black and white toolbar icons (see FAQ)
Readded line wrap toolbar icon

2.1.5 *2006-10-16*
Added Czech localisation
"Close Tag" now recognises self-closing tags
Readded "Split Window" toolbar icon
Assorted improvements and bugfixes

2.1.4 *2006-10-06*
Added syntax definition for RHTML
Fixed a possible temporary hang at startup
Some improvements to the syntax colouring
A few other small improvements and bugfixes

2.1.3 *2006-09-23*
Fixed a possible crash
Assorted improvements and bugfixes

2.1.2 *2006-09-05*
Fixed possible crash at startup

2.1.1 *2006-09-04*
Fixed a memory leak

2.1 *2006-09-02*
Added Italian and Finnish localisation
Added syntax definition for Active4D
Added a way to unregister the sync schema (see FAQ)
New icons

2.0.4 *2006-08-06*
New toolbar icons
Added syntax definition for R

2.0.3 *2006-07-22*
Added German, Chinese (Traditional) and French localisation
Improved the handling of projects

2.0.2 *2006-07-11*
Now the window doesn't reappear after hiding under certain settings
Now Open Hidden... works
Improved the Live Find
Improved the count of documents in projects
Fixed a rare crash which could occur when importing and exporting snippets

2.0.1 *2006-07-09*
Snippets can now be inserted from contextual menu
Now indicates if a document is edited in the tab bar
Fixed a bug when manually changing syntax definition
Improved the automatic indentation
Some other small bugfixes and improvements

2.0 *2006-07-02*
Split view
View document in separate window Tabs
Functions list
Run commands
Info window
Improved projects handling
Full screen editing
.Mac synchronization of snippets and commands
Improved syntax colouring
Print syntax colours
Resolution-independent graphics
Page guide
Now works with multiple selections
Slightly more AppleScript support

1.2.7 *2006-03-09*
Some small fixes
Added Dylan syntax definition

1.2.6 2006-01-01
Updated the appearance
Prepared it for x86
Added Russian localisation
Added Spotlight importer
Added Apache syntax definition

1.2.5 2005-05-08
Fixed an issue when working as an external editor

1.2.4 2005-05-07
Fixed an issue with auto-complete

1.2.3 2005-05-04
Added Spanish localisation
Improved the syntax colouring
Fixed an issue where you sometimes would have to authenticate unnecessarily when saving
Fixed an issue where the unsaved-document sheet could be closed without choosing an option

1.2.2 2005-04-20
Fixed a few small bugs

1.2.1 2005-04-07
Fixed some issues with the syntax colouring

1.2 2005-04-05
Better syntax colouring
Faster syntax colouring
Much faster opening of large documents
Command line utility can now open several files and it understands standard wildcards
Added Czech localisation

1.1.2 2005-04-02
Fixed an issue with diacritical characters
Added Simplified Chinese localisation

1.1.1 2005-03-29
Fixed an issue when functioning as an external editor
Fixed an issue with the Haskell and Matlab syntax definitions

1.1 2005-03-28
Implemented the ODB Editor Suite so Smultron can be used as an external editor by e.g. FTP-programs
Added auto-completion
Added a Save All menu item
Added a menu item to either Comment or Uncomment the selection
Implemented a standard Open Hidden... dialogue
Added a Preview toolbar icon
Added word count in the status bar
Can now rearrange the order of the documents
Tab-key now shifts text right when text is selected instead of replacing it with a tab
Code Snippets are now added to the previous when imported and will not replace the previous

Added Haskell and Matlab syntax definitions
Made adding your own syntax definition simpler

1.0.2 *2004-11-10*
Added localisations (Italian and Chinese - Traditional)
Fixed a few small bugs
Added Lua syntax definition

1.0.1 *2004-08-07*
Added Japanese localisation Improved the syntax colouring
Fixed a couple of bugs
Added Stata syntax definition

1.0 *2004-08-04*
Added localisations (Swedish, French and German)
.Mac synchronisation of the preferences
Added a Open Recent Project-menu
Added option to automatically open all documents that were open the last time Smultron was closed
Improved the syntax colouring
One can now replace with an empty string in Advanced Find, i.e. delete
Added option to colour multi-line strings
Now sets the insertion point to the same colour as the text
Now guesses the syntax definition from the contents of the first line if there isn't an extension
Added syntax definitions (PDF, LaTex and Objective Caml)

0.8 *2004-07-17*
Added possibility to use authentication when saving and opening files
Added a command-line utility (install it from the Help-menu)
Added Services (Insert Selection into Current Document, New Document Containing Selection, Open Selected File)
Added command to save all open documents as a project to be able to open all of them in one step
New icons, by Gaëtan Ark
One can now remove "Quickly Find Next..." from the toolbar
Added option so Smultron opens all documents within a folder, instead of giving a warning
Added option to set colour of invisible characters (when shown)
Fixed an issue where under certain circumstances the current document would be closed instead of the Find-window
Fixed a bug which under certain circumstances could delete the wrong character

0.7 *2004-07-01*
Added multi-document find and replace with regular expressions
Added possibility to show invisible characters
Can now drag-and-drop files into the documents list
Can now choose a different font for printing, and can also choose to print header, only selection and to set margins.
Added option to show position in status bar
Can now find hidden files and open them
Added option to indent with spaces instead of tabs
Can now choose an encoding in Open... sheet
Can now scroll the text by grabbing it with the option key and mouse
Can now choose to Save A Copy As...

Now updates the window properly when it is resized with no open documents

A few small fixes with the syntax colouring

ActionScript syntax added

0.6.1 *2004-06-06*

Added settings for tab width.

Added an option to make Smultron immediately recolour the whole visible page after every pressed key. This makes it slightly slower but it removes the flickering that otherwise can occur in languages like PHP and JSP.

Added a Close Tag-command that will in a somewhat intelligent way close your tag. It will try to find the right tag to close and skip those that don't need closing like e.g.
.

If it is set to guess encoding and it can't find an explicitly stated encoding, it now checks for either EFBBBF, FEFF or FFFE at the start of the file and, if found, sets the encoding to UTF-8 or UTF-16.

Now updates the status of the document after a Revert.

Fixed two small bugs with the display of find-results in the drawer.

Fixed so that Smultron can open documents of all Types with drag-and-drop.

Now in Quickly Find Next... after a failed search, Smultron sets the focus back to the document so that it receives the next pressed key.

Adjusted the wording of the alert-sheet that is presented if another application has updated the document, and made sure that its checks are performed when they should occur.

Corrected the icons in Help.

Fixed so that all preferences are remembered properly.

0.6 Smultron *2004-06-03*

First official release