# Textcavator

| | |
|---|---|
| Name: | Textcavator |
| Author: | Mike Pasini |
| Version: | 2.0a |
| Last Update: | 6 August 2020 |
| Requires: | Keyboard Maestro v8, CocoaDialog 3 |

## Table of Contents


Intro

Installation

Using the Extended Prompt

Using and Configuring the Google Prompt

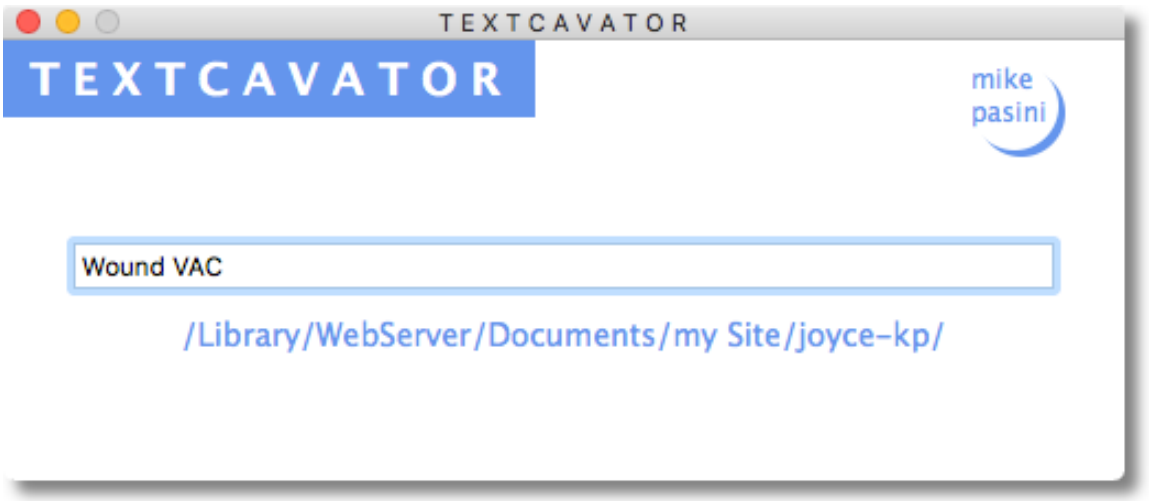
Customization

Running Textcavator

Notes

Release Notes

Contact

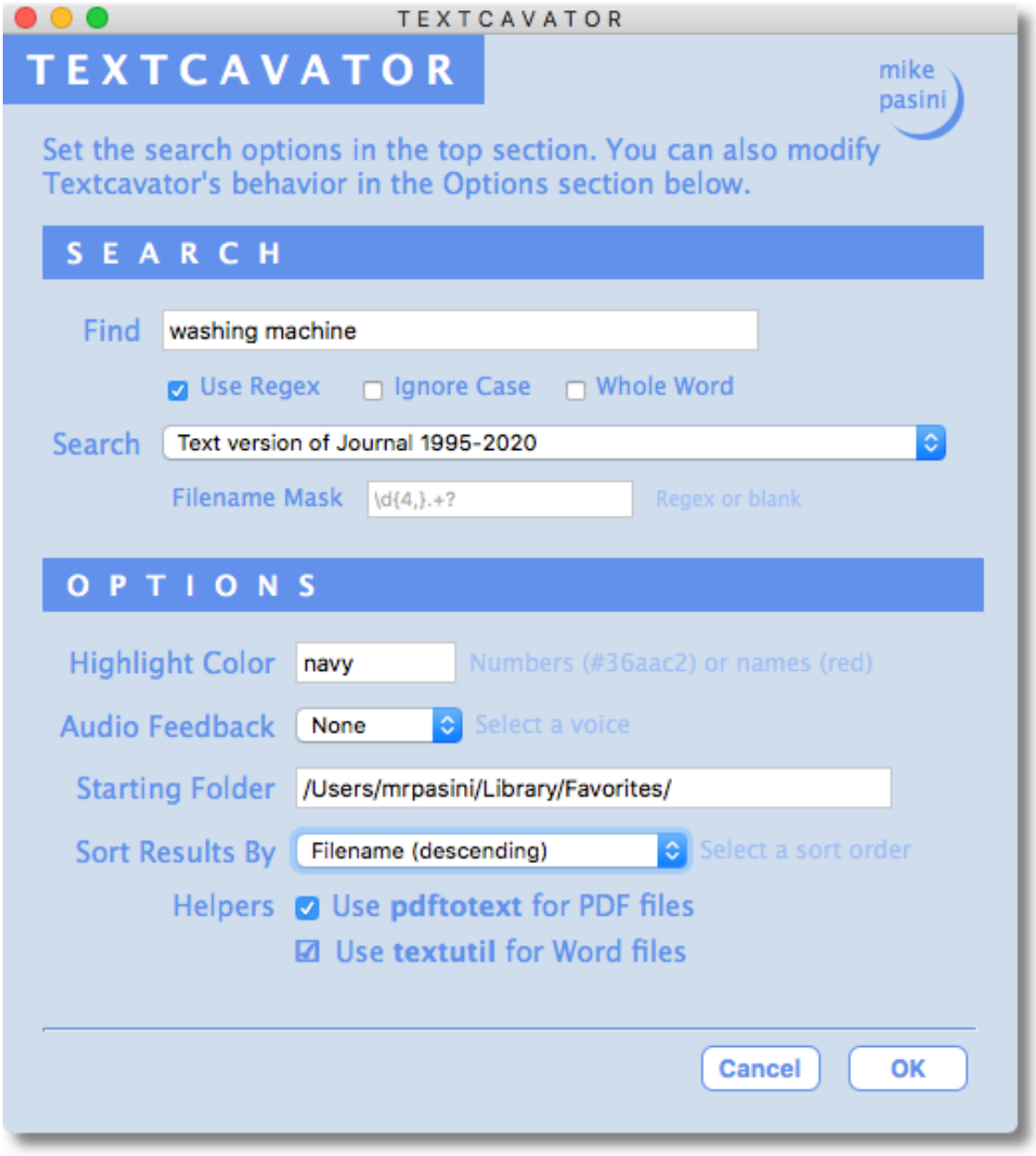

Textcavator is a very fast text retrieval tool that reports its results in context. It's like Google for your hard drive.

## INTRO

There are already *many* ways to scan local file content and the most powerful are free beginning with **`grep`** itself, which you can run from the command line in Terminal.

There are even free tools with a graphical user interface like DEVONtechnologies' free EasyFind.

So why Textcavator?



Most text search applications, including EasyFind, suffer an annoying limitation. They report filenames where the content has been found but don't show the content in context. Which obliges you to open a dozen or so files to see if the hit is actually the reference you want.

To display results in context of a search of my Web site files, I wrote a Perl CGI for the Web which I called Mike's Method, based on Perl code by Kevin Reid from 2000. That has run very nicely for many years.

Why can't I have it search my hard drive, too?

With Keyboard Maestro I can. It really does function like Google for my computer with one slight difference. There are a lot of files on a computer so Textcavator narrows down its focus to any directory and its subdirectories.

This version of Textcavator introduces a simplified Google-like interface by default (shown on the first page). Just type in the search term, press Enter and away you go.

It displays a prompt for what you're looking for and a line showing where you're looking. Searches enable regular expressions, ignore case, don't limit the search to whole words (which you can do using a regex) and ignore any file mask. The settings of the fuller prompt screen are protected and otherwise in use.

The search directory is either the currently selected or the one set in the more extended interface.

To see the more extended interface (shown in this chapter, an improvement on the original interface), with all the options just hold down the shift key when you press the trigger key.

It's a good idea to run the extended interface on the first run, just to set all your preferences.

## INSTALLATION

The .zip file contains the following files:

- CocoaDialog
- Textcavator macro with embedded Perl code
- Textcavator.pdf

To install Textcavator:

1. Copy **CocoaDialog** to your Applications folder
2. Import the **Textcavator macro**
3. Optionally install **pdftotext** to handle PDF files (recommended).

You can then configure the macro to reflect your environment and preferences.

To configure the macro:

1. Set the **hot key trigger** you prefer (without using the Shift key, which is reserved by Textcavator to select the extended interface)

2. Change the Default Values for the Variable "**TC Start Folder**" in the first Custom HTML Prompt to a list of your favorite locations, each ending in a forward slash ("/"). The first line provides a simple example of the HTML syntax:

   ```
   <option value="Other">Other</option>
   ```

   Here's how to point to your Desktop, using an informal explanatory tag ("Desktop"):

   ```
   <option value="/Users/[username]/Desktop/">MyDesktop</option>
   ```

   Substitute "[username]" with your own user name.

   The `description` between the > and `<` delimiters can be anything. The form passes the actual path listed in the `value` to the Perl script for processing.

   If you want a rule (or a space if you omit the line) between groups of options use this code:

   ```
   <option disabled>──────────</option>
   ```

   Alternately, you can use the `<optgroup label="[your label]"></optgroup>` tags around your options, adding a descriptive word for the label.

   Make sure `Other` remains in the list grouped with the long item just below it with the `value` "Last." That's where your last used folder is recorded. The

order isn't important to the code so make it convenient for yourself. **`Other`** with **`Last`** may be last if you don't tend to search random places, for example.

3. Confirm that **`$CD`** is set to the location you copied **CocoaDialog** in the **Execute Shell Script** action to the location you copied Textcavator.pl if you did not embed it.

PDF files can use character spacing that disrupts words themselves, making a search less accurate. You may, for example, look for "Value" in a PDF that renders it "(V)(alue)" to letterspace the word properly.

The free utility **pdftotext** will overcome that issue by converting the PDF to straight text. If you download and install it on your system, you can enable it in the Options section. Textcavator will then filter PDFs through pdftotext before searching the file.

After downloading the Mac 64-bit Xpdf tools from http://www.xpdfreader.com/download.html, you'll find **pdftotext** in the **`bin`** folder. Drag it to your Desktop and copy it to **`/usr/local/bin`** using this command, which will prompt for your admin password:

```
cp ~/Desktop/pdftotext /usr/local/bin sudo
```

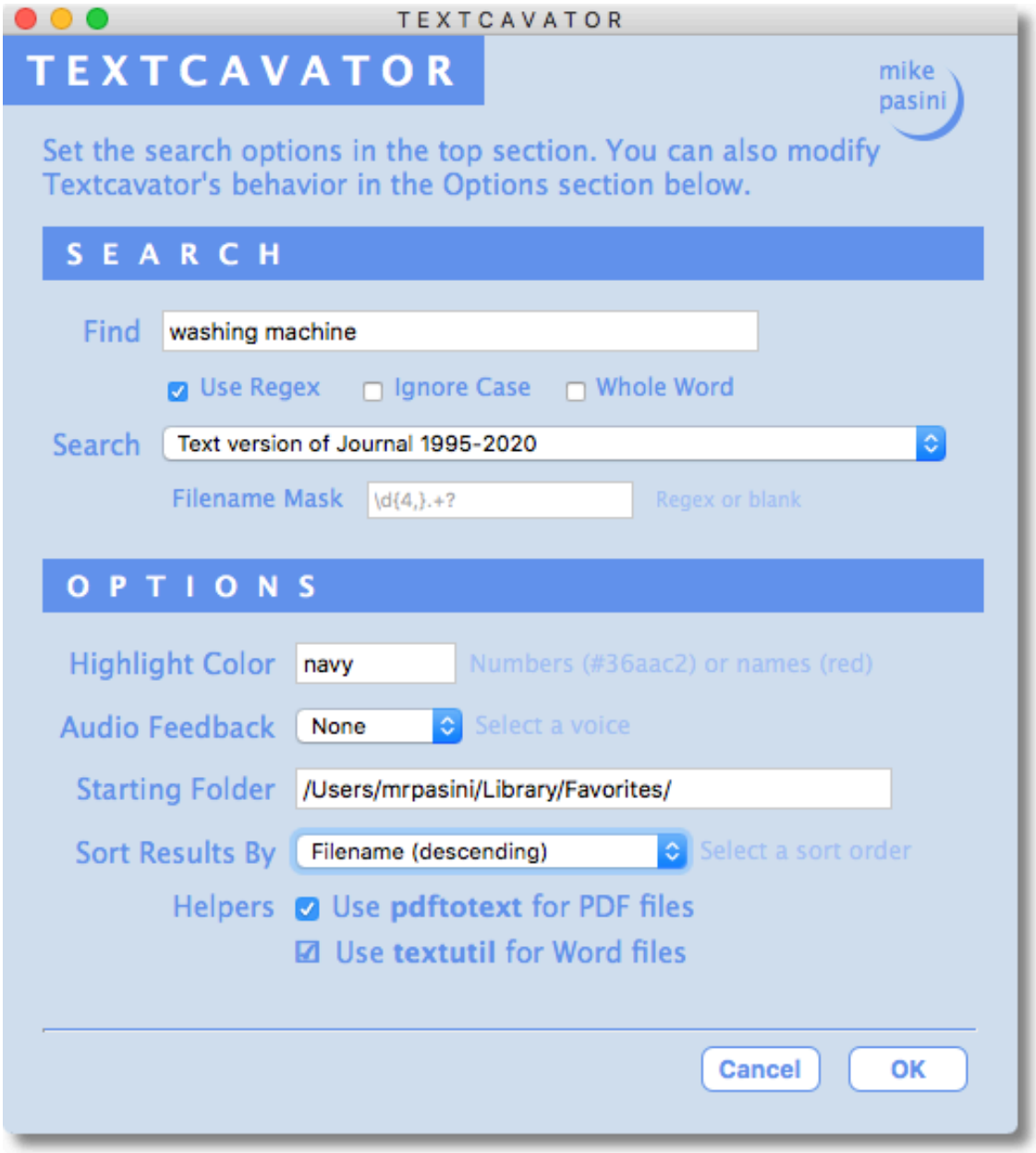
Note that not all PDFs permit that conversion. Textcavator will silently skip those.

If you install pdftotext in a location other than the default one, tell the Perl code about it.

That's it. The macro should now be set to respond to your trigger with a list of your preferred targets and handle PDFs more accurately.

## USING THE EXTENDED PROMPT

The extended prompt, like the first version of Textcavator, gives you complete control over Textcavator. It is just a fast as the Google prompt but remembers its settings.




When you first trigger the macro, some of its fields may be unpopulated. Subsequent runs will remember your entries from last time. When you use the Google prompt, these settings are preserved and restored.

The extended prompt has two parts: the essential Search parameters you'll use every run and a separate section for Options you use to configure Textcavator.

# Search

The top Search section of the form presents a prompt for three things:

1. A Search String with “Find” label
2. The option to Use Regex
3. The option to Ignore Case
4. The option to search Whole Words
5. A Start Folder
6. An optional Filename Mask

Let's look at each of them to see what Textcavator expects:

The **Search String** can be a literal like "Sam's vitamins" or a regexp like "(Sam|Ethyl)'s vitamins" to look for vitamins for either Sam or Ethyl. If you want to use a metacharacter like "?" literally, you can escape it with the backslash (“\”) or uncheck the **Use Regex** option so there are no metacharacters. Other metacharacters that require a backslash in a literal search include `[ ] { } ( ) . | ^ $ * \`.

If enabled, **Use Regex** will interpret what you enter as a regular expression. That will assign special meaning to the metachacters `[ ] { } ( ) . | ^ $ * \`. Disable this option to search literally for exactly what you typed in.

Note that in the Perl code, all searches are regex searches. So this simply determines whether special characters are escaped (don’t use regexp) before the regex search or not (do use regex).

If enabled, **Ignore Case** will match regardless of case. It is unaffected by the Regex setting. So "sam" will match both "Samantha" and "samovar." If you uncheck the option to disable it, only "samovar" would match “sam."

If enabled, **Whole Words** will not match words that contain your search string. If you look for ‘quote’ it won’t find ‘quotes.’ This is especially useful when search proprietary file formats like InDesign to avoid long lines of parameters that are not part of the text but which contain the search string.

**Start Folder** sets the root folder to begin the recursive search down through all the folders it contains. Every file will be opened and read by Textcavator as long as it has one of these extensions: `txt html? indd? qxp php pdf pl pdf rtf`. Here "`html?`" is a regexp that includes files ending in both "`htm`" and “`html`."

If you select `Other`, Textcavator will present the CocoaDialog file picker pointed at the default or starting directory you specified in the `Options` section of the prompt. Textcavator will remember which directory you picked and show it under `Other` on the popup list. If it was the last directory used, it (rather than `Other`) will be the default shown next time you run Textcavator.

Textcavator can't be pointed at an individual file (just open the file and use the application's `Find` tool if that's what you want). It does, however, look at all the files of the legal types that match any file mask in a directory and its subdirectories.

**Filename Mask** allows you to enter a regex to limit the search to files whose path and root name matches.

Say you have files with names like this:

> /Users/[username]/Documents/Journal/2020/03/25-Work.txt
> /Users/[username]/Documents/Journal/2019/07/15-Vacation.txt

And your Start Folder is:

> /Users/[username]/Documents/Journal/

But you only want to look at file from 2020. You'd use a File Mask of **2020** — simple as that.

If you have mixed file types in a directory, you can pick which ones to search by using the appropriate file extension in the mask. For example, **2020.+pdf$** will search just PDFs in 2020 while **2020.+(pdf|txt)$** will search both PDFs and text files in 2020.

The mask is applied to the whole path, root filename and the extension after the directory has been pruned of unsupported file extensions.

Filename Mask can also explain unexpected results. Textcavator remembers the mask between runs, so if you don't delete it, the last mask will be applied. That usually returns no matches, especially when you're certain there are some.

# Options

The Custom HTML Prompt also allows you to set some options in addition to the basic search parameters. Those include:

### HIGHLIGHT COLOR

You can set the highlight color for each or every run of Textcavator by entering either its RGB code (like "#36aac2" or HTML name (like "navy") in the prompt provided.

Use any color you like that will both contrast with the white background of the report and the light blue color of the text.

If you use **`cornflowerblue`**, the effect will be as if no highlighting has been done (because that's the color of the text itself). Try "**`navy`**" or "**`red`**" but avoid light colors like "**`yellow`**."

### AUDIO FEEDBACK

Textcavator can provide audio feedback. I find it useful for longer running searches, allowing me to flip to another screen and work on something else. I particularly like hearing Federica talk to me in Italian.

But Federica is a large, high-quality, optional voice that must be downloaded and installed. And Italian is an acquired taste.

A selection of standard male and female voices is available in the popup. And they all speak English. Think of them as a staff of assistants.

You can disable the audio by simply selecting "**None**" from the list.

**STARTING FOLDER**

If you select **Other** as your Search target, the file picker uses the location you enter in **Starting Folder** as its default. I find it handy to default a Custom search to my Favorites folder, for example, so I don't have to add all of the paths in it to the popup menu.

**SORT RESULTS BY**

There are six ways to sort the results of a search:

- Filename (ascending)
- Filename (descending)
- Creation Date (ascending)
- Creation Date (descending)
- Modification Date (ascending)
- Modification Date (descending)

If your filenaming strategy uses dates (like “2020.10.05 something.txt”) and each file has a different date, the first two options should suffice.

But if more than one item might share the same filename date, you can use the Creation Date options to sort them in the order they were created.

And if you are more interested in listing the most recently edited files in order, the last two options are what you want.

**HELPERS**

There are two Helper apps that filter files for more accurate results.

- **pdftotext** reads PDF files and writes text files
- **textutil** reads Word files and writes text files

You will have to download and install the free pdftotext, as explained above. But texutil is built into macOS. That’s why the checkbox isn’t live. Textcavator will always use it.

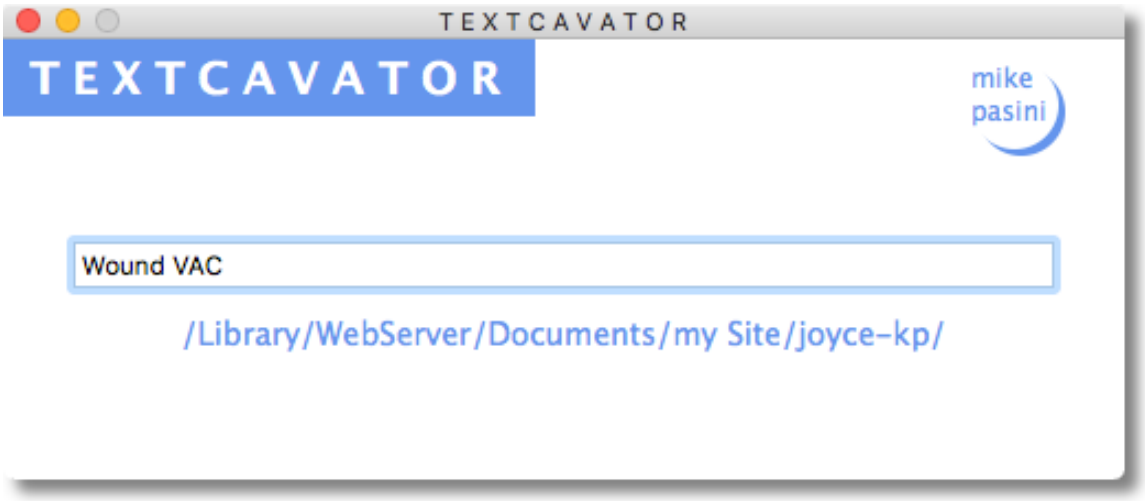
When a PDF or a Word file is passed to Textcavator, it calls the appropriate utility to convert the original file into a temporary text file that Textcavator processes normally. Textcavator then deletes the temporary file.

Hover over the text description for the details. Click on the bold pdftotext to go to the Web site to download it.

## USING AND CONFIGURING THE GOOGLE PROMPT

This version of Textcavator defaults to a Google-like search prompt with default options already set. In many cases, it may be all you need.

It displays a prompt for what you're looking for and the Search target.




Searches from this prompt enable regular expressions, ignore case, don't limit the search to whole words (which you can do using a regex) and ignore any file mask. The settings of the fuller prompt screen are protected and otherwise in use.

The search directory is either the currently selected folder in the Finder or the one set in the more extended interface. If nothing has been set, it will use the default Start Folder. And if that hasn't been set, it will point to the Desktop.

So the most efficient way to use this quick search option is to select a folder in the Finder and hit the trigger to bring up Textcavator.

### MODIFYING DEFAULT BEHAVIOR

You can easily modify the default behavior of this prompt by editing the values in the macro itself.

The default behavior reflects the simplest query you might form. It sets the following options:

- **Search String**: Not saved
- **Use Regex**: Off
- **Ignore Case**: On
- **Whole Word**: Off
- **Filename Mask**: None

You can edit the actions that set these defaults to suit your own preference.

The action Set Variable “TC Regex” to Text “0” in the group Set Options for Google Prompt is where regex is disabled. If you prefer to enable it for all searches with this prompt, set it to 1 in the action.

Settings that affect the report itself are not changed by using this interface.

The behavior dictated by these options has no effect on the extended prompt’s last settings, which are preserved and restored when you use the Google prompt. So you have two personalities active with this version of Textcavator.

Beyond these modifications, you can further customize the macro. The next section explains how to do that.

# CUSTOMIZATION

You can customize Textcavator beyond setting options by editing the macro or KM-Textcavator.pl.

A couple of useful customizations include adding file types and changing the location of CocoaDialog. And you can certainly change the hot key trigger.

### FILE TYPES

As shipped, Textcavator will only open files whose extensions include **`txt`**, **`doc`**, **`docx`**, **`htm`**, **`html`**, **`ind`**, **`indd`**, **`qxp`**, **`php`**, **`pl`**, **`pages`**, **`pdf`** and **`rtf`**.

You can include other extensions by adding them to the list in the subroutine "**`wanted`**" at the very end of KM-Textcavator.pl.

Word files are preprocessed using Apple's textutil software (included with OS X since version 10.4) into text files. HTML files are stripped of code, RTF files are stripped of their own codes and any file not listed explicitly as a text file is stripped of unprintable characters. To add a text file format to the exclusions list, search for "**`# preprocessing`**" in KM-Textcavator.pl and add it to list.

It's safe to add the extension for any text file but proprietary files are all different and may cause issues even with just printable characters. If you don't add the extension to the preprocessing section, it will be treated like a proprietary file.

Saving the output file should help diagnose any problem. In fact any runtime error encountered will show up in the HTML report.

### COCOADIALOG LOCATION

If you copy CocoaDialog to your main Applications folder, where it will be accessible to any user on your system, you don't have to edit the **`$CD`** variable in KM-Textcavator.pl. If you prefer to put it somewhere else, change the **`$CD`** variable location in the Perl code.

### TRIGGER

The base hot key trigger I use (a 'T' for Textcavator) assumes the macro is on a palette with no other T's. With the Shift key, it brings up the alternate, more elaborate interface.

You may prefer a function key, key chord or some other trigger, of course.

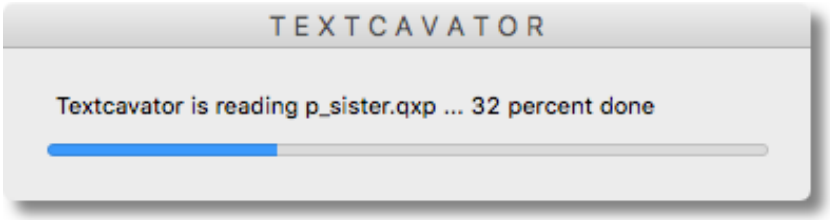
The only thing to keep in mind if you set a different trigger is that the base trigger is modified by the Shift key in the If action to bring up the more elaborate extended interface with all the options.

# RUNNING TEXTCAVATOR

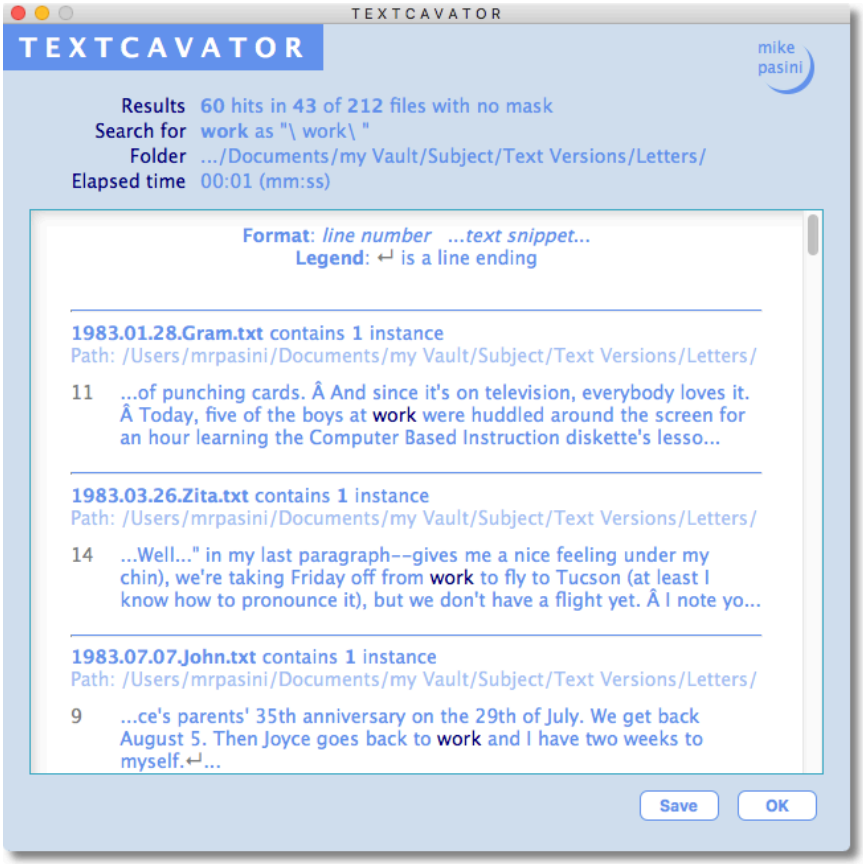
After entering your options on the extended prompt, click OK to start Textcavator. On the Google prompt, just press enter after entering your search term.

An indeterminate file progress bar (it throbs) is displayed as Textcavator builds a file list. Sometimes this takes a few seconds and you'll see the progress bar and other times you'll just see the window flash (it happens so fast).

A progress bar shows the current file being searched and what percentage of the files TextCavator has processed.




When it has finished processing the files, it will present a window with its report. The report includes the number of files with the search phrase and how Textcavator rendered it as a regex, the total files, any filename mask, the number of total occurrences and the elapsed time.



In testing on a spinning hard disk, Textcavator was able to go through over 4,200 HTML files in just over a minute (1:04) on the first run and 37 seconds on the second. Textcavator took 3:55 minutes to read through 756 long (20+ pages each) QuarkXpress and InDesign documents in multiple directories.

The elapsed time is shown in the header section of the report.

The window also includes a **Save** option to write the HTML report to disk. You'll be prompted for a filename and location.

## NOTES

Scanning **proprietary formats** like QuarkXPress and InDesign documents requires stripping non-printing characters (except for a few) from the text stream (but not the file itself, which is left untouched).

These files also encode typographic niceties like l**igatures, quotes and dashes**, so you won't be able to search for them. They will also be rendered oddly in the report.

**Line numbers** for page layout programs are meaninglessly high. They may look more like character counts. Ignore.

Loading CocoaDialog (for the file picker or the progress bars) takes a few seconds, slowing things down. So does audio feedback. Once cached the **delay** disappears, so second runs are usually much quicker.

Although this is version 2.0a, it's still a **work in progress**. The macro is solid but the filtering routines can always be improved. Meanwhile, though, it's been very helpful to me so I'm releasing it in its current state.

If **no report form** is produced, the likely culprit is a file that failed to open. The Engine log for Keyboard Maestro will have the details.

Happy to discuss further **customizations** or feedback on the interfaces. And pleased to see any code revisions you make, particularly for other file formats.

# RELEASE NOTES

**v2.0a on 31 August 2020**

- Use Regex checkbox
- Whole Word checkbox
- Filename Mask
- Redesigned Main Window
- Redesigned Report Window
- Added six options for sorting results
- Added Google-style search window alternative

**v1.1a on 31 March 2018**

- Added support for Microsoft Word doc and docx formats via Apple's textutil.

**v1.0b on 1 March 2018**

- Corrected the Keyboard Maestro variable name for the default folder that did not have the TC precede.
- Added a tip in this documentation about describing folder targets.

**v1.0a on 22 February 2018**

- Initial release.

# CONTACT

You can reach me at http://mikepasini.com. Or by clicking the Moon icon on any of Textcavator's HTML pages.