

# FLExTools Help

For FLExTools Version 2.2,  
and FieldWorks version 9

25 April 2023

## ***Introduction***

FLExTools is a framework for running custom Python scripts on FieldWorks Language Explorer projects. FLExTools provides core functions for creating small Python scripts (a *module*) to do the processing work, and a user interface to make it easy to run the modules individually or as a set. A set of modules is called a *collection*.

This guide explains how to use the FLExTools application. Further information on programming custom FLExTools modules can be found on the Help menu:

- Programming Help
- API Help

## ***Choosing a project***

When FLExTools is run the first time there is no FieldWorks project selected. Use the menu FLExTools | Choose Project or click the Project button on the toolbar. In the Choose Project dialog double-click on a project to select it.

Notice that the status bar at the bottom shows the selected project's name.

## ***Viewing module information***

The main window shows two reporting modules that are included with the installation. You can see a module's documentation by selecting it and clicking the *Module Info* toolbar button, or with the Ctrl-I shortcut key.

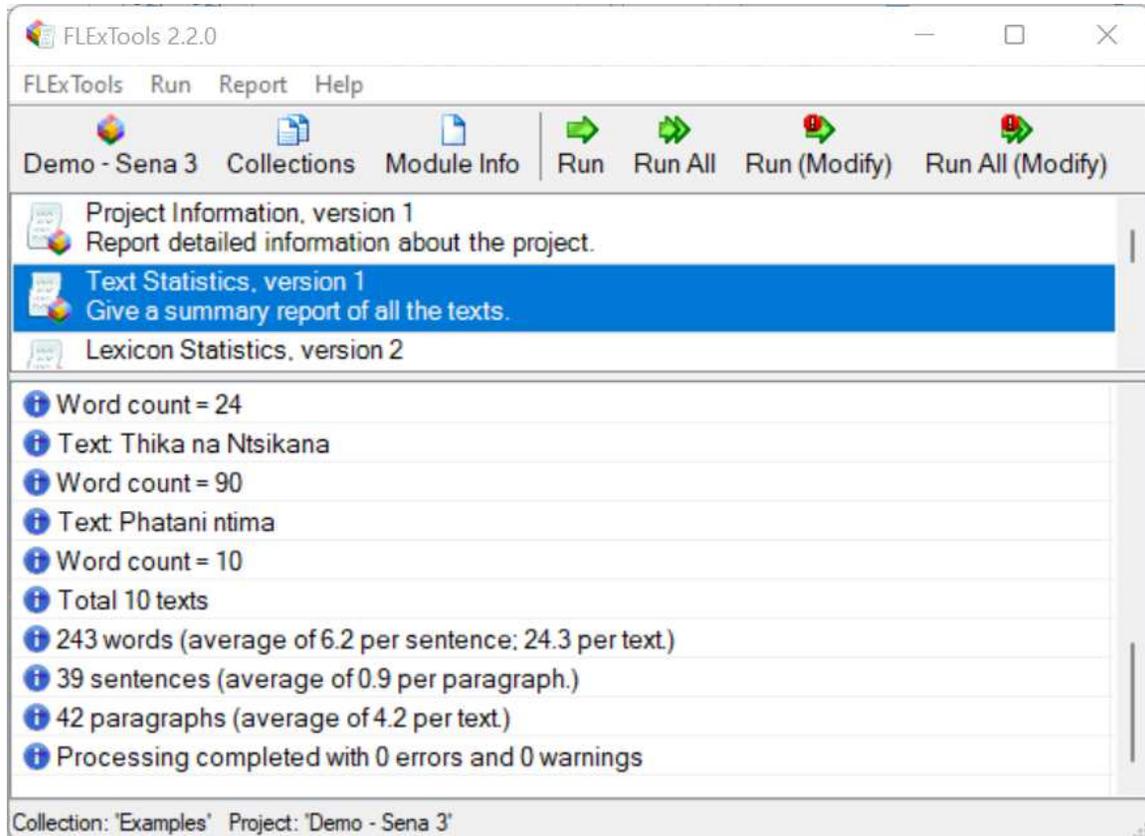
(Note, these modules are configured in the *Examples* Collection. The status bar at the bottom shows the current collection name. See the collections section later for more information about managing collections.)

## ***Running modules***

Having chosen a project, run the *Text Statistics* module. There are four ways to run a module:

- double-click the module;
- select the module and click the *Run* button in the toolbar;

- select the module and use the Run command in the Run menu;
- select the module and use the Ctrl-R shortcut key.



The results appear in the report panel like this:

To run all of the modules in the list use the *Run All* toolbar or menu options.

### ***Project sharing***

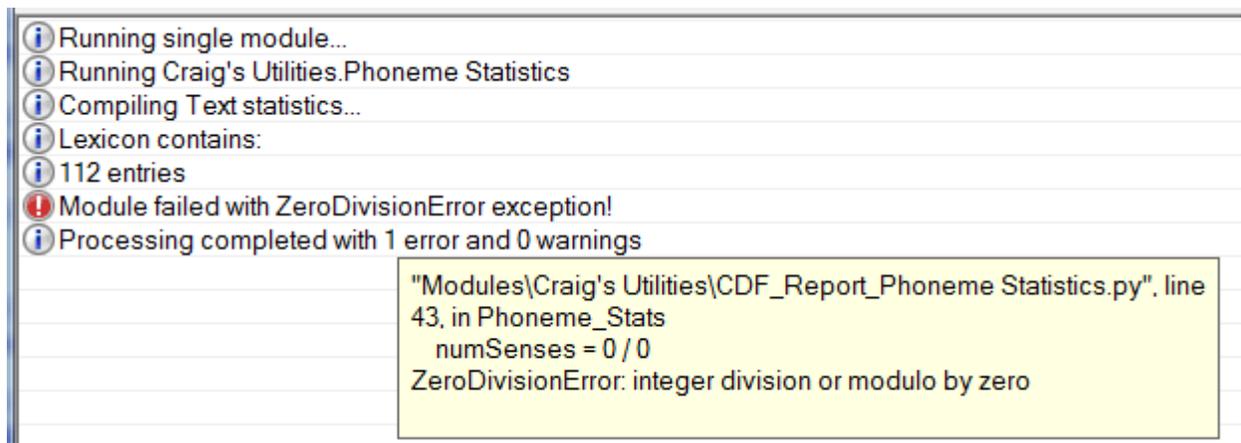
If you try running a module on a project that is open in FieldWorks, then you will get an error message about the project being in use by another program. You must either close the project in FieldWorks, or configure the project to allow multiple programs to access it at the same time. To do this, in Fieldworks use the menu File | Project Management | FieldWorks Project Properties, and select the Sharing tab. Tick the box for "Share project contents with programs on this computer."

## Report window

The report panel can show three types of message with the following icons:

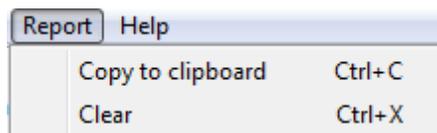
- Information 
- Warning 
- Error 

Move the mouse over an entry to see more information in a tooltip. For example, a programming error might look like this:



Some messages in the report window contain links to the relevant entry or place within FieldWorks. If this is the case a tool tip will appear when you hover over the message, and double-clicking that line will open FieldWorks at the right place.

Under the Report menu there are options to clear the report window and to copy the contents to the clipboard:



## Modifying the project

For some situations you will want a FLEXTools module to make changes to the FieldWorks project. This might be to make automated fixes to data, to insert notes in a lexical entry about an issue that needs to be addressed, or some other specific need.

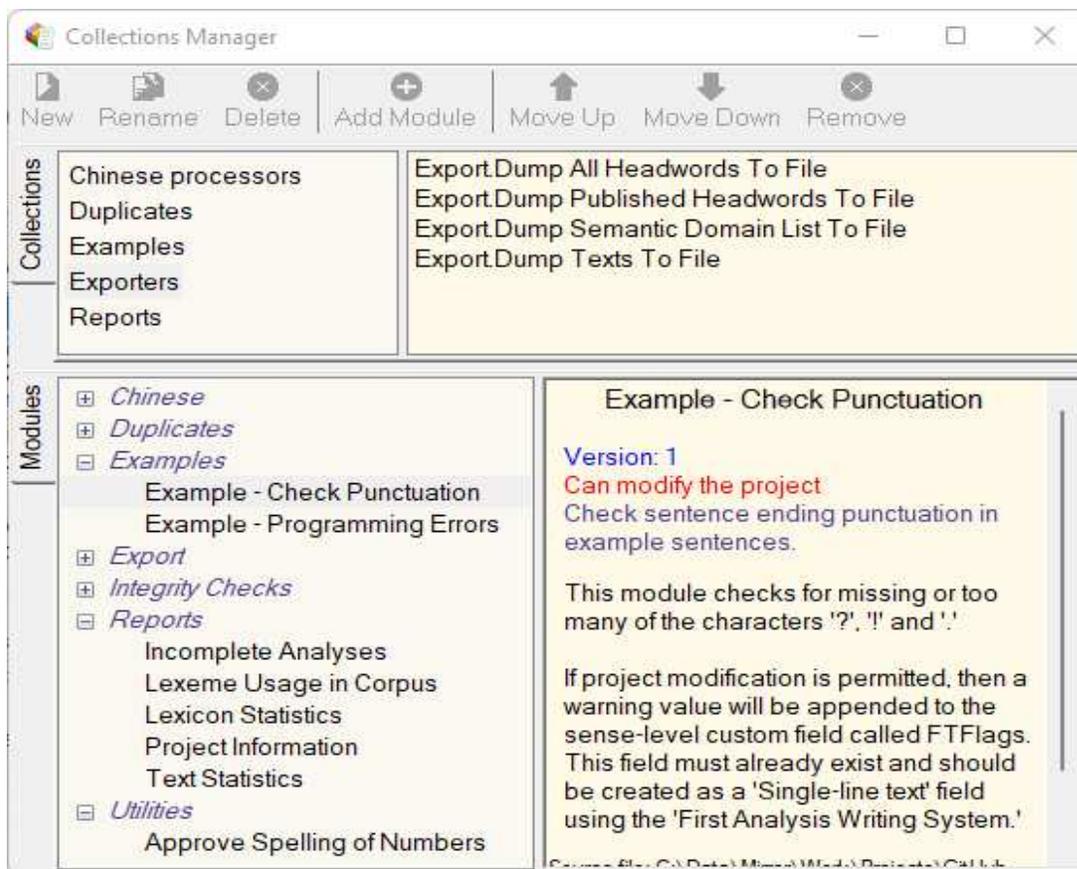
Be *very careful* when making changes to the project via FLEXTools or any Python script. Back up your project before attempting any changes and check the results carefully.

In FLExTools, any module that can change the project shows this in the main list and in the Module Info dialog. As a safeguard for your data, FLExTools will not make changes to the project *unless* you are holding down the *Shift* key when you invoke the *Run* or *Run All* commands, or if you click the *Run (Modify)* or *Run All (Modify)* toolbar buttons. This allows you to do a dry run or preview of a module to check that it is doing what you expect. Then you can run it again with *Shift* down to actually make the changes. (See the *FLExTools Programming* help file for important information on implementing this behaviour in your own modules.)

## Collections

FLExTools modules are organised into *collections*. A collection is a set of modules that you want to group together. The current collection name is shown in the status bar. For a fresh installation the “Examples” collection will be active.

To create new collections or edit existing ones, open the Collections Manager either with the *Collections* toolbar button, with Ctrl-L, or via the FLExTools | Collections menu. The Collections Manager looks like this:



- The top panel shows all the collections, with the right side showing the modules that are in the selected collection.

- The bottom panel shows all the modules that are installed, with information about the selected module on the right.
- The toolbar buttons are enabled and disabled according to where the input focus is and the functions available.

To learn how to use the Collections Manager follow the steps below to create a new collection called "Quality checks." This will contain modules that you want to run on a regular basis to check certain aspects of consistency or correctness in your project.

1. Click in the top-left area—the collections list—and click the *New* button. Type in "Quality checks" for the name.
2. Choose the modules you want to be in this collection from the bottom panel: select the *Check Punctuation* module in the *Examples* group and click the *Add Module* button, or double-click the module name.
3. Add other modules to the collection as required. In the list in the top-right panel use the *Move Up* and *Move Down* buttons to change the order. You can also use the *Remove* button to remove a module from the collection.
4. Finally, double-click the collection name to select this as the current collection and return to the main screen.

You will now see the *Check Punctuation* module in the top list, along with the other modules that were added at step 3 above. The collection name "Quality checks" is shown in the status bar. Click *Run All* to run all of the modules in this collection on the current project.

### ***Installing modules***

In the FLEXTools installation folder there is a folder called 'Modules.' Within this folder you can create a 1-level hierarchy of sub-folders for grouping your modules into logical categories. In the Collections Manager you will see that the modules are grouped according to the directory structure within the Modules directory. For example, 'Examples' and 'Reports' in the screen-shot above correlate to folders in the Modules directory.

Copy your modules (Python script files) into the modules folder or any sub-folder. See [Help | Programming Help](#) for details on how to write a module.

### ***Configuration options***

These options can be changed in the flextools.ini file. They are case-insensitive.

WarnOnModify	Set to False to disable the confirmation message when running in modify mode.
StopOnError	Set to True to stop running any more modules if a module reports an error.
SimplifiedRunOps	Set to True to hide the dry-run/preview buttons and always run with changes enabled. This is independent of WarnOnModify, so if WarnOnModify is True the confirmation message will still be displayed.