

Configuring Crimson Editor to work with Python on Windows

You can use whatever editor you like to write Python scripts, as long as the files are saved as plain text (you could even use something like Microsoft Word if you made sure it was saving the file correctly, but there are *much* better options). You can even use PythonWin if you like, although you will not be able to run your scripts for this course using PythonWin in the computer lab (although you can at home if you're willing to install all of the necessary libraries). My favorite free editor is Crimson Editor and that is what we will be using in class. These directions will help you configure Crimson Editor so you can run Python Scripts directly from the editor, whether you're using FWTools or another Python installation.

The first thing you need to do, obviously, is get Crimson Editor from <http://www.crimsoneditor.com/> and install it.

FWTools version of Python

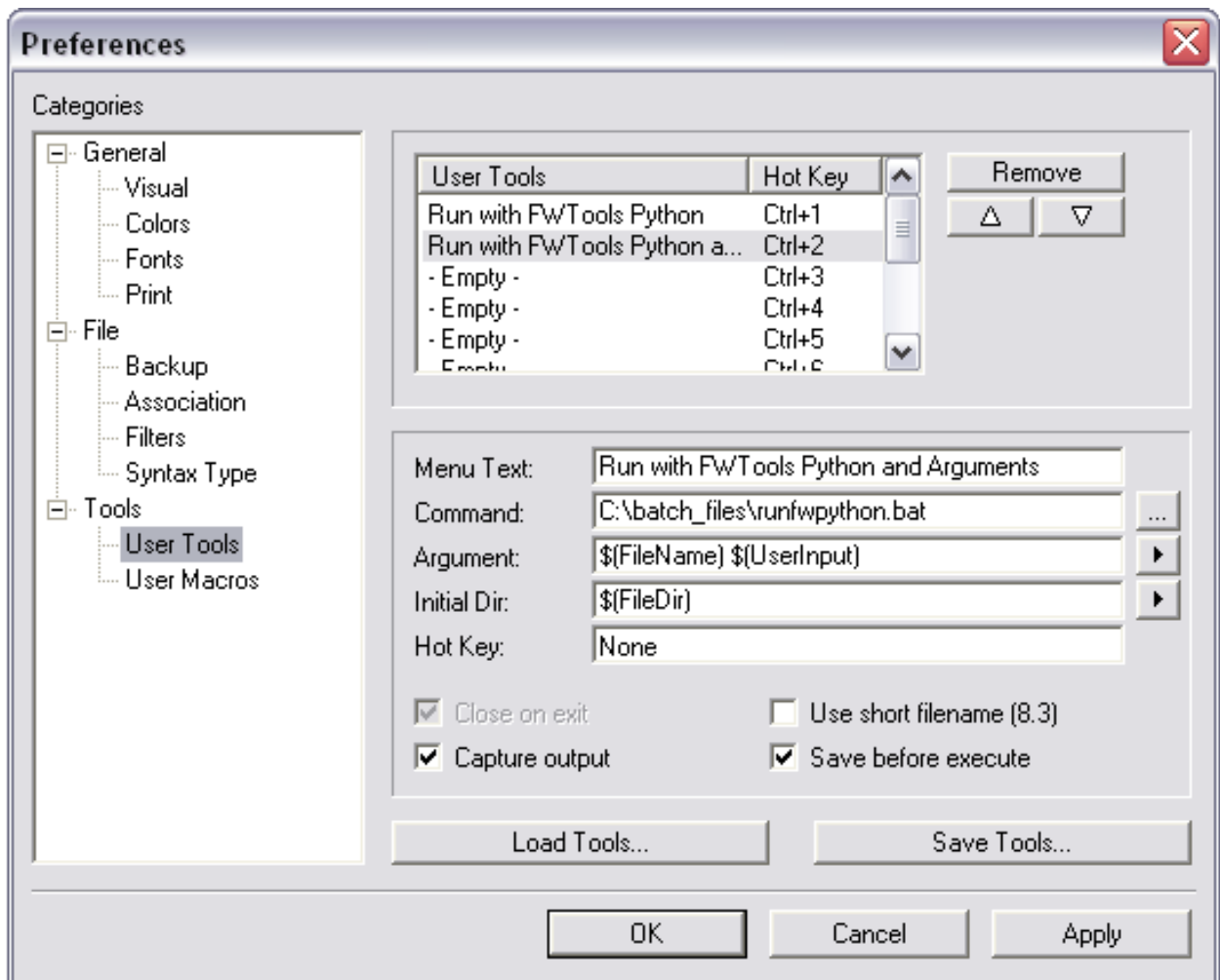
Installing FWTools is the easiest way to get the software needed for this course, but configuring Crimson Editor to run the FWTools version of Python is slightly more involved. These directions should get you through it, though.

1. Get FWTools from <http://fwtools.maptools.org/> and install it.
2. Create a file called `runfwpython.bat` (or whatever you want, but make sure you use the correct name in the steps below) and save it somewhere on your hard drive. On my system, I put it in `C:\batch_files\`. Put the following code in the file, changing the path to `setfw.bat` (which comes with FWTools) as needed to match your installation. Make sure the file is saved as plain text with a `.bat` extension.

```
call "C:\Program Files\FWTools2.2.8\setfw.bat"  
python %*
```

3. Choose *Conf. User Tools...* from the Tools menu in Crimson Editor.
4. To create a tool that does not request the user to enter parameters for the script:
 - a. Highlight one of the *-Empty-* entries in the User Tools list.
 - b. Put whatever you want in the Menu Text section. This is the text that will be shown in the Tools pull down menu. I use *Run with FWTools Python*.
 - c. Put the full path and filename for the file created in Step 2 in the Command section. In my case, this is `C:\batch_files\runfwpython.bat`.
 - d. Put `$(FileName)` in the Argument section (you can do this with the arrow next to the textbox).
 - e. Put `$(FileDir)` in the Initial Dir section (again, you can do this with the arrow).
 - f. Set a new Hot Key if you like. I've always used the defaults, which are shown in the list of tools at the top of the dialog.
 - g. Make sure the *Capture output* and *Save before execute* boxes are checked.
5. To create a tool that requests parameters from the user:
 - a. Highlight one of the *-Empty-* entries in the User Tools list.

- b. Put whatever you want in the Menu Text section. This is the text that will be shown in the Tools pull down menu. I use *Run with FWTools Python and Arguments*.
 - c. Put the full path and filename for the file created in Step 2 in the Command section. In my case, this is `C:\batch_files\runfwpython.bat`.
 - d. Put `$(FileName) $(UserInput)` in the Argument section, making sure there is a space between the two parts (you can add these with the arrow next to the textbox).
 - e. Put `$(FileDir)` in the Initial Dir section (again, you can do this with the arrow).
 - f. Set a new Hot Key if you like. I've always used the defaults, which are shown in the list of tools at the top of the dialog.
 - g. Make sure the *Capture output* and *Save before execute* boxes are checked.
6. Now you can run a script by selecting one of the tools you just created from the Tools pull down menu in Crimson Editor.



Another version of Python

You can also configure Crimson Editor to run other versions of Python, such as the one that gets installed with ArcGIS (and then you can use Crimson Editor instead of PythonWin, if you like).

1. Choose *Conf. User Tools...* from the Tools menu in Crimson Editor.
2. To create a tool that does not request the user to enter parameters for the script:
 - a. Highlight one of the *-Empty-* entries in the User Tools list.
 - b. Put whatever you want in the Menu Text section. This is the text that will be shown in the Tools pull down menu. I use *Run with ArcGIS Python*.
 - c. Put the full path and filename for the file created in Step 2 in the Command section. In my case, this is `C:\Python25\python.exe`. If you have ArcGIS 9.2 then your Python will be 2.4 rather than 2.5.
 - d. Put `$(FileName)` in the Argument section (you can do this with the arrow next to the textbox).
 - e. Put `$(FileDir)` in the Initial Dir section (again, you can do this with the arrow).
 - f. Set a new Hot Key if you like. I've always used the defaults, which are shown in the list of tools at the top of the dialog.
 - g. Make sure the *Capture output* and *Save before execute* boxes are checked.
3. To create a tool that requests parameters from the user:
 - a. Highlight one of the *-Empty-* entries in the User Tools list.
 - b. Put whatever you want in the Menu Text section. This is the text that will be shown in the Tools pull down menu. I use *Run with ArcGIS Python and Arguments*.
 - c. Put the full path and filename for the file created in Step 2 in the Command section. In my case, this is `C:\Python25\python.exe`. If you have ArcGIS 9.2 then your Python will be 2.4 rather than 2.5.
 - d. Put `$(FileName) $(UserInput)` in the Argument section, making sure there is a space between the two parts (you can add these with the arrow next to the textbox).
 - e. Put `$(FileDir)` in the Initial Dir section (again, you can do this with the arrow).
 - f. Set a new Hot Key if you like. I've always used the defaults, which are shown in the list of tools at the top of the dialog.
 - g. Make sure the *Capture output* and *Save before execute* boxes are checked.
4. Now you can run a script by selecting one of the tools you just created from the Tools pull down menu in Crimson Editor.

