Use TonePars with FLEx User Documentation

H. Andrew Black
SIL International
blackhandrew@gmail.com
20 February, 2023
Version 0.2.0 Beta

Contents

1 Introduction

Use TonePars with FLEx is a tool that works as a utility in *FieldWorks Language Explorer* (aka *FLEx*). *Use TonePars with FLEx* runs the *XAmple* program followed by the *TonePars* program on a text or a portion of a text that exists in a *FLEx* project. You tell *Use TonePars with FLEx* the *TonePars* rule file to use as well as an *XAmple* input control file. Then you can choose a text or a portion of that text and ask *Use TonePars with FLEx* to process it. The result will show in *FLEx* the same as it does when using either of the two morphological parsers that come with *FLEx*.

The input to the *XAmple* program is the same as what *FLEx* uses for the default morphological parser (which is *XAmple*). This means that you must control *XAmple* using the capabilities *FLEx* offers, not what you may have used with *AMPLE* via, say, *CARLAStudio*.

Use TonePars with FLEx works with version 9.1.18 or higher of *FLEx* and is only available on 64-bit Windows computers.

1.1 Installation

To install *Use TonePars with FLEx*, obtain the installer from https://github.com/sillsdev/pcpatrflex/blob/TonePars FLEx/InstallerTonePars/Output/UseTonePars WithFLExDllSetup.exe. The installer will check to see if you have installed *FieldWorks Language Explorer* version 9. If not, it will abort.

We recommend that you close *FieldWorks Language Explorer* before running the *Use TonePars with FLEx* installer.

1.2 Invoking *Use TonePars with FLEx* from within *FLEx*

While running *FLEx*, use Tools menu item / Utilities.... Find the "Use TonePars with FLEx" item, check it, and then click on the "Run Checked Utilities Now" button.

1.3 Initial invocation

The first time you invoke *Use TonePars with FLEx* on a *FLEx* database, it will automatically add to your *FLEx* database the following:

1. a custom field to each sense (called "ToneParsSense");
2. a custom field for each allomorph/lexeme form[1] (called "ToneParsForm"); and
3. a custom list (called "TonePars Properties"). You use this custom list to create any allomorph or morpheme properties used in your *TonePars* rule file.

The names shown above are always in the English analysis writing system and English is the only writing system containing these names.

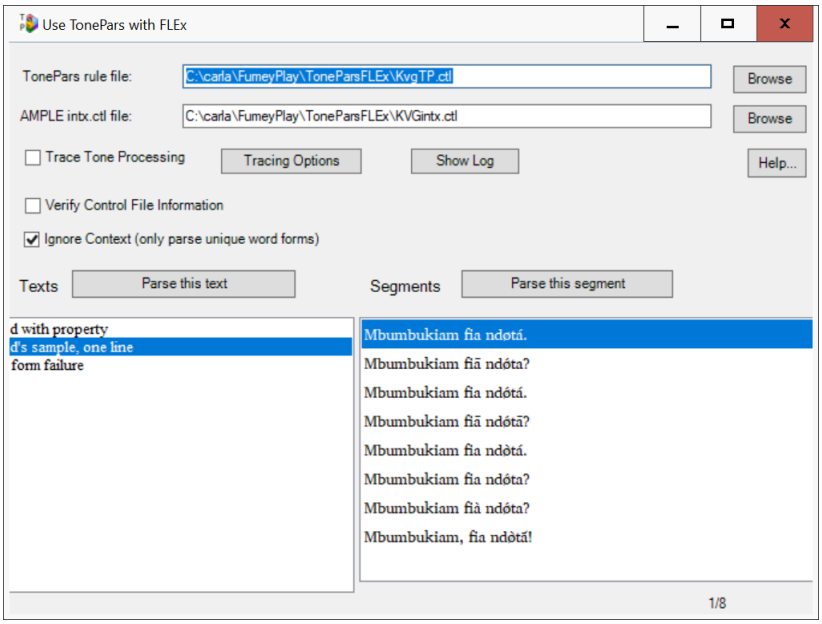
You can find the custom list by clicking on the "Lists" button in *FLEx*.

[1] The current version of *FLEx* does not show this custom field on Lexeme Form. You can set it, though, by swapping the lexeme form with an allomorph. See the *FLEx* help system for how to do this.

1.4 Appearance

Use TonePars with FLEx looks like what is shown in (1).

(1)



The texts in the *FLEx* database are shown in the left pane and the segments of the first text are shown in the right pane.

There are buttons you can click. Each is discussed in section 2 below.

2 Buttons and check boxes

You control *Use TonePars with FLEx* by using the various buttons and check boxes. This section briefly describes them.

2.1 *TonePars* rule file Browse button

To choose which *TonePars* rule file to use, click on the topmost Browse button. By convention, *TonePars* rule files have an extension of ".ctl" so this is what the file browser uses.

2.2 *AMPLE* intx ctl file Browse button

As you most likely already know, when using *TonePars*, one first parses a text via *AMPLE* but as part of the processing, *AMPLE* strips out tone marking. The result is then passed to *TonePars*. In order to correctly strip out the tone markings, *Use TonePars with FLEx* needs to know the location of the input text control file needed. To choose which input text control file to use, click on the Browse button. By convention, *AMPLE* input text control file names end with "intx.ctl" so this is what the file browser uses.

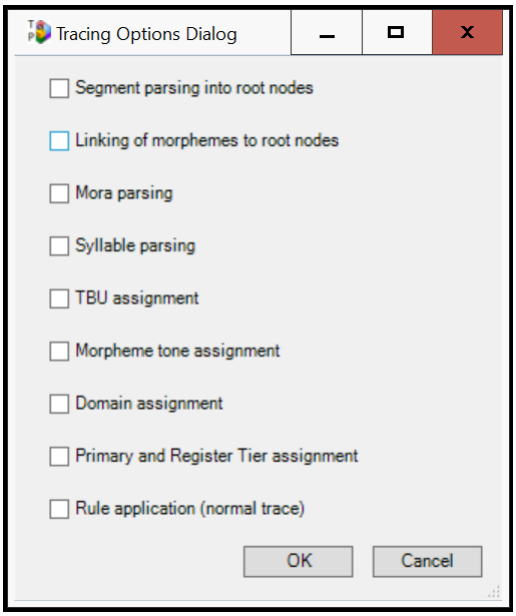
2.3 Trace Tone Processing check box

The next item is a check box with a label of "Trace Tone Processing." When working with *TonePars*, you often need to get a trace of what the tool is doing. When this check box is checked, *Use TonePars with FLEx* invokes *TonePars* with tracing turned on. The log file will show the results of the tracing process. See section 2.4 for the various tracing options available and see section 2.5 for how to see the resulting log file.

2.4 Tracing Options button

When you click on the "Tracing Options" button, it brings up a dialog box that looks like what is in example (2):

(2)



The options listed are the same as the options available in *CARLAStudio*.

2.5 Show Log button

When you click on the "Show Log" button, the log file generated by the last invocation of *TonePars* will be displayed.

2.6 Verify Control File Information check box

One run time option for *TonePars* is to verify various pieces of information. When the "Verify Control File Information' check box is checked, *Use TonePars with FLEx* will invoke *TonePars* in such a way that this information will be included in the log file. You can see it by showing the log file (see section 2.5).

2.7 Ignore Context check box

When the “Ignore Context” check box is checked, *Use TonePars with FLEx* will determine all the unique word forms in the text (or segment) and parse them. This means that each unique word form is parsed once and only once no matter how many times it appears in the text (or segment). It therefore runs much faster especially on a text. This is the default setting.

When this check box is not checked, then the input to parsing is like it is for *AMPLE*: Each word is parsed in turn, even if it occurs multiple times. So this takes longer to parse. On the other hand, if your tone rules need to go across word boundaries,[2] then you may need to process texts (and segments) this way.

2.8 Help button

The “Help...” button is used to get this user documentation file, show the TonePars Manual, show the TonePars Grammar documentation file, or show the “About” dialog box.

2.9 Parse this text button

Above the pane containing the texts is a button labeled “Parse this text.” You use this button to parse this entire text via *XAmple* and then *TonePars*. The mouse icon will change to the “busy” shape until it is done. The results will show in *FLEx* the same way as using one of the morphological parsers that come with *FLEx* show their result. It is easiest to see this in the “Texts & Words" / “Interlinear Texts" view or the “Texts & Words" / “Word Analyses” view.

During the parsing process, the bottom left of the window will display the current step that is occurring.

2.10 Parse this segment button

Above the pane containing the segments of the selected text is a button labeled “Parse this segment.” You use this button to parse this particular segment via *XAmple* and then *TonePars*. The mouse icon will change to the “busy” shape until it is done. The results will show in *FLEx* the same way as using one of the morphological parsers that come with *FLEx* show their result. It is easiest to see this in the “Texts & Words" / “Interlinear Texts" view or the “Texts & Words" / “Word Analyses” view.

During the parsing process, the bottom left of the window will display the current step that is occurring.

[2] That is, if some of your tone rules are edge rules…

3 Maximum analyses setting for *XAmple*

By default, *FLEx* only returns a maximum of the first twenty parses found by *XAmple*. This is often very reasonable when one is not using *TonePars*. With *TonePars*, however, this could easily be too few. In one *TonePars* project, there are twenty-six nulls possible for various tone possibilities within a given word. Only returning twenty will never do. We have changed the default to be 1000. If this is ridiculously high for your situation, you can always change the setting to a different number. To do so, in the main *FLEx* window, use the Parser / Edit Parser Parameters… menu item and set the "MaxAnalyses" value to what you need. Note that a value of -1 will be treated as 1000 by *Use TonePars with FLEx*.

4 Restarting *Use TonePars with FLEx*

Whenever you exit and restart *Use TonePars with FLEx*, it will do the following:

1. remember the size and position of the *Use TonePars with FLEx* window;
2. remember which *TonePars* rule file you last chose;
3. remember which *AMPLE* intx ctl file you last chose;
4. remember the settings of "Trace Tone Processing," "Tracing Options." and "Verify Control File Information;"
5. remember which text in that project you last selected; and
6. remember which segment in that text you last selected.

5 Known problems

The following items are known to be less than desirable with this version of *Use TonePars with FLEx*:

1. *Use TonePars with FLEx* only works with version 9.1.18 or higher of *FLEx*.
2. The location of the *TonePars* rule file and the *AMPLE* intx ctl file work best if the path to them does not contain any spaces.
3. If a lexical entry in *FLEx* is marked as either a proclitic or an enclitic, it may not parse correctly. This is because *FLEx* creates two entries for it with the same morphname; one is as an affix and the other is as a root. *Use TonePars with FLEx* may not process it correctly due to this ambiguity. It might be possible, however, to model these as affixes instead of as proclitics/ enclitics.
4. When you need to mark an allomorph with an allomorph property using the custom field of "ToneParsForm," the custom field only shows up in *FLEx* for an allomorph. It does not show for a lexeme form. To add an allomorph property to a lexeme form, you can use the "Swap Lexeme Form with Allomorph" capability (on the Lexeme Form item) or the "Swap Allomorph with Lexeme Form" capability (on the Allomorph item). See the *FLEx* help system for how to do this.

5. The current version of *FLEx* does not always parse a capitalized word. Here is one way to try and deal with this:[3]
   a. Go to the Baseline tab and change the upper-case letter to lower case.
   b. Return to the Analyze tab and parse the segment in *Use TonePars with FLEx*.
   c. If there are multiple parser-generated analyses available, select the correct one, then click on the green check mark. (This marks that particular analysis as "user-approved" in the Word Analyses area, which the interlinear view uses as one of its default sources.)
   d. Go back to the Baseline tab and change the letter back to upper case.
   e. Return to the Analyze tab, click on the word, then use the drop-down arrow on the Morphemes line to select the lower-case form of the word. (This should enable FLEx to associate the previously generated analysis of the lower-case form with the current instance.)
6. The user interface is in English only.
7. Be sure to close *Use TonePars with FLEx* *before* you close *FLEx* or there may a version of *FLEx* running in the background. This can prevent *FLEx* from starting again.
8. *Use TonePars with FLEx* produces a copy of the Tone Rule file with an extension of ".hvo" in the directory where the Tone Rule file is. This file is used by *TonePars* in order to correctly handle "morphname is" statements in the rule file.

6 Output files

When you parse a segment or a text, *Use TonePars with FLEx* produces several temporary files. There may be times when seeing these files will prove useful, especially if you are used to seeing such files in *CARLAStudio*. To see the files, open a Windows Explorer window in the temp directory. One way to do this is to click in the address bar and then replace its contents with "%TEMP%" (without the quotes). Next, view the directory showing details and click on the "Date modified" column header so that the most recently used files are ordered first. The files used by *Use TonePars with FLEx* are shown in example (3) below.

[3] This is a slightly modified version of Kevin Warfel's work-around in *FLEx*'s issue tracking system at LT-5722.

(3)

File name	Contents
*FLExProjectName*adctl.txt	The analysis data control file used by *XAmple*.
*FLExProjectName*gram.txt	The PC-PATR grammar file used by *XAmple*. (Most likely this file will not make much sense unless you have a lot of experience with PC-PATR grammar files.
*FLExProjectName*lex.txt	The lexicon file used by *XAmple*.
*FLExProjectName*TPadctl.txt	The analysis data control file used by *TonePars*.
*FLExProjectName*TPlex.txt	The lexicon file used by *TonePars*.
ToneParsCmd.cmd	A file containing the files loaded by *TonePars*.
ToneParsFLEx.bat	The batch file which invokes *TonePars*.
ToneParsInvoker.ana	The ANA file produced by *XAmple* which is used as the input to *TonePars*.
ToneParsInvoker.ant	The output ANA file produced by *TonePars*.
ToneParsInvoker.log	The log file produced by *TonePars*. See section 2.5 for an easier way to see this.

Note that most of these files use numbers instead of text for things like morph-names. This is how it needs to be done for *FLEx* to properly process the data.

7 Error messages

When parsing a segment or a text, you may see an error message box. The following tries to explain them.

Message	Explanation
Log file does not exist; please parse a segment or a text.	This appears when you try to show the log file and we could not find one.
Somehow the result file was empty. Please try again.	We detected that the resulting *TonePars* file was empty. This seems to be a result of some kind of timing issue.
There was a timing problem. Please try again.	We detected that the resulting *TonePars* file was not usable.

8 Support

If you have any questions with *Use TonePars with FLEx* or find bugs in it, please send an email to blackhandrew@gmail.com.