Technical Notes on Interlinear Import

March 22, 2017

Contents

This document describes the process of importing FieldWorks Interlinear XML or SFM data into FieldWorks Language Explorer version 7.2 or later. At this point it only imports baseline text, free translations, literal translations, notes, and some of the basic information in the Info tab. All fields except baseline can be in any number of languages. In future versions we expect to support full interlinear import.

Note: Because FLEx currently breaks segments at a period, exclamation point, question mark, or section sign (§), if your SFM file has any of these characters in the middle of a text segment, you'll get an extra segment without any translations, and the combined translations will come at the normal segment break in the file. If your segment ends without one of these punctuation marks, the import process will insert a section sign at the end of the segment to force FLEx to break the segment at this location, thus maintaining free translation alignment.

1 Importing SFM Interlinear text

The import process allows you to import multiple files at once, with each file containing one or more interlinear texts. When using multiple files, they all need to have consistent SFM markers. The interlinear import dialog lets you specify the language and an optional encoding converter for each field in your SFM file, along with how this field should be mapped to the interlinear model. You can also ignore any fields that are irrelevant.

Most interlinear SFM files can be imported into FLEx without any change to the file. There is some basic structure required for the fields to be imported.

Each text can have header fields including a title, abbreviation, source, or comment in any number of languages as long as each field and language is represented by a unique SFM marker. The body of the text comes after the header block. The body consists of optional paragraph breaks and segments. A segment begins with a reference marker followed by text blocks and an optional freeform block. A text block consists of aligned fields beginning with the baseline and then various other fields that are typically aligned using tabs or spaces under the baseline. The freeform block following the text blocks may include one free translation in any number of languages, one literal translation in any number of languages, and multiple notes in a single language.

There are two ways you can have multiple interlinear texts in a single file. You can use a special SFM marker to designate the beginning of each text, or FLEx will automatically start a new text if it encounters another header block after a body block. If your import includes multiple files, a new text will be started with each new file.

Here's a simple example of an interlinear SFM file.

```
\_sh v3.0  520  Text

\id Frog Meets Fish

\ref Fish.001
\tx Todn       lyfch    nyr
\mb tod  -n    lyfch    nyr
\ge frog -Nom  lily.pad by
\ps n1   -case n        postp
\tx velgoi.
\mb vel  -goi
\ge live -DPst
\ps v    -tns
\ft Long ago a frog lived by a lily pad.

\ref Fish.002
\tx Lyfch    tap   plap joygoi.
\mb lyfch    tap   plap joy  -goi
\ge lily.pad on    sit  like -DPst
\ps n        postp v    v    -tns

\ft He liked to sit on the lily pad.
```

In this file the \id field is an English title. The file has two segments starting with the \ref reference marker. If there is content following the reference marker, it will be stored in a special reference field in FieldWorks (Segment:Reference), but at this point in development it is not visible anywhere. The first segment has two text blocks while the second has a single text block. Each text block starts with the vernacular \tx field that will be mapped to the baseline. The additional fields (\mb, \ge, \ps) cannot be imported at this point in FLEx development, so they should all be set to "Don't Import". The freeform block simply consists of a \ft field in this case which represents the English free translation.

With this simple text, the fields would be mapped as follows in the interlinear dialog, assuming the vernacular language is French:

\id	Title	English
\ref	Reference	English
\tx	Baseline	French
\mb	Don't Import	
\ge	Don't Import	
\ps	Don't Import	
\ft	Free Translation	English

When specifying a field mapping, be sure you always select the correct writing system and encoding converter for the field. The dialog allows you to add new writing systems or converters without leaving the dialog. If an encoding converter is not specified, the field should be in Unicode UFT-8 encoding. If the field is not in Unicode, then you need to supply some converter to convert your data to Unicode during the import. One converter that is always available is

Windows 1252<>Unicode. This converter will convert normal ANSI text to Unicode. Tabs and CRLF are converted to spaces during import.

If a field occurs more than once consecutively, the contents will be concatenated into a single field in FieldWorks. As long as you have a unique marker, you can have any number of languages for all fields except the baseline.

\ti The house
\ti is red.
\tif La maison est rouge.
\tis La casa es rojo.

Assuming these mappings:

\ti	Title	English
\tif	Title	French
\tis	Title	Spanish

The result will be an English title, “The house is red.”, a French title, “La maison est rouge.”, and a Spanish title, “La casa es rojo.”

\ft The house is red.
\ftf La maison est rouge.
\nt This is a note about the house.
\note This is a note about the color red.

Assuming these mappings:

\ft	Free Translation	English
\ftf	Free Translation	French
\nt	Note	English
\note	Note	English

The result will be a free translation in English and French, and two English notes.

Note: Although the import process can import multiple languages, translations, and notes, you may not see them after the import until you use Tools...Configure...Interlinear to enable the fields you want to display.

The import dialog accepts one or more interlinear SFM file names, plus an XML mapping file that determines how each field is mapped. On the final page of the dialog you specify where you want the mapping file stored. By default, it will append -import-settings.map to the input file name and store the file in the same directory as the input file. When an input file is selected, if a matching map file is located in the same directory, it will use that mapping file. Otherwise it uses the last mapping file you used. The first time you import, it defaults to a default settings file that will work for many different SFM files. On the first page of the dialog, you can always revert to the default settings by clicking the blue “Use default settings file” link.

This table shows the default mappings that are pre-defined. You can override them as needed in the import dialog.

Code	Destination	Lang	Code	Destination	Lang
ab	Abbreviation		lx	Baseline	{vern}
abb	Abbreviation		mp-e	Source	en
au	Source		mp-s	Source	es
author	Source		name	New Text	
cmt	Note		nd	Note	
co	Comment		nt	Note	
com	Note		p	Paragraph	
comp	Source		po	Baseline	{vern}
compiler	Source		ref	Reference	
description-e	Comment	en	rf	Reference	
dt	Note		s	Baseline	{vern}
en	Note	en	sectn	Title	
et	Free Translation	en	sn	Note	es
etitle	Title	en	so	Source	
f	Free Translation		source	Source	
fe	Free Translation	en	st	Free Translation	es
filename	Title		t	Baseline	{vern}
fn	Note		te	Free Translation	en
fr	Free Translation		ti	Title	
fre	Free Translation		tit	Title	
ft	Free Translation		title	Title	
id	Title		title-e	Title	en
itm	Title		title-s	Title	es
l	Literal Translation		title-v	Title	{vern}
li	Literal Translation		tn	Reference	
lit	Literal Translation		tx	Baseline	{vern}
lt	Literal Translation		wn	Note	

Special intermediate file

By default, an input file is imported without writing intermediate files to the disk. In some special cases where you want to investigate the XML form being used for import, or when you may need to modify the XML in some way prior to import, there is a special mode that causes the process to write a FieldWorks Interlinear XML file instead of importing directly. On the final page of the dialog, if you hold Shift down before pressing Finish, FLEx will write a FieldWorks Interlinear XML file rather than importing. The XML file is written in the directory with the input file and will have -intermediate.xml appended to the input file name. If you include more than one file as input files, each file will be written to a separate intermediate XML file when the

Shift option is used. These XML files can be imported into FieldWorks using File...Import...FieldWorks Interlinear.

Constructing literal translations

The current import process cannot import word or morpheme glosses. However, if you have a simple interlinear SFM file with baseline, gloss, and free translations, you can use SIL Consistent Changes program with the CC table below to convert the glosses into a literal translation that can be imported into FieldWorks. This may be useful as you do actual glossing within FieldWorks. The literal translation for the segment will be inserted before each free translation.

For example, if you have this kind of input.

```
\ref KKFOOT unit 2
\tx Meke o     taki, a   mamanten leti  sigisi yuu
\mg make 1/2pl say   LOC morning  right six    hour
\tx ten.
\mg time
\ft Let's say, in the morning, right at six o'clock.
```

After running the CC table on the input it will look like this.

```
\ref KKFOOT unit 2
\tx Meke o     taki, a   mamanten leti  sigisi yuu
\tx ten.
\lt make 1/2pl say LOC morning right six hour time
\ft Let's say, in the morning, right at six o'clock.
```

This is the CC table that makes the changes

```
define(writeLiteral) > ifneq(gloss)  "
   begin
      '\lt ' out(gloss) nl store(gloss) endstore
   end

group(main)
'\mg  ' > '\mg ' back(4)
'\mg ' > append(gloss)
   ifneq(gloss) "
   begin
      ' ' back(2)
   end
   use(inGloss)
'\ft' > next
endfile > do(writeLiteral) dup

group(inGloss)
nl > endstore use(main)
11 > ' ' back
' ' > ' ' back
' ' nl > nl back
```

2 FieldWorks FLExText Interlinear XML

The FieldWorks Interlinear XML format used by FLEx is based on the Electronic Metastructure for Endangered Languages Data (EMELD) standard. This format can be exported from FieldWorks using File…Export Interlinear using the FieldWorks Interlinear XML option. The Export Interlinear option is not available for the Info and Baseline tabs. The export is controlled by settings under Tools…Configure…Interlinear. If you export from a FieldWorks project, you can import into any other FieldWorks project as long as the required writing systems (in the lang attributes) are present.

Although the export process will include desired information from the interlinear bundles, the import will only import baseline text, free translations, literal translations, notes, and some of the basic information in the Info tab.

Here are the steps for importing a FLExText Interlinear XML file into FieldWorks Language Explorer (FLEx).

1. In FLEx choose File…Import…FLExText Interlinear.
2. In the Interlinear Import dialog click the Browse button and select the FieldWorks Interlinear XML file to import.
3. Click OK to import your text.

If you have trouble seeing your text after the import it may be that your writing systems are not set up properly. Choose Tools…Configure…Interlinear to make sure that everything is enabled that you want to see.

Dealing with audio in interlinear

The audio writing system approach for linking to audio files is currently not available in interlinear Note fields (JIRA LT-18244).

To import audio links into interlinear text, mark the audio link in some way in the FLExText file ‘note’ element that you can later find and modify the file name and the text that gets the blue link, if different. Here’s an example:

```
<item type="note" lang="en">This is a note. An external audio
[AudioVisual\av23.mp3--link] in a note.</item>
```

Import the FLExText file into Flex, then close Flex.

The Note fields are stored similar to this in your project *.fwdata file.

```
<rt class="Note" guid="79862085-7768-45e2-b5b5-5d11760a0487"
ownerguid="6f0e5583-8b0f-4304-b9a7-c0a91dc2e393">
<Content>
<AStr ws="en">
<Run ws="en">This is a note. An external audio [AudioVisual\av23.mp3--link]
in a note.</Run>
</AStr>
</Content>
</rt>
```

Notice at this point the audio links are plain text. You'll need to provide some method for converting these links into appropriate externalLink runs. The above Note needs to be converted to this:

```
<rt class="Note" guid="83a4da03-f371-4495-b145-95de64b28202"
ownerguid="96141640-41f3-44c3-9d1c-ff31c872272b">
<Content>
<AStr ws="en">
<Run ws="en">This is a note. An external audio </Run>
<Run externalLink="AudioVisual\av23.mp3" namedStyle="Hyperlink"
ws="en">link</Run>
<Run ws="en"> in a note.</Run>
</AStr>
</Content>
</rt>
```

Note it now has an externalLink run that specifies the file name and the hyperlink text. The file names in this context would be relative to the LinkedFiles folder in your project. Here is a SIL Consistent Changes (CC) table that would make this change.

```
group(main)
'<rt class="Note"' > dup use(startNote)

group(startNote)
'/>' > dup use(main)  c skip blank notes.
'>' > dup use(inNote)

Group(inNote)
'</rt>' > dup use(main)
'<Run ws="' > dup store(ws) use(storeWs)

group(storeWs)
'"' > out(ws) dup use(inRun)

c This field needs to look for audio links and process them appropriately.
group(inRun)
'</Run>' > dup use(inNote)
'[' > '</Run>' nl '<Run externalLink="'
'--' > '" namedStyle="Hyperlink" ws="' out(ws) '">'
']' > '</Run>' nl '<Run ws="' out(ws) '">' use(inNote)
```

After converting the fwdata file, it should have live external links to your audio files when you open the file.