

Interlinear text implementation in FLEx

Ken Zook August 3, 2022

This document describes the technical aspects of interlinear text in FieldWorks Language Explorer (FLEx) in Version FW 9 and earlier versions after FW6, when FieldWorks switched to using an XML file for data storage instead of a SQL Server database.

Interlinear text is a complex model that integrates texts with a wordform inventory and with the lexical database. An interlinear display pulls bits and pieces from the lexicon, the wordform inventory, and some lists to form the final display.

In this document I show views from the lexicon and interlinear text, and show the underlying objects from the XML fwdata file. I only show relevant parts of the objects for this discussion. A more complete model diagram which has not been updated for quite a few years is version 7 ModelDocumentation.chm. A comprehensive list of current classes and fields are in MasterFieldWorksModel classes and fields 7000072.xlsx. Also, LCMBrowser.exe, installed in the FieldWorks 9 program directory gives several views of the current model and shows the data structure when you open a project file. For a more complete description of the fwdata XML structure, see http://downloads.sil.org/FieldWorks/Documentation/FieldWorks_7_XML_model.pdf. At the end I have a diagram showing the various connections between the classes for the simple sentence of interlinear text. This data can be seen in the Little French FW9.0.7 project.

Lexicon

Entries (LexEntry) in the lexicon can have any number of senses (LexSense). For interlinear purposes we only show the Lexeme Form or Allomorphs, Gloss, and Grammatical Info. We can show multiple languages/writing systems for each of these fields, although in this example I only show one writing system.

Lexeme Form (LexemeForm) and Allomorphs (AlternateForms) owns a MoForm subclass (usually MoStemAllomorph or MoAffixAllomorph). To display the actual lexeme form for an entry, you need to get the Form from the MoForm object. MoStemAllomorph or MoAffixAllomorph also have a MorphType field that references a MoMorphType object which may have a Prefix and/or Postfix marker. When present, these are added before and after the lexeme form. Finally, if the LexEntry has a HomographNumber, that is appended to the result to complete the display.

Grammatical Information is stored in a subclass of MoMorphSynAnalysis (usually MoStemMsa or MoInflAffMsa). Entries can store one or more instances of these in the MorphoSyntaxAnalyses property. Each sense references one of the MSAs owned by its entry. MSAs are considerably more complex than just a part of speech, but in this example, I limit it to the part of speech, which is a PartOfSpeech object owned in the Parts Of Speech CmPossibilityList. Possibility items contain both a Name and Abbreviation in one or more language/writing system. The lexicon data entry shows the name, and interlinear shows the abbreviation. Dictionary views can be configured to show either one.

Senses have many properties, but interlinear text only uses the grammatical information and the gloss. Note that although you can have multiple languages/writing systems in a gloss, a sense only has one gloss field. There have been discussions to add multiple translation equivalents in the future.

Note Lexeme Form and Gloss can have any number of writing systems, but they are simple strings without embeddings.

Entry		Show
garçon <i>n</i> young male child		
Lexeme Form	Fre	garçon
Morph Type		stem
Sense 1		
Gloss	Eng	boy
Definition	Eng	young male child
Grammatical Info.		Noun
Grammatical Info. Details		
Category Info.		Noun

The main entry above is represented by the following classes in the fwddata file.

```
<rt class="LexEntry" guid="13bf2994-ed83-40e7-a313-44334594f95b">
<LexemeForm>
<objsur guid="a66d0f19-4d33-48e6-b345-ce22c792f4a1" t="o" />
</LexemeForm>
<MorphoSyntaxAnalyses>
<objsur guid="bb673a3f-cbeb-4fc8-a071-8c3c70082685" t="o" />
</MorphoSyntaxAnalyses>
<Senses>
<objsur guid="77725752-bc2f-4c75-900f-db99dc258157" t="o" />
</Senses>
</rt>

<rt class="MoStemAllomorph" guid="a66d0f19-4d33-48e6-b345-ce22c792f4a1" ownerguid="13bf2994-
ed83-40e7-a313-44334594f95b">
<Form>
<AUni ws="fr">garçon</AUni>
</Form>
<MorphType>
<objsur guid="d7f713e8-e8cf-11d3-9764-00c04f186933" t="r" />
</MorphType>
</rt>

<rt class="MoStemMsa" guid="bb673a3f-cbeb-4fc8-a071-8c3c70082685" ownerguid="13bf2994-ed83-
40e7-a313-44334594f95b">
```

<PartOfSpeech>
<objsur guid="a8e41fd3-e343-4c7c-aa05-01ea3dd5cfb5" t="r" />
</PartOfSpeech>
</rt>

<rt class="LexSense" guid="77725752-bc2f-4c75-900f-db99dc258157" ownerguid="13bf2994-ed83-40e7-a313-44334594f95b">
<Definition>
<AStr ws="en">
<Run ws="en">young male child</Run>
</AStr>
</Definition>
<Gloss>
<AUni ws="en">boy</AUni>
</Gloss>
<MorphoSyntaxAnalysis>
<objsur guid="bb673a3f-cbeb-4fc8-a071-8c3c70082685" t="r" />
</MorphoSyntaxAnalysis>
</rt>

<rt class="MoMorphType" guid="d7f713e8-e8cf-11d3-9764-00c04f186933" ownerguid="d7f713d8-e8cf-11d3-9764-00c04f186933">
<Abbreviation>
<AUni ws="en">ubd stem</AUni>
</Abbreviation>
<Name>
<AUni ws="en">stem</AUni>
</Name>
</rt>

<rt class="PartOfSpeech" guid="a8e41fd3-e343-4c7c-aa05-01ea3dd5cfb5" ownerguid="f558e415-dad4-4c69-a595-c9915fea6b44">
<Abbreviation>
<AUni ws="en">n</AUni>
</Abbreviation>
<Name>
<AUni ws="en">Noun</AUni>
</Name>
</rt>

Entry <input type="checkbox"/> Show Hidden Fi	
-s <i>n sfx plural</i>	
Lexeme Form	Fre s
Morph Type	suffix
Sense 1	
Gloss	Eng PL
Definition	Eng plural
Grammatical Info.	n:Any
Grammatical Info. Details	
Category Info.	Affix which inflects Noun

The suffix entry above is represented by the following classes in the fwddata file.

```
<rt class="LexEntry" guid="052bec58-16d0-4fdb-9617-1ba148db23b5">
```

```
<LexemeForm>
```

```
<objsur guid="c2884d72-96f2-4a1b-a6e8-34932cfac0f7" t="o" />
```

```
</LexemeForm>
```

```
<MorphoSyntaxAnalyses>
```

```
<objsur guid="b879f06f-a612-4195-8b47-15c9b4eacadd" t="o" />
```

```
</MorphoSyntaxAnalyses>
```

```
<Senses>
```

```
<objsur guid="37cfcd18-136a-45e6-8dd6-15db22c774ac" t="o" />
```

```
</Senses>
```

```
</rt>
```

```
<rt class="MoAffixAllomorph" guid="c2884d72-96f2-4a1b-a6e8-34932cfac0f7" ownerguid="052bec58-16d0-4fdb-9617-1ba148db23b5">
```

```
<Form>
```

```
<AUni ws="fr">s</AUni>
```

```
</Form>
```

```
<MorphType>
```

```
<objsur guid="d7f713dd-e8cf-11d3-9764-00c04f186933" t="r" />
```

```
</MorphType>
```

```
</rt>
```

```
<rt class="MoInflAffMsa" guid="b879f06f-a612-4195-8b47-15c9b4eacadd" ownerguid="052bec58-16d0-4fdb-9617-1ba148db23b5">
```

```
<PartOfSpeech>
```

```
<objsur guid="a8e41fd3-e343-4c7c-aa05-01ea3dd5cfb5" t="r" />
```

```
</PartOfSpeech>
```

```
</rt>
```

```

<rt class="LexSense" guid="37cfd18-136a-45e6-8dd6-15db22c774ac" ownerguid="052bec58-16d0-4fdb-9617-1ba148db23b5">
<Gloss>
<AUni ws="en">PL</AUni>
</Gloss>
<MorphoSyntaxAnalysis>
<objsur guid="b879f06f-a612-4195-8b47-15c9b4eacadd" t="r" />
</MorphoSyntaxAnalysis>
</rt>

```

```

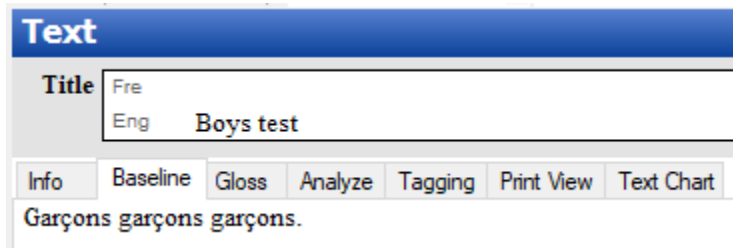
<rt class="MoMorphType" guid="d7f713dd-e8cf-11d3-9764-00c04f186933" ownerguid="d7f713d8-e8cf-11d3-9764-00c04f186933">
<Abbreviation>
<AUni ws="en">sfx</AUni>
</Abbreviation>
<Name>
<AUni ws="en">suffix</AUni>
</Name>
<Prefix>
<Uni></Uni>
</Prefix>
</rt>

```

Interlinear Text Baseline

FLEx allows any number of Text objects for displaying interlinear text. A Text has a Name in any number of writing systems, and owns a single StText object which holds any number of StTxtPara objects. StTxtPara has a Contents property which holds a FieldWorks string which can have embeddings. Only one writing system is used for the baseline. Any embedded writing systems will not allow interlinearization and will be lumped with punctuation in PunctuationForm objects.

The Baseline display simply prints the Contents of each StTxtPara with a paragraph break between each one.



The baseline for this interlinear text is represented by the following classes in the fwdata file.

```

<rt class="Text" guid="754a048e-48a1-4b4e-92de-20897834f630">
<Contents>
<objsur guid="e1461362-6de5-42c9-a644-ac0f52c01d79" t="o" />
</Contents>

```

```

<Name>
<AUni ws="en">Boys test</AUni>
</Name>
</rt>

<rt class="StText" guid="e1461362-6de5-42c9-a644-ac0f52c01d79" ownerguid="754a048e-48a1-4b4e-92de-20897834f630">
<Paragraphs>
<objsur guid="9007af3e-0ff3-46c1-812a-d68704968f43" t="o" />
</Paragraphs>
</rt>

<rt class="StTxtPara" guid="9007af3e-0ff3-46c1-812a-d68704968f43" ownerguid="e1461362-6de5-42c9-a644-ac0f52c01d79">
<Contents>
<Str>
<Run ws="fr">Garçons garçons garçons.</Run>
</Str>
</Contents>
<Segments>
<objsur guid="cd4f5759-5cdc-450f-9094-884136d8fe51" t="o" />
</Segments>
</rt>

```

Interlinear Text Analysis

Interlinear texts show information from the baseline, from a wordform inventory, and from the lexicon.

In addition to Contents, a StTxtPara also owns a sequence of Segment in the Segments property. The baseline is automatically broken into segments based on period, question mark, exclamation point, and section mark. So normally a segment represents one sentence in the baseline.

A Segment has a BeginOffset value to show where the segment starts in the StTxtPara Contents string. It holds a sequence of pointers in the Analyses property. These pointers indicate the level of interlinearization that has been approved by the user. The example shown here has the same word which is analyzed to different levels. Default analyses are shown in blue until the user confirms the analysis.

The first word has not had any analysis, so it shows the proposed analysis that comes from previous analyses. The only part that is confirmed here is the WfiWordform which happens automatically when a baseline is modified. The Analyses pointer for this unanalyzed state is a pointer to the WfiWordform.

The second word has had lexical analysis approved, but no word glosses. At this point the user has broken the word into morphemes, and each morpheme is connected to a sense in the lexicon. The Analyses pointer for this word is to a WfiAnalysis owned by the WfiWordform.

The third word is fully analyzed and includes a Word gloss and Word category. The Analyses pointer for this word is to the WfiGloss object owned by the WfiAnalysis. All words in the interlinear text baseline are assigned to one of these levels.

Any punctuation or embedded writing systems are represented by a pointer to a PunctuationForm. The Form of this PunctuationForm holds a string representing a single punctuation mark, a cluster of punctuation marks, or text in an embedded writing system.

Text				
Title		Fre		
Eng		Boys test		
Info	Baseline	Gloss	Analyze	Tagging
1 Word	garçons	garçons	garçons	garçons
Morphemes	garçon -s	garçon -s	garçon -s	garçon -s
Lex. Entries	garçon -s	garçon -s	garçon -s	garçon -s
Lex. Gloss	boy PL	boy PL	boy PL	boy PL
Lex. Gram. Info.	n n:Any	n n:Any	n n:Any	n n:Any
Word Gloss	boys	boys	boys	boys
Word Cat.	n	n	n	n
Free Boys boys boys.				

The interlinearization for this interlinear text is represented by the following classes in the fwdata file.

```
<rt class="Segment" guid="cd4f5759-5cdc-450f-9094-884136d8fe51" ownerguid="9007af3e-0ff3-46c1-812a-d68704968f43">
<Analyses>
<objsur guid="1dd8cf5c-beb0-4463-9b51-9b98e47b7879" t="r" />
<objsur guid="ccaad034-ad7f-4d16-a547-9cd8ca845365" t="r" />
<objsur guid="13f9d3fa-70be-473b-ac2a-4ee1d215d5be" t="r" />
<objsur guid="a5ec72fa-99d8-49a0-bca3-ec29a52db3fc" t="r" />
</Analyses>
<BeginOffset val="0" />
<FreeTranslation>
<AStr ws="en">
<Run ws="en">Boys boys boys.</Run>
</AStr>
</FreeTranslation>
</rt>
```

The punctuation and any text that is not in the primary writing system in the baseline is represented by the following class in the fwdata file, which covers the period at the end of the sentence.

```
<rt class="PunctuationForm" guid="a5ec72fa-99d8-49a0-bca3-ec29a52db3fc">
<Form>
<Str>
```

```
<Run ws="fr">|</Run>
</Str>
</Form>
</rt>
```

Wordform Inventory

Text in the main writing system of the baseline is broken up into wordforms. The wordforms are stored in the wordform inventory in WfiWordform objects, and in the Analyses section of the Segment there are pointers to the WfiWordform until they are analyzed further.

During analysis, the user can select to use the entire word, or they can break it up into morphemes. This information is stored in a WfiAnalysis object owned by the WfiWordform. The next step is to connect each morpheme to a sense in a lexical entry. The user is able to create new entries, allomorphs, or senses in this phase.

For each morpheme, a WfiMorphBundle is added to the WfiAnalysis in the MorphBundles property. When first created, the form the user specified for the morpheme is stored in the Form of the WfiMorphBundle. As the user creates and/or selects an entry, the Morph pointer is set to the MoForm of the LexemeForm or AlternateForms of the LexEntry. When a sense is chosen from that entry, the Sense property is set to the LexSense, and the Msa pointer is set to the MSA pointed to from the MorphoSyntaxAnalysis reference property on Sense.

If a user specifies a different set of morphemes, or different analyses, a new WfiAnalysis is created and filled in as above.

When the user fully analyzes the word, they can set the Category on WfiAnalysis to a PartOfSpeech for the entire word. They can also add a word gloss. The word gloss goes into a WfiGloss object owned by the WfiAnalysis in the Meanings property. Each WfiGloss can have a Form in multiple writing systems. It's possible to have more than one word gloss, so WfiAnalysis owns a sequence of WfiGloss objects. During full analysis the user can choose which gloss they want for this particular instance of a wordform. This is why the Analyses pointer goes to a WfiGloss at the fully analyzed state.

There are two parsers built into FLEx that can do automatic parsing. These parsers will use the lexicon and grammar to go through wordforms and create possible analyses including the Morph and Msa pointers. It then flags the analysis as a parser approved or disapproved parse.

Wordform Analyses

garçons [Assign Analysis...](#)

Spelling Status

User Approved (Analyses)

Analysis 1

Analysis	Morphemes	garçon	-s
	Lex. Entries	garçon	-s
	Lex. Gloss	boy	PL
	Lex. Gram. Info.	n	n:Any

Parse result

Word Category

Word Gloss Eng boys

The wordform and analysis of garçons is represented by the following classes in the fwddata file.

```
<rt class="WfiWordform" guid="1dd8cf5c-beb0-4463-9b51-9b98e47b7879">
<Analyses>
<objsur guid="ccaad034-ad7f-4d16-a547-9cd8ca845365" t="o" />
</Analyses>
<Form>
<AUni ws="fr">garçons</AUni>
</Form>
<SpellingStatus val="1" />
</rt>

<rt class="WfiAnalysis" guid="ccaad034-ad7f-4d16-a547-9cd8ca845365" ownerguid="1dd8cf5c-beb0-4463-9b51-9b98e47b7879">
<Category>
<objsur guid="a8e41fd3-e343-4c7c-aa05-01ea3dd5cfb5" t="r" />
</Category>
<Evaluations>
<objsur guid="8caa11bb-cac4-4836-a081-1666245106b9" t="r" />
</Evaluations>
<Meanings>
<objsur guid="13f9d3fa-70be-473b-ac2a-4ee1d215d5be" t="o" />
</Meanings>
<MorphBundles>
<objsur guid="9ef5cf19-f055-4df8-9b32-bee07367d8a3" t="o" />
<objsur guid="70550026-e174-4bee-98c5-914ee31a5c11" t="o" />
</MorphBundles>
</rt>
```

```
<rt class="WfiGloss" guid="13f9d3fa-70be-473b-ac2a-4ee1d215d5be" ownerguid="ccaad034-ad7f-4d16-a547-9cd8ca845365">
```

```
<Form>
```

```
<AUni ws="en">boys</AUni>
```

```
</Form>
```

```
</rt>
```

```
<rt class="WfiMorphBundle" guid="9ef5cf19-f055-4df8-9b32-bee07367d8a3" ownerguid="ccaad034-ad7f-4d16-a547-9cd8ca845365">
```

```
<Form>
```

```
<AStr ws="fr">
```

```
<Run ws="fr">garçon</Run>
```

```
</AStr>
```

```
</Form>
```

```
<Morph>
```

```
<objsur guid="a66d0f19-4d33-48e6-b345-ce22c792f4a1" t="r" />
```

```
</Morph>
```

```
<Msa>
```

```
<objsur guid="bb673a3f-cbeb-4fc8-a071-8c3c70082685" t="r" />
```

```
</Msa>
```

```
<Sense>
```

```
<objsur guid="77725752-bc2f-4c75-900f-db99dc258157" t="r" />
```

```
</Sense>
```

```
</rt>
```

```
<rt class="WfiMorphBundle" guid="70550026-e174-4bee-98c5-914ee31a5c11" ownerguid="ccaad034-ad7f-4d16-a547-9cd8ca845365">
```

```
<Form>
```

```
<AStr ws="fr">
```

```
<Run ws="fr">s</Run>
```

```
</AStr>
```

```
</Form>
```

```
<Morph>
```

```
<objsur guid="c2884d72-96f2-4a1b-a6e8-34932cfac0f7" t="r" />
```

```
</Morph>
```

```
<Msa>
```

```
<objsur guid="b879f06f-a612-4195-8b47-15c9b4eacadd" t="r" />
```

```
</Msa>
```

```
<Sense>
```

```
<objsur guid="37cfcd18-136a-45e6-8dd6-15db22c774ac" t="r" />
```

```
</Sense>
```

```
</rt>
```

The FLEx LangProject currently has four CmAgents owned in AnalyzingAgents. Current agents are M3Parser, HCPParser, Computer, and default user. Each CmAgent owns 2 CmAgentEvaluations; one in an Approves property and one in the Disapproves property. Here's the default user agent.

```

<rt class="CmAgent" guid="9303883a-ad5c-4ccf-97a5-4add391f8dcb" ownerguid="e6c446c7-bd0b-42ab-a4d4-776d3ae7876f">
<Approves>
<objsur guid="8caa11bb-cac4-4836-a081-1666245106b9" t="o" />
</Approves>
<Human val="True" />
<Name>
<AUni ws="en">default user</AUni>
</Name>
</rt>

<rt class="CmAgentEvaluation" guid="8caa11bb-cac4-4836-a081-1666245106b9"
ownerguid="9303883a-ad5c-4ccf-97a5-4add391f8dcb" />

```

WfiAnalysis refers to one or more CmAgentEvaluations in an Evaluations property. Thus, when a user approves an analysis, the Evaluations property references the approved CmAgentEvaluation for default user. This normally happens in the Interlinear Analyze view when a user approves an analysis. A user can also change the user analysis status by right-clicking the analysis in Wordform Analyses and selecting the User Option from Approve, Unknown, or Disapprove.

- When Unknown is selected, the analysis does not have any CmAgentEvaluation references to the default user CmAgent and the analysis shows under “User Opinion Unknown (Analysis Candidates)” section of the Wordform Analysis view.
- When Approve is selected, The approved CmAgentEvaluation for default user is added to the WfiAnalysis Evaluations property and the analysis shows under the “User Approved (Analyses)” section of the Wordform Analysis view.
- When Disapprove is selected, The disapproved CmAgentEvaluation for default user is added to the WfiAnalysis Evaluations property and the analysis shows under the “User Disapproved (Test Case Analyses)” section of the Wordform Analysis view.

When a parser approves an analysis, the analysis Evaluations property references the Approves CmAgentEvaluation for the parser CmAgent. This status is shown in the Parse result field in the Wordform Analyses as “Successful”. When the parser disapproves an analysis, the analysis Evaluations property references the Disapproves CmAgentEvaluation for the parser CmAgent. This status is shown in the Parse result field in the word analyses as “Failure”. When analysis does not have any references to the parser CmAgentEvaluation for the parser CmAgent, the Parse result field shows “Untested”.

The diagram on the next page lists all of the instances of classes that are used to complete this interlinear example including the lexicon, wordform inventory, and text classes. The numbers are the first part of the guids from the XML classes above.

The actual display of interlinear text is controlled through Tools...Configure...Interlinear. This determine which fields are being shown and which writing systems to use for each one.

FLEx Interlinear Text Instance Diagram

