**SFM import of variants and subentries**

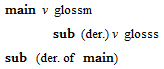
Ken Zook  
June 17, 2016

Understanding how the current (FW8.2.8) SFM import works with variants and subentries is very important to get desired results. There are numerous limitations with the current import that require special strategies (e.g. preprocessing) in order to have a successful import. Let’s look at various examples using root base view to see how the import process works at this point.

# Subentries

Example 1

\lx main  
\ps v  
\ge glossm  
\se sub  
\ps v  
\ge glosss

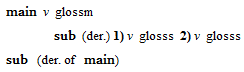


During the import process when we encounter a subentry (\se) we automatically create a second entry and link it as a subentry of the entry in which the \se occurs. So the default view will show it as a subentry of the main entry and will also show a minor entry pointing to the main entry. If you don’t want to include the minor entry for a given entry, you can uncheck the Show Minor Entry field in the minor entry.

Example 2

But often the SFM dictionary will already have a minor entry like this:

\lx main  
\ps v  
\ge glossm  
\se sub  
\ps v  
\ge glosss  
  
\lx sub  
\mn main  
\ps v  
\ge gloss



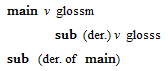
What happened during the import? Just like the first example, we process the \se and related fields as a new entry that is a subentry of the main entry. But as a result we have two ‘sub’ entries; the one we created for the \se, in addition to the minor entry that was already there. But we don’t want to leave the user with a bunch of extra entries in this process, so after the data is imported, we go through the results and when we find an existing minor entry, for the one we created, it combines the two into a single entry. It does this right now by appending the sense we created to the existing sense of the minor entry. Unfortunately, our current merge process doesn’t detect that the two senses are really identical. In this case is should have only had one sense. Appending a generated sense that is different from the existing would make sense.

For the merge to work properly between the subentry and the minor entry, the subentry \se field must be identical to the \lx field of the minor entry. Also, the minor entry must have a \mn field that matches the \lx field off the main entry. As long as these two reciprocal links are present, then Flex will recognize the two entries as equivalent and will merge the results.

Example 3

There are three ways you can work around the existing merge limitation that results in two identical senses. The first is to store the sense content in the minor entry and remove the sense fields following the \se.

\lx main  
\ps v  
\ge gloss  
\se sub  
  
\lx sub  
\mn main  
\ps v  
\ge glosss



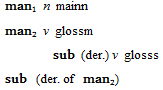
The result here is what we want. A second way to get the same result is to put all of the sense content following the \se and remove sense content from the minor entry.

\lx main  
\ps v  
\ge glossm  
\se sub  
\ps v  
\ge glosss  
  
\lx sub  
\mn main

A third way, would be to visit each minor entry after the import and use the Flex merge sense menu option.

Example 4

\lx main  
\hm 1  
\lc man  
\ps n  
\ge mainn  
  
\lx main  
\hm 2  
\lc man  
\ps v  
\ge glossm  
\se sub  
\ps v  
\ge glosss  
  
\lx sub  
\mn man2



Notice that the \mn field in the minor entry needs to specify the main entry correctly for the merge to work. First, if there is a citation form on the main entry, the \mn field should specify the citation form since that is what gets displayed in the final result. Also, when there are homographs, the homograph number should be specified as an appended number in the \mn field.

The above examples are all clean data which result in good imports. Without clean data the import will be a mess. You may see things like this:

\lx bre2  
\lx bre / bree / brap, etc.  
\lx ' , or q, glottal consonant  
\lx aban , abin (PC)  
\se housesit, see note in appendix  
\se 0

Where homograph numbers are appended to the \lx instead of using a \hm, more than one form may be included in the same field, or other commends and glosses. When you don’t have proper \lx \se \mn fields, Flex will do what it can, but the results will not be pretty.

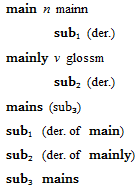
If a person has more than one subentry in a single \se field  
\se abarn, aban  
it would look for a \lx field with the same words. In most cases, that will not be found. Instead, the input needs to have each form in a separate \se field  
\se abarn  
\se aban

There are other issues where people have an \se in an entry with the same form as the \lx field that will cause problems.

Example 5

If there are multiple \se fields with the same form will not attempt to merge the results, so it creates a new minor entries with homographs . Also, if a \mn field refers to something that doesn’t exist, we create an entry as a main entry for the subentry. When this happens it defaults to a variant.

\lx main  
\ps n  
\ge mainn  
\se sub  
  
\lx mainly  
\ps v  
\ge glossm  
\se sub  
  
\lx sub  
\mn mains



# Variants

Variants work much like subentries. Here’s a simple example.

Example 6

\lx main  
\va var  
\ps n  
\ge main



When Flex encounters a \va field, it creates a variant entry that points back to the main entry.

Example 7

\lx main  
\va var  
\ps n  
\ge mainn  
  
\lx var  
\mn main  
\ps v  
\ge glossv



As with subentries, if there is a minor entry already present for the variant, in the process Flex creates a second variant entry, but then in the cleanup after the import, it recognizes that the variant entry already exists, so it merges the results into a single variant minor entry. The merge process depends on the \va field matching the minor \lx field, and the \mn field matching the \lx of the main entry. As with subentries.

Variants of senses

Multiple variants on one line

A \mn by itself

Subentries of senses.

Example from Agutaynen agsa entry with subentries on sense.

Agta example where sub and minor entries are different and how that affects us.