

# WeSay Roadmap

2/14/2008

## Installer

- ✓ Install sample data so evaluators can try it out fast. <sup>1</sup>
- ② Update old version without requiring uninstalling first.
- ② Allow user to choose an installation location other than “Program Files”.

## WeSay Configuration Application

- ✓ Create a new project
- ✓ Launch a project in WeSay

## Task Setup

- ✓ Control which tasks to enable for the user at this time
- ✓ Provide several ready-to-use tasks: “Add Meanings”, “Add Examples”, “Collect Word List”, “Collect Words By Semantic Domains”.
- ✓ Control which fields are shown for each task.
- ① Support configuring tasks without having to edit xml. E.g., choosing the wordlist and wordlist writing system in the Collect Wordlist Task.

## Field Setup

- ✓ Control which writing systems are shown for each field, and in what order.
- ✓ Allow user to create custom fields in the Setup App. Until then create by editing the tasks.xml file directly.

---

<sup>1</sup> Key

✓ Done

① Planned for first version

~① Planned for first version, partially implemented

② Planned for second release

(no mark) Imagined, but unplanned

## Writing Systems

---

- ✓ Specify a font for the writing system
- ✓ Mark if the writing system uses right-to-left text.
- ✓ Choose a keyboard to use when in fields of this writing system. The keyboard can be either a system one or [Keyman](#).
- ✓ Support Keyman 7 [[WS-390](#)], [[WS-427](#)].
- ② Support alternative keyboard on Linux, such as [kmfl](#).
- ✓ Support complex non-roman fonts on Windows
- ② Support complex non-roman fonts on Linux
- ✓ Allow multiple writing systems per field.
- ✓ Specify which writing systems appear in which tasks.
- ✓ Changing the writing system id automatically updates the lexicon LIFT xml.
- ✓ Specify an on-screen abbreviation for the writing system, separately from the writing system id.
- ✓ Specify a language which Windows knows about in order to sort like that language.
- ① Specify which writing system to sort on, in the tool and for export. Currently just uses the first ws it finds in the first lexeme form. [[WS-532](#)]
- ② Specify transducers (e.g. SIL Converters) to automatically create sort keys or convert to other writing systems.
- ② Allow sorting on a different writing system than the display one. This will allow sorting on transducer-created sort keys.
- ② Store the information for a writing system in its own xml file, conforming to [LDML](#) with some SIL enhancements: default font, keyboards. Allow reuse of this data across applications by storing a copy in a common location.

## Option Lists

---

- ✓ Allow custom fields to be limited to a list of options.
- ✓ Edit option lists in Configuration Tool (e.g. parts of speech).

## WeSay

- ✓ Localization via [poedit](#) or equivalent. Doesn't require submitting anything back to the developers (though you should so we can send it out with WeSay). Localization via [www.launchpad.net](#) .
- ✓ Provide Dictionary Services to other applications:
  - ✓ Lookup words
  - ✓ Add word with definition
  - ① Add variant and lookup using variant
- ② Provide spell-checking with red squiggles

## Home Tab

---

- ✓ Tell user how many word they have so far
- ② Show histogram indicating progress over time
- ✓ List all the tasks the advisor has enabled using the Setup application.
- ✓ At the top, show and highlight the task the user was in previously, to remind him where he was.
- ✓ Next to each task show how many words/senses still need that task.
- ✓ Right-clicking enables quickly opening the project in the Setup application.

## Lexical Model (what's in the database)

---

- ✓ Supported *objects* are Word, Sense, and Example
- ✓ Each object can have unlimited *custom fields*. Each is one of the following types:
- ✓ A field can contain one or more strings, one per writing system.

Here, a *string* is a list of Unicode characters in a single writing system.

✓ An option Reference (LIFT' s trait) is a value restricted by an Option List (aka Range Set).

✓ An Option Reference Collection (a sequence of LIFT' s traits) is an unordered collection of values restricted by an Option List (aka Range Set).

~① Support LIFT's *relation* object. Currently, you can add any number of relations, but only of the type that points to a single entry (e.g. BaseForm).

Support LIFT's *variant* object.

Support LIFT's *reversal* object.

Support for strings which contain styled information (LIFT's *span*).

- ✓ Words or phrases, in one or more writing systems.
- ✓ Words can have one or more Senses
- ② Words can have one or more sound recordings.
- ✓ Possible to Produce Subentries. Use the BaseForm relation field in each subentry.
- ✓ Senses can have one or more examples
- ✓ Senses can have a single picture.
- ✓ Examples have a single translation (though it can be in multiple writing systems).
- ✓ Option Lists (aka Range Sets)
- ✓ Option Lists can limit single-item traits, like Part Of Speech, and multiple item trait collections, like Semantic Domains.
- ✓ Options have an underlying *key*, and but can be displayed in a chosen language.
- ✓ Options with a *key* that cannot be found in the Option List display the raw key with a red background.

### Missing Fields Tasks

---

- ✓ Show list of words that are missing the field
- ✓ Move words to the “completed” list when the field is filled in
- ✓ Show other fields for context, but read-only

### Dictionary Browse and Edit nTask

---

- ✓ Add New Word
- ✓ Delete This Word
- ✓ Show list of all words down the side.
- ✓ Change the writing system of the list via a drop-down above it, so that you can view, for example, all the English glosses, or all the words in a particular writing system.
- ✓ Type-ahead search box.
- ✓ Search box uses approximate matching to find word even without knowing the exact spelling.
- ✓ Show dictionary output preview of this word, at the top.
- ✓ Underline the part of the dictionary output corresponding to the field which has the cursor.

- ✓ Always provide a blank line to add more information, such as another meaning (sense), or another example sentence. (*This is under review... does it make the program seem like it is never satisfied?*)
- ✓ Delete a sense or example by simply erasing its contents.
- ✓ Provide a star button for indicating something special about a field, such as the need to review it later.
- ✓ Add pictures to senses. The picture is copied to the “pictures” folder of the WeSay project and shown in the dictionary task.
  
- ✓ Edit collection of semantic domains assigned to a sense. Find domains by typing the domain number, a word in the domain title, or a word in the description of the domain.
- ② Support editing of option entry references (e.g. subentries).
- Support setting and maintaining homograph numbers.
- Support reordering senses.

### **Collect Word List**

---

- ✓ Collect words in the target language by displaying one word at a time in a foreign language, and asking the user to translate it into one or more words in his language.
- ✓ Words are entered into the dictionary, with the foreign word listed as a gloss.
- ✓ If the word is already in the dictionary, a new sense is created, with the foreign word as its gloss.
- ✓ Addition of Words animated to reinforce what’s going on.

### **Semantic Domain Task**

---

- ✓ Addition of Words animated to reinforce what’s going on.

### **Other imagined Tasks**

---

- ②Review. View starred items and items that fail checks.
- Split/Merge words/ homographs
- Sense disambiguation
- Record pronunciations
- Record Stories
- Transcribe Stories

Collect words based on a foreign dictionary. Would import selected fields into each vernacular word that is elicited. For example, if the source dictionary entry has semantic domains, copy those into the newly created entry.

## Actions

---

- ✓ Provide an easy way for other programmer to add plug-ins which do some action.
- ✓ Export to Standard Format.
- ✓ One-click zip and backup to USB Flash Drive
- ✓ Email advisor.
- ~① Print draft dictionary. (In Oct 2007 doesn't output pictures)
- ② Print progress report.
- ② Synchronize with version control repository.
- ② Open dictionary in Lexique Pro (can be done in version 1 by opening the exported Standard Format document).

## Notes

Since WeSay seeks to shield the user from the file system, there is no "Open..." command. WeSay can be started with a particular project in these ways:

By double clicking on the LIFT file

Via Project:"Open this project in Wesay" in the WeSay Setup program

Via a shortcut or command-line parameter which gives the path to the LIFT file

By running WeSay, in which case it opens the last project it was run with.

The setup program places new project in the "My Documents" folder, but you can move them anywhere.

## Keyboard Shortcuts

### Standard Windows ones

---

- ✓ Tab: move to next field
- ✓ Shift-Tab: move to previous field
- ✓ Ctrl+PageDown Move to Next Tab Page
- ✓ Ctrl+PageUp Move to Previous Tab Page
- ✓ Alt+Down Open up a combo-box (e.g. Part of Speech)

## Dictionary Task

---

- ✓ Alt+N: New Word
- ✓ Alt+D: Delete Word
- ① Ctrl+F: Move to find box

## Collect By Semantic Domain

---

- ✓ Enter: Add word to list of words with the displayed semantic domain
- ✓ Page Down: Move to the next question or, if out of questions, to the next topic.

## Collect By Semantic Domain

---

- ✓ Enter: Add word to list of words which are equivalent to the displayed foreign word.
- ✓ Page Down: Move to the next word to translate.

## Missing field tasks (e.g. Add Meanings)

---

- ✓ Page Down: Move to the next word which is missing this field

## LIFT XML File format support

LIFT, by its nature, is not a format that every application is expected to fully support. Since WeSay is aimed at basic dictionary making and not full-on lexicography, we don't aim to support editing of every kind of data LIFT supports.

- ✓ When opening a new LIFT file, preprocess it to add guids (Globally Unique Identifiers) to every entry.
- ✓ Drop unsupported data in the LIFT file.
- ② Allow opening of LIFT files that are not inside a WeSay Project directory. Create a default task/field configuration for them.
- ② Allow unsupported LIFT data to be retained in the LIFT file.

See <http://www.wesay.org/wiki/LIFT> for a description of our progress in supporting LIFT.