# Installer

## ✔Install sample data so evaluators can try it out fast.

## Update old version without requiring uninstalling first.

## Allow user to choose an installation location other than “Program Files”.

# WeSay Setup Application

## Create a new project

## Launch a project in WeSay

## Task Setup

### ✔Control which tasks to enable for the user at this time

### Control which fields are shown for each task. Until then can be tweaked by hand in the tasks.xml file.

## 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.

## Writing Systems

### ✔Specify a font for the writing system

### ✔Mark if the writing system uses right-to-left text.

### ✔Enter the name of a keyboard to use when in fields of this writing system. The keyboard can be either a system one or [Keyman](http://www.tavultesoft.com/keyman/).

### Support alternative keyboard on Linux, such as [kmfl](http://kmfl.sourceforge.net/).

### ✔Allow multiple writing systems per field.

### ✔Specify which writing systems appear in which tasks.

### ✔Changing the writing system id automatically updates the lexicon xml.

### Specify an on-screen abbreviation for the writing system, separately from the writing system id.

### Specify sort order other than standard Unicode order.

### Specify a language which Windows knows about in order to sort like that language.

### Use a library like [ICU](http://www-306.ibm.com/software/globalization/icu/index.jsp) to expand the number of build-in sorting systems.

### Provide a way to define custom sorting systems.

### Store the information for a writing system in its own xml file, conforming to [LDML](http://www.openi18n.org/specs/ldml/1.0/ldml-spec.htm) with some SIL enhancements: default font, keyboards. Allow reuse of this data across applications by sharing these files.

## Option Lists

### ✔Allow custom fields to be limited to a list of options.

### Edit option lists in separate program (e.g. parts of speech). Currently xml-only.

# WeSay

## Missing Fields Tasks

### Show other fields for context, but read-only

## Semantic Domain Task

### Support editing of collection traits (e.g. Semantic Domains) in the Dictionary view.

# 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

### Ctrl+F: Move to find fox

## 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