Solid Documentation

Solid Documentation

Table of Contents

Introduction

About This Document

Who Should Read This

This document is intended primary for field linguists who want to validate and/or clean up their Toolbox (SFM) dictionaries, whether in preparation for publication or for migration to another program such as Flex. The actual publication/migration process should be handled by an expert.

You can set up Solid yourself, or perhaps a Language Technology consultant has set it up for you and asked you to run it periodically to keep your lexicon clean and consistent. Little technical knowledge is required to for that purpose, and there are not many technical details in this manual.

For technical details regarding specific goals--that is, what rules your SFM file should end up conforming to, please see Technical Notes on SFM Import (in FLEx's Help menu). The documentation of MDF is also helpful, as are the wiki pages at lingtransoft.info :

http://tiki.lingtransoft.info/Preparing+Legacy+data+for+Flex?structure=Navmenu

http://wiki.lingtransoft.info/doku.php?id=tutorials:toolbox

What You Will Learn

This document covers how to install Solid, open your dictionary, make corrections, and save or export your dictionary to LIFT. The sections are roughly in procedural order, but can be read individually.

Document Conventions

In this document you will find three basic conventions that make important notes stand out. There are:

Info signs, which are for clarification or expanding on the topic;

Lightbulbs, which indicate tips and extra information;

and Warning signs, which are for areas of great consequence.

Also, if you come across a set of words in italics, separated by a series of | marks, this indicates a series of steps you need to go through. For instance, *My Documents* | *My Music* means you need to open My Documents and then open the My Music folder.

The terms 'record' and 'entry' are used interchangeably in this document. In an SFM file that represents a lexicon (a dictionary), each record/entry begins with a record marker (\lx). Any subentries (\se) are still part of the same record/entry.

The term 'marker' or 'field marker' technically refers to only the backslash code (e.g. \lx), but "moving a marker" should be understood to refer to the whole field including the field content (e.g. \lx someword). Note that most fields are a single line of text, but some field such as long, hard-wrapped definitions may span multiple lines of text in the file.

Software License

This program is made available under the MIT License shown below (also available at http://opensource.org/licenses/MIT). Note that Solid itself is open source, but it makes use of compiled components that are not, including:

- Glacial ListView - Copyright Glacial Components Software 2004 - http://www.glacialcomponents.com/

Installation

Downloading Solid

Before you can start using Solid you need to install it on your computer. You can download the installer off the web from http://solid.palaso.org. There is a link on that page for downloading the latest version of Solid.

On the page surrounding the link you can find information about Solid, including a list of features, and links to websites mapping the development of Solid.

The process is pretty self-explanatory, and you will not be asked to configure the installation itself, except to indicate whether or not to launch Solid after installing. For details and screenshots of the installation process, please see Appendix A.

A Tour of Solid

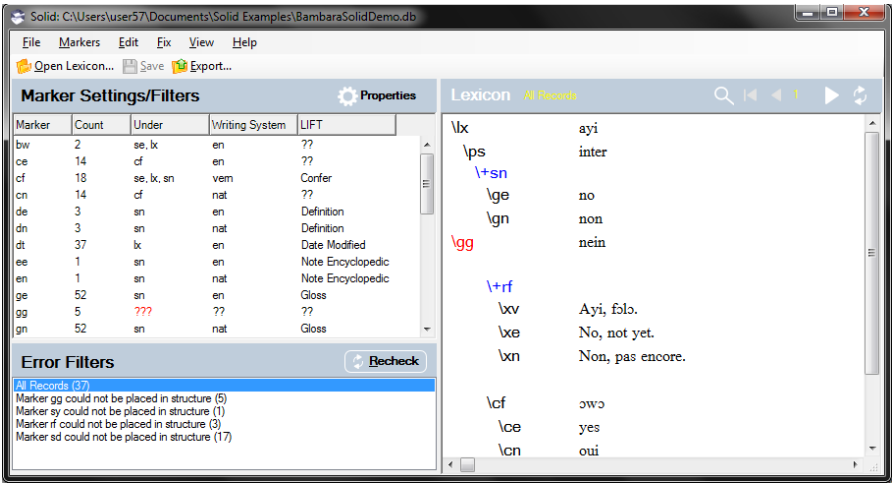
Quick Start

The following sections describe Solid in action. If you want to follow along with your own lexicon, first close any other editor such as Toolbox that may have it open, **make a backup** of it, and then open your lexicon in Solid. If you're prompted for a template and aren't sure which to choose, just choose 'MDF Unicode' for now; you can always delete your .solid file later and start fresh from a different starting point.

Most buttons have keyboard shortcuts, which you can view by hovering the mouse over them.

Screen Layout

The Solid program is composed of three separates panes – the panes for Marker Settings, Error Filters, and Lexicon editing.



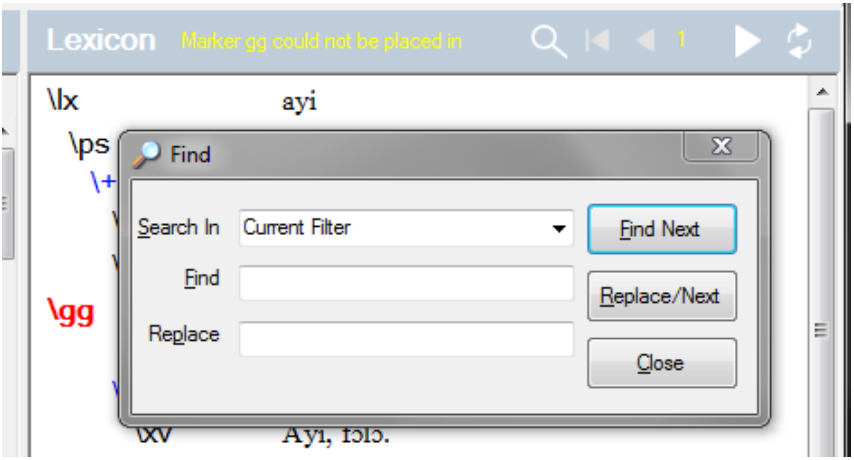
Lexicon

The Lexicon pane on the right is where you will edit your lexicon entries. It is in this pane that you can scroll through and modify the information in your database. Initially it will show the All Records 'filter', starting with the first record, but when you've applied a specific filter, fewer records might be shown. The current set of records is indicated by a yellow filter label at the top.

At the top of the pane there are also buttons for navigating through the set of records. Press the arrows to scroll through them, and the arrow with the line at the left edge to jump to the first entry.

The number between the arrows is the entry number. You can press the left arrow (or Ctrl+PgUp) to select the previous entry, and the right arrow (Ctrl+PgDn) to select the next entry. You can also press the back arrow with the line on it (Ctrl+Shift+PgUp) to skip to the first entry, or Ctrl+Shift+PgDn to skip to the last entry.

You can also use the Find feature in the Lexicon pane (the magnifying glass) to look for selected keywords in either your entire dictionary, or in all entries that have a certain error or warning (i.e. that match your currently-selected filter).

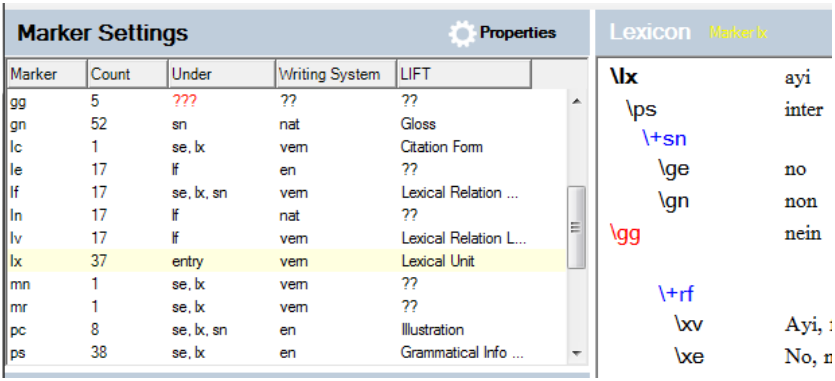


Type in a keyword to search for and press 'Find Next' to scan through your entries. It is first configured to only look through entries that match the current filter (typically selected in the Error Filters pane), but you can change it to look through your entire lexicon by selecting 'Entire Dictionary' from the dropdown menu. In the case shown above, you can either search through the 5 records with \gg errors, or through the entire lexicon.

Error Filters (formerly 'Check Results')

The Error Filters pane holds a list of all the errors Solid has found in your lexicon. Press 'Recheck' to bring up an updated list of all the errors in your lexicon. Each line shows how many instances of a certain error your lexicon has. Some entries can have multiple errors.

When Solid opens, it initially displays an All Records 'filter' which doesn't filter out any records. You can get a similar effect by clicking on the \lx marker's row, since every record has an \lx field.



Usually, as you work, you'll have some other filter selected from one of the left panes.

Some markers have a higher count than the number of entries in your lexicon, like the 'gn' marker in the screenshot above. This is because some markers appear multiple times in a single entry.

Click on a line in list of error filters to begin fixing instances of that kind of error in the Lexicon pane, one record at a time. As each record is shown, the problematic marker will be shown in bold red wherever it appears to be incorrectly place (note that this may actually be the fault of the marker right before it), and it will be shown in bold everywhere it occurs.

Most errors are structural: a field marker is found in an unexpected location, so the entire field (including any child fields nested under it) may need to be moved, or perhaps a missing parent marker will need to be added above it. Other errors relate to the data: the text in the field is not encoded using the encoding expected for that field.

Encoding errors can be serious. Do not use Solid (or any other editor) to save a file unless you are sure it is using the right encodings for your special characters. Otherwise, they may be saved as some totally different character. See "HowTo Set Up Markers Properties".

Under Marker Settings, select any fields that are not in the same encoding as the template you used as your starting point. For example, if you chose MDF Unicode, select each field that is not in unicode, then press the Properties button in the top right of the Marker Settings pane and untick the 'unicode' box.

Marker Settings/Filters

The Marker Settings pane (in the top left corner) is where you can change the settings and labels for any of the markers in your database. To do this click on one of the links in the row for that marker.

This pane also serves as a set of marker filters that you can apply at any time. (Thus, clicking the '\lx' row is equivalent to clicking the 'All Records' row, except that the \lx markers will all be bolded.)

Whenever you open a file, Solid will automatically display a list of all of the markers that currently exist in that file, along with the settings currently associated with each marker. Each lexicon file (e.g. lexicon.db) gets its own custom marker settings, which are stored in a .solid file of the same name (e.g. lexicon.solid). The first time you open a lexicon file, no .solid file exists, so Solid creates one for you by copying an existing template of your choice (e.g. 'MDF Unicode.solid').

A Guide to Solid

The following sections contain more detailed information about using Solid to work with a lexicon.

Getting Started

Backing up Your Lexicon

Before you begin anything in Solid, back up your work! Solid affects your whole lexicon, and lacks an undo button. If you get undesirable results and choose Save, you may not be able to reverse them.

The quickest way to protect your work and just try out Solid is to make a separate copy of your lexicon to use with Solid. Once you are happy with the results you can rename/move both files again. Having two files allows you have both Toolbox and Solid running at once, **but you'll need to remember which file is the master file that you're actively editing.**

Another option is to just make a backup copy of your main lexicon file before each editing session. You can edit the master copy in either Toolbox or Solid, **but don't run Toolbox at the same same that you are editing in Solid.**

A more complete way to make backups and track revisions is to install a version control system such as TortoiseHg for Windows and 'commit' a new version before each editing session. You'd have fewer extra copies floating around in plain sight, **but you'd still need only use one editor at a time.**

Opening a Lexicon

When Solid first opens, everything will be disabled except a the 'Open Lexicon...' button in the upper left corner. Click it to choose a lexicon to begin working with. Remember to make a copy or a backup first!

There are two example lexicons included with Solid that you can open and use for test-driving Solid, named 'BambaraSolidDemo' and 'BambaraTutorial2'. They can be found in *My Documents* | *Solid Examples*. Or, if you already have a Toolbox database on your computer, you can select that instead.

Toolbox databases usually have the extensions '.db', '.txt', 'sfm', 'mdf', or '.lex' at the end of their names, but they could be anything.

Note that Solid is not used to create new dictionaries. It only refines and corrects lexicons already created with Toolbox.

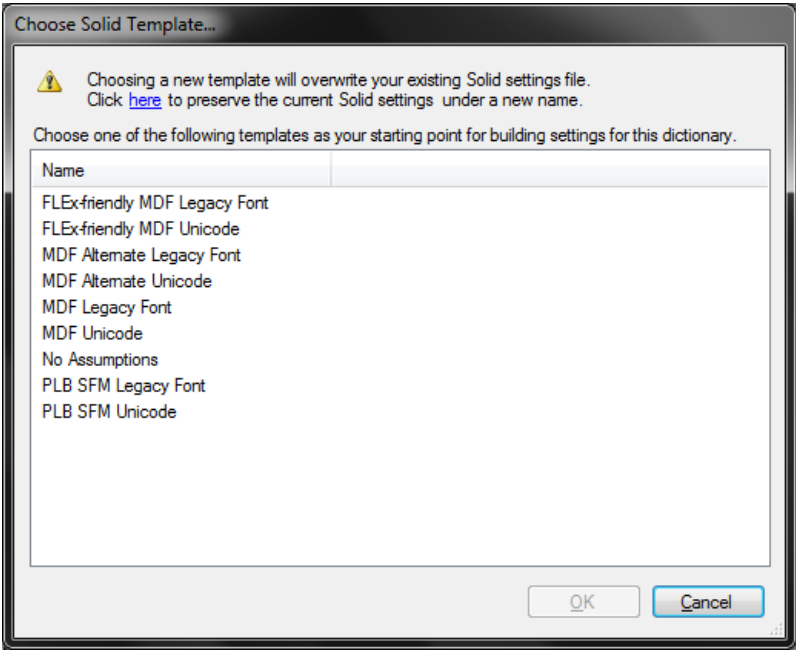
Opening Files More Conveniently

It probably doesn't make sense for Solid to make itself the default editor for the various lexicon file extensions, but you can manually tell Windows to **associate the .solid extension with Solid**. Then, when you double-click a .solid file, Solid will use it to open a file of the same name that's in the same location (with preference given to known lexicon extensions).

The most convenient option is probably to just **copy a shortcut for Solid into your user account's SendTo menu (%HOMEPATH%\SendTo)**, and then you can right-click and send any lexicon file to Solid for editing.

Choosing a Template

After you choose a lexicon to open, Solid will show a window like the one below, prompting you to choose a template to use.



Solid comes with some predefined 'templates', each template is simply a collection of preconfigured marker settings. If you have any special characters and have not yet converted to unicode, you may want 'MDF Legacy', but most users should choose 'MDF Unicode'.

For the sake of completeness, 'MDF Alternate' is provided in addition to 'MDF', but this is a rarely used convention which is usually not compatible with tools such as FLEx and WeSay. (The MDF Alternate hiercharchy was designed to allow a sense to

have one or more subentries nested under it, but it is rarely used and does not allow multiple senses per subentry. A single lexicon should not mix the two hierarchy systems.)

'FLEx-friendly MDF' is a stricter version of 'MDF' which is intended to flag things as errors if they'd cause trouble for FLEx import. It is still being tweaked (and currently has a non-standard structuring of \ps and \sn).

'PLB SFM' is a template for use with lexicons from the Philippine branch. Most of the field markers are different, and the structure is quite different from MDF (and FLEx).

If you're not sure which one to choose, MDF Legacy is the safest template to use. It should preserve unicode data but will not display it properly. However, for most lexicons you should use MDF Unicode--but only if the whole file really is in unicode; otherwise special characters that don't display correctly in Solid may be lost when you save.

The best practice when first opening a file in Solid is to set the encodings, save the file, and immediately use a diff program such as WinMerge to compare it to a backup made beforehand. This is one-time task.

NOTE: The only difference between the unicode and legacy versions of each template is whether every field's 'unicode' checkbox is ticked or not. If you picked legacy by mistake, or if you start with legacy and later want to **switch to unicode**, you can close Solid, open the .solid file in a text editor, and replace all "<Unicode>false<" with "<Unicode>true<". And vice versa.

Once you decide on the Solid template you want to use, press OK to bring your lexicon into Solid.

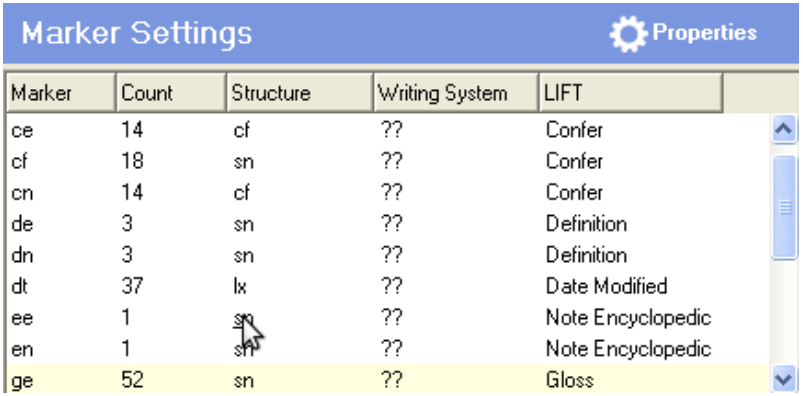
Setting Up Marker Properties

Marker settings/properties are used to:

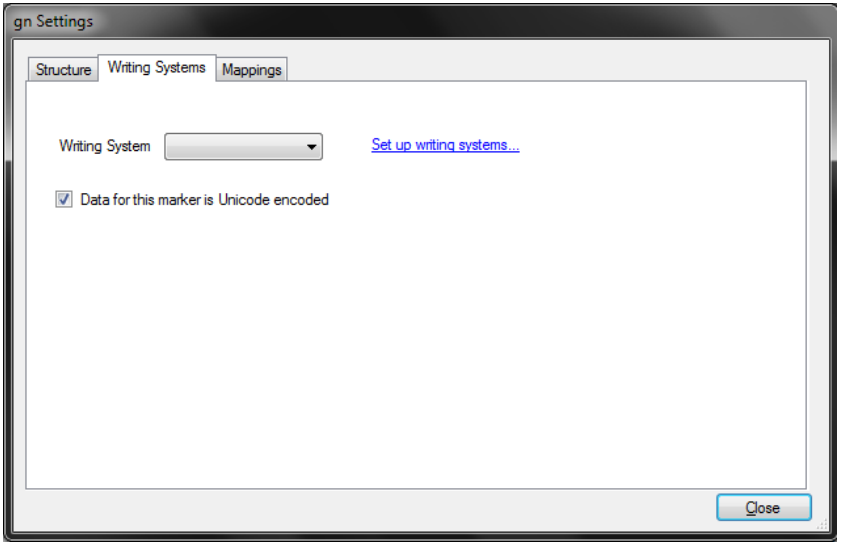
- Set the encoding used for this field's data, and indicate the writing system.
- Set the desired hierarchical structure: that is, what is the parent marker that this one is 'under'? Every marker except the record marker (\lx) must have at least one valid parent marker.
- Set the mapping to LIFT fields, for use in LIFT export.

You can edit the properties of any marker by selecting a marker in the Marker Settings pane and pressing the Properties button in the top right corner of that panel.

You can also select individual properties to change by clicking in any of the columns in the Marker Settings pane.



When you open the Properties menu, there are three different tabs to choose from: Structure, Writing Systems and Mapping.



Writing Systems

In this tab you will find an option to select a writing system, which you may not need at first. (This tab also includes the option to add a new writing system. You can press the 'Set up new writing system...' button to create a new writing system, and then choose that writing system in the dropdown menu.)

More importantly, it has a check box to choose whether or not this marker's data is encoded in Unicode. **These encoding settings must be set correctly right away--for all fields--before you can safely click Save.**

For example, if you have chosen MDF Unicode as your Solid template, but one of your marker fields is not Unicode, you must uncheck this box for that marker field. (For every field that is not in unicode, press the Properties button in the top right of the Marker Settings pane and untick the 'unicode' box.)

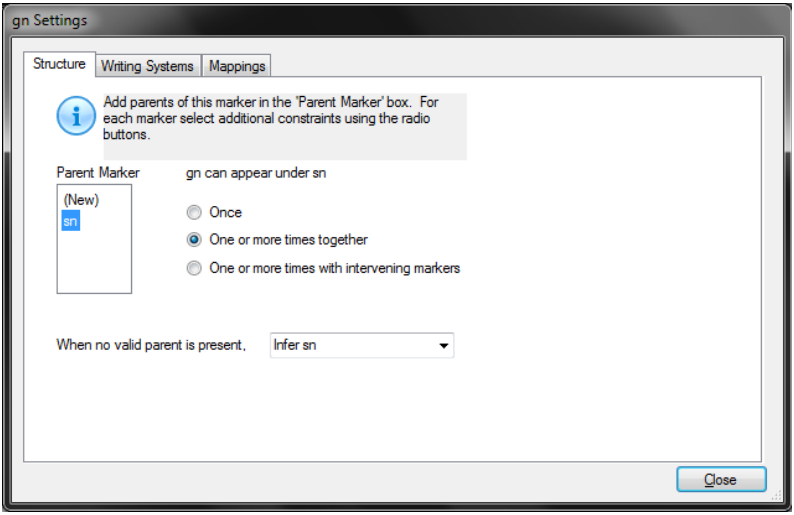
Mixing multiple encodings within a single text file is usually a bad idea, however, and Toolbox and Solid are among the few editors that can handle such files. It's much better to just convert everything to Unicode if possible.

Incorrect encodings can be serious. Do not use Solid (or any other editor) to save a file unless you are sure it is using the right encodings for your special characters. Otherwise, they may be saved as some totally different character. (In Solid, if you have legacy fields that are incorrectly set as Unicode, their special characters will be permanently replaced with question marks when you save.)

Again, if a field marker is not Unicode encoded – if it uses a Legacy font – but you've left the 'Unicode' box checked, any special characters in that field will be lost when you save. (Conversely, however, if a field has Unicode data but is not set correctly, the special characters will most likely be preserved but not displayed correctly.)

Marker Structure

In the Structure tab you can choose what markers the marker you're editing should be allowed to occur under, by selecting its Parent Markers. For each Parent Marker you can also choose how many times the marker you're editing can be nested under a single valid parent marker. These are the options: once, one or more times 'together', and one or more times with other sibling markers in between.



The 'together' restriction does not apply to child fields, so if \se is a child that can occur multiple times together under \lx, and \ps is a child of \se, then in the following sequence the \se fields are still 'together', as the indentation shows:

\lx
. \se
. . \ps
. \se
. . \ps

It is almost never necessary (nor recommended) to choose the third option. One example might be where there are extra kinds of subentries (such as \seco for compounds). In that case, the third option would be needed to make the following sequence acceptable. (Otherwise, the second \se would trigger an error.)

\lx
. \se
. . \ps
. \seco
. . \ps
. \se
. . \ps

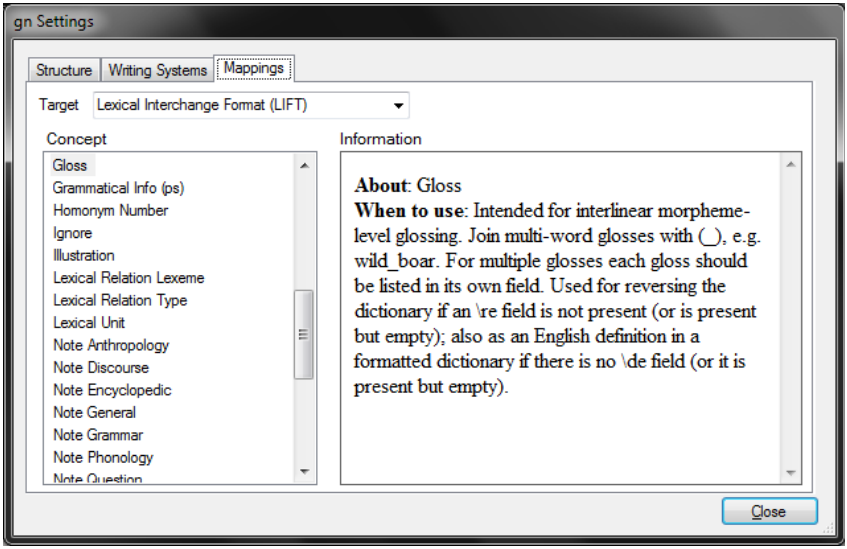
The Structure tab also has the option to choose what Solid should do when there is no valid parent above the marker in question. You can choose to either have an error appear (recommended), or have Solid automatically infer a parent field.

Inferred parent markers can mask other problems and generate extraneous fields, so they should be used sparingly, especially if you plan to "make inferred markers real" at some point.

In a typical MDF scenario, only \sn and \rf would be inferred, and by as few child fields as is feasible. Thus, if most senses begin with \ge, you'd tell Solid to infer \+sn above \ge as needed, and to infer \rf above \xv as needed.

Marker LIFT Mapping

The last tab in Properties, Mapping, shows the meaning/function of the marker. You can indicate in this tab whether a marker is for glosses, parts of speech, example sentences, etc. These are used when exporting to LIFT, and they are also useful for helping people later on to make better sense of your markers' intended meanings.



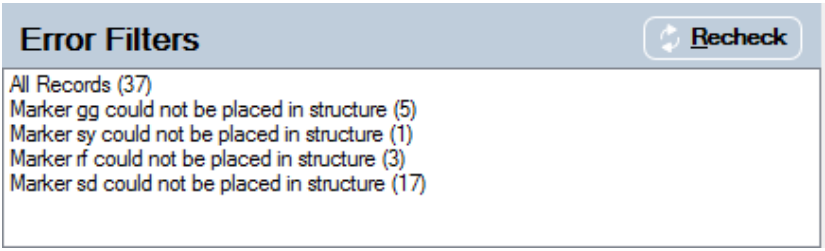
Manually Editing the Lexicon

A Closer Look at Errors

To edit your lexicon without Quick Fix, you will need to use the Error Filters pane and the Lexicon pane. The Error Filters pane will tell you what errors exist in your dictionary, and you can use the Lexicon pane to correct these errors.

Every line in Error Filters is a type of error Solid found in your lexicon. The number in parentheses represents how many entries contain that type of error.

The numbers beside the different errors may look large sometimes, but keep in mind that some entries will appear under multiple error types. The number in the Error Filters pane reflects the total number of errors, not the number of entries that have errors. Also, a single fix, such as adding a missing parent marker, will often fix multiple errors.

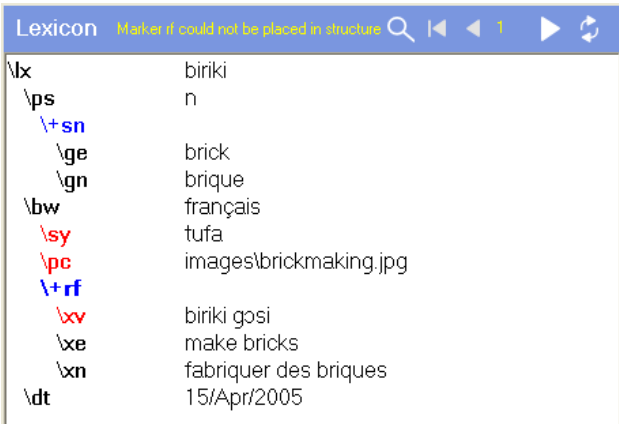


For example, in the above pane there are 5 entries where the 'marker gg could not be placed in structure', as seen by the first error line.

If we wanted to fix one of the errors – in this case, let's say the 'marker rf could not be placed in structure' line that we've highlighted – we need to click on that line and bring up those errors in the Lexicon pane.

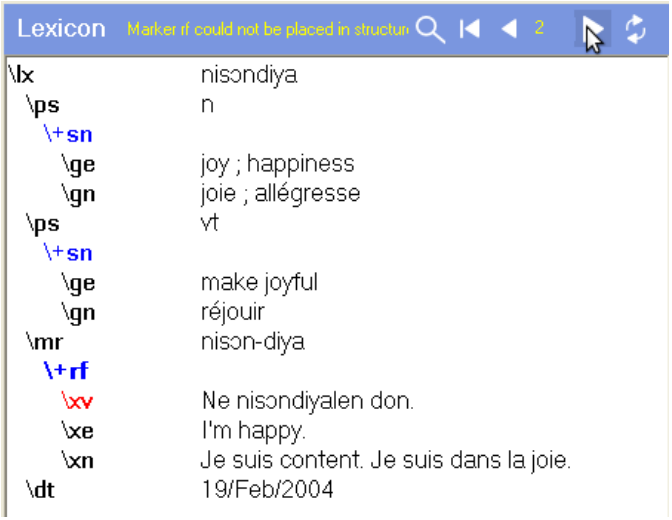
Editing Lexicon Entries

When you select a line in Error Filters, the Lexicon pane immediately changes to view only errors of that type, as with the \rf error below.



The yellow words at the top explain which filter is currently being applied--in this case,

only entries containing these \rf error will be shown. You can use the arrows on the top right part of the screen to scroll to another entry with the same form of error.

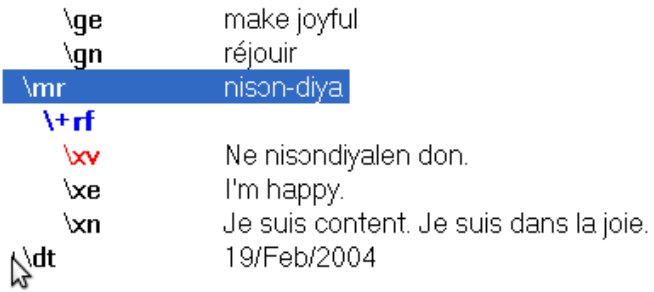


In order to fix the errors that Solid finds, you will need to have an understanding of how the markers work in Toolbox lexicons. In the above example, the marker \mr is in the wrong place. It needs to go just below the \lx marker, or else just above the \dt marker.

In this example, note that although Solid said that the error was with the \rf marker, it is the \mr marker that we need to move. Very often, the errors Solid finds can be corrected by moving or inserting a preceding marker.

First, to the left of the marker you need to move, click all the way at the left edge and that should select the entire line. Otherwise, click to the far left, then drag to right, dragging beyond the last character to make sure you select the entire line including the newline (carriage return) at the end. A third option is to use the keyboard: Home, Shift+End.

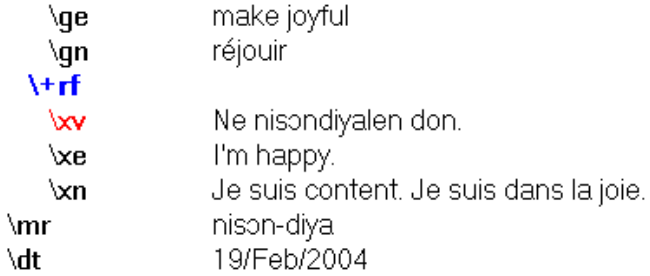
Now, you can drag the line to where it needs to go, as shown below.



Or, you can use the keyboard, using Ctrl+X to cut the selected line, down arrow several times, Home if necessary (it shouldn't be necessary), and Ctrl+V to paste it in the desired location.

If you don't fully select the line you are preparing to move, you may end up with a blank line in the first location, and one field merged into another field in the second location.

Once you have the markers in the right order, press the refresh icon in the top right of the Lexicon pane (or F5). If you corrected the problem, the red marker will disappear.



The right pane will also automatically refresh itself whenever you move to the next or previous record.

Checking Progress

As you fix the problems Solid is showing, you might notice that the Error Filters panel still shows the same number of errors. This is because it doesn't automatically refresh as you work. You will have to click Recheck (or Ctrl+F5) periodically in order to get an accurate picture of your progress.

In this case, after we press Recheck, several error types decrease in number. This is because, as said before, the lines show the number of errors, not the number of entries that have that type of error. Correcting a single entry may fix multiple errors.

By using the error lines in the Error Filters pane, and correcting the problems with the Lexicon pane, you can manually fix your entire lexicon.

Correcting Errors with Quick Fix

Checking for Errors

Once you have your marker properties configured in Solid, you can see potential problems with your data in the 'Error Filters' pane. In some cases, you can use a Quick Fix function to fix up your lexicon, which is much faster than editing manually. But these fixes are also risky, so please confirm, "I know what I am doing and have backed up," in order to run them.

The changes you make with Quick Fix happen across your entire lexicon and are entirely irreversible. Use Quick Fix cautiously, and back up often.

One good practice is to back up first, then use a diff tool such as WinMerge to compare the two files and get a 'before and after' view.

Making corrections is a two step process:

1. Use the Error Filters pane to find where the errors are.
2. If there are only a few errors, try to fix them manually in the right-hand pane. If there are many, either loosen up the settings (if these actually are not errors) or else do a bulk fix operation. It may be possible to do this using a provided Quick Fix; otherwise, you'll likely need to do a regular expression Replace All (perhaps in a different editor such as Eclipse or Notepad++).

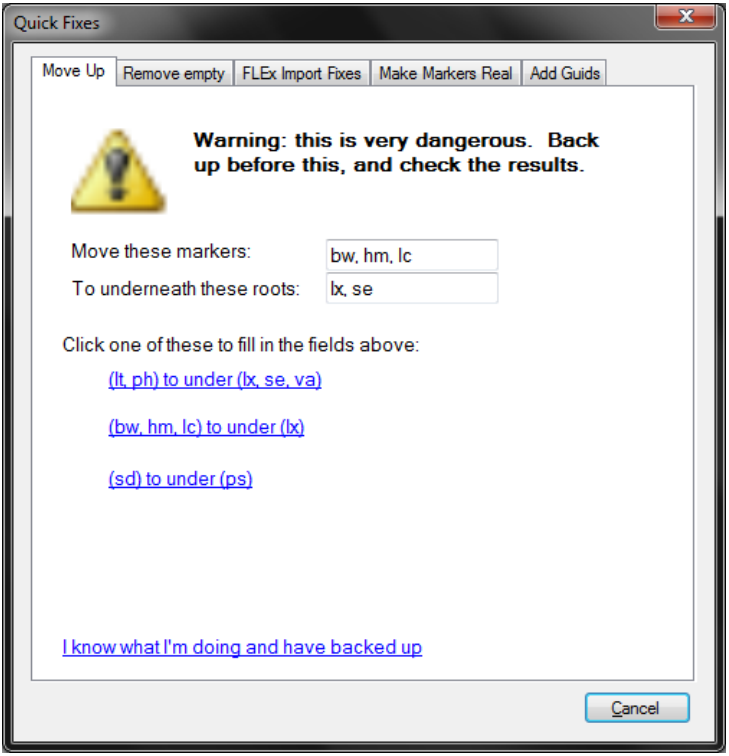
To begin, press 'Recheck' in Error Filters. This will give you a list of all the potential problems detected in your lexicon.

These errors are caused by markers being out of the order your template says they should be in. To fix this you will have to rearrange your markers to align with the order your template dictates. You can do this manually in the right hand pane or you can use Quick Fix for some quick bulk edits.

The manual method of re-arranging the lexicon entries will be covered later.

Running the Move Up Quick Fix

The Move Up function moves all markers of a selected type to just underneath markers of another type.



In this dialog you can choose which markers to move and where to move them. There are three presets included that are often used. Based on the patterns you saw while browsing through the error filters, you can decide which markers need to be moved, and where.

Once you've decided on which markers you want to move, and have made sure your lexicon is fully backed up, click on the line 'I know what I'm doing and have backed up'. Quick Fix will then go through your entire lexicon and move all of the markers of the type you selected.

For example, the manual task of fixing the misplaced \mr fields could be solved by moving it up under \lx. (There is no Move Down quick fix, but moving up results in much less ambiguous data--which also makes it migrate to FLEx more easily). But first, you should make sure that that is always the correct interpretation. (If you've also use \mr under the \se field sometimes, then this 'fix' will shove those all up under the root \lx, and there will be no way to automate a reverse fix.)

Remove Empty (risky)

This tab gives you the option to instantaneously remove all empty fields in your entire lexicon, except for specific fields that you choose to exempt. Solid will remove all

empty fields in your lexicon that were not in the text box in the Remove Empty tab. This can significantly decrease export time if you are exporting to LIFT, but it is dangerous unless you first make sure you've exempted all fields that might be parents of other fields.

FLEx Import Fixes

This tab is important if you are preparing to move your Toolbox lexicon into FLEx, because FLEx import has certain quirks and expectations about the data.

Make Markers Real

This is one of the most useful quick fixes, as it takes Solid's interpretation of the lexicon file and adds structural markers in as needed. This is a useful last step when cleaning up a dictionary, but it should not be used until you are confident that all the fields Solid is inferring really ought to be inferred.

This fix is most useful for inserting missing \sn and \rf fields.

Switching Templates

As you go through your marker settings and make changes, the template you are using is updated. So even if you started with a copy of the MDF Unicode template, if you make changes to your markers, your .solid file will no longer be identical to that template.

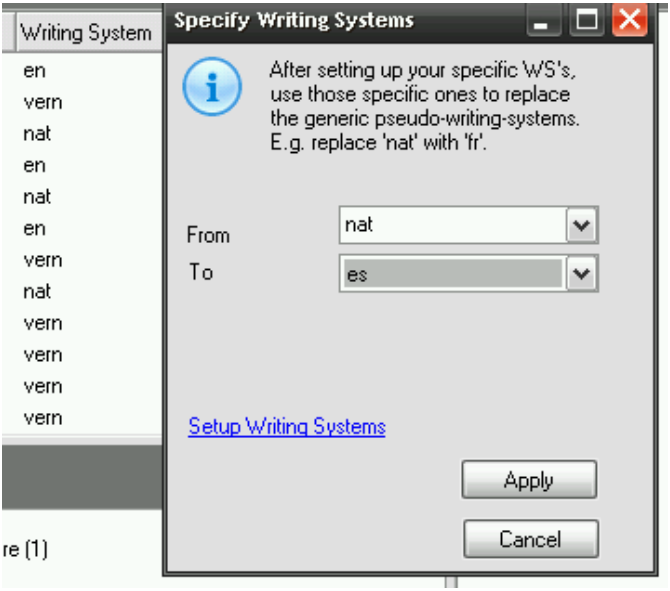
If you would like to switch back to one of Solid's presets--or to any .solid file in the same location as your lexicon--you can use the 'Change Template...' button at the middle of the top of the Solid screen.

This is basically the same as manually deleting or renaming your .solid file and then reopening your lexicon in Solid.

Filling in Your Writing Systems Quickly

If you'll want to export to LIFT, or you just want to document clearly which languages are being used in which fields, you don't have to set these manually for every field.

The Solid templates come with generic values set for the writing system (WS) of each field marker. For example, \ge comes already set to English. However, each project will have its own vernacular WS, and national and regional languages will also vary across projects. So, the MDF templates comes with many fields' WS's set to these placeholders ('pseudo' or 'meta' writing systems): vern, nat, and reg. Once you have set up each WS that you intend to use, you can use the Specify Writing Systems feature to replace these with actual WS's.



Exporting to LIFT (experimental)

The export function is not a finished product. The normal way to migrate and SFM lexicon to FLEx is via the SFM importer built into FLEx. However, if you would like to try exporting lexicon directly into the LIFT format (an XML format), you have that option.

The LIFT format that Solid's export uses is outdated. Solid will export in LIFT 10, whereas the current format is LIFT 13. However, LIFT 10 will still work in some programs, like WeSay.

After choosing Export, you can choose the name to export your lexicon as, and the format you would like to save it in. Once you've chosen these, press Save to complete the export.

Make sure that all of your fields are explicitly mapped to a LIFT field or to Ignore, or you will get errors during the export.

The export feature has a option for reporting errors to the developers, but please make sure first that the errors are not data errors. The LIFT export is quite strict in what it expects. (E.g. it will fail for unmapped fields or missing writing systems.)

Appendix A: Installation process

Running the Installer

To download Solid, please go to http://solid.palaso.org. There is a link on that page for downloading the latest version.

Once the file has finished downloading, double-click on the icon to start the setup. You will probably need to approve installation of software from "an unknown publisher", and you will need to accept the license agreement.

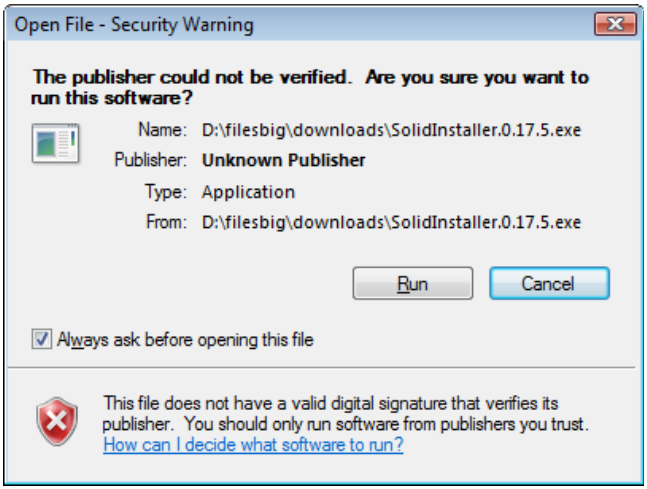
You can open Solid immediately by pressing Finish without unchecking the 'Run Solid.exe' check box. However, if you want to open Solid later, you can find it in *Start Menu* | *Programs* | *Solid*.

When you open Solid for the first time you may see a dialog requesting that your register to send usage statistics to the developers of Solid. This is optional; you can decide whether or not to share your information, then press OK to continue.

Once you've finished this, Solid will open and you can begin working with the sample project (recommend), or working directly with your own lexicon.

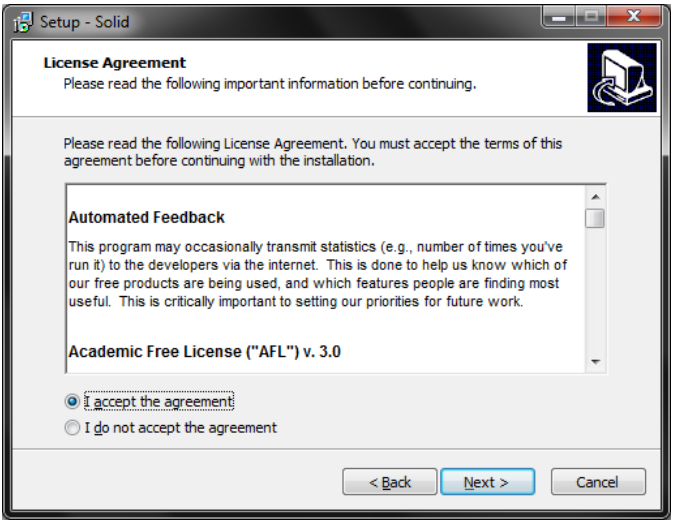
Several simple screens (shown below) will appear during the installation of Solid.

1. Click Run.



2. Click Next.

3. Click to accept, then click Next.



4. Click Finish.

5. Registration