

Introduction to Shoebox and Toolbox

with notes on Econv, Transcriber and Elan

Introduction

This document augments the workshop session on Shoebox and Toolbox and is intended to serve as a guide for those who either have no prior knowledge of these tools or who are unsure how to apply their existing experience in the Dobes context. As such it is not a complete 'users manual' but concentrates on matters of particular relevance to corpus building, i.e. lexicons, interlinearized texts (linked to audio/video) and dictionary production.

Where information is relevant to both Shoebox and Toolbox I shall simply refer to 'Shoebox'. Toolbox functionality goes beyond Shoebox and some notable differences will be mentioned as they arise.

All the files referred to below, plus those needed in support (e.g. the Shoebox setting files) are included in the subfolders. The example project files can be conveniently opened by double clicking on the shortcuts in the 'data' folder (e.g. Shoebox_Session\data\MyProject_1.prj) - the projects themselves are stored in the settings folder as discussed later. Ideally you should install 'Shoebox_Session' directly under your C: drive – this gives Shoebox the best chance to locate the data files when you open a project file (Shoebox uses absolute paths to find the files belonging to a project). If however you get a message on opening a project indicating that the program can't find a particular file, navigate to your Shoebox_Session\data folder to locate the file. The program should then open this, and any other required files without further complaint. You will have to do this (once only) for each project that you open.

Overview of Contents

A Dobes workflow logically starts with the session recordings and proceeds via the mark-up process to Shoebox and Elan. Because of this, a typical way to set up a new Shoebox project is to open a newly marked-up text (i.e. the transcribed text result of an audio/video session) in Shoebox, link it to a new or existing Lexicon database and then interlinearize the text and/or add a free translation to it. This edited version can then be imported into Elan so that it can be seen in the context of the associated media.

The example Shoebox 'projects' described below are generally set up to illustrate this scenario. Note that there are many other possible arrangements for a project. For example it may consist of just a lexicon, or set of lexica. Or it might contain a concordance linked to a set of text files. Moreover there are many ways to set up the 'database types' of the various files – different markers sets, 'processes' (e.g. the parse process for interlinearizing), language encodings, etc. What follows then, is certainly not the only way to set up a Shoebox project but a particular scheme that works with Transcriber and Elan, as well as the Shoebox Multi-Dictionary Formatter (MDF).

If you are just beginning, I would suggest that, rather than attempting to build something from scratch you simply adopt the example projects supplied as your starting point. Add some of your own data, delete the example records and experiment. By examining the existing settings (particularly the database properties for each file type) you will find out how to customize things to your taste, and meanwhile you will have a functioning system.

The detailed instructions that follow should help you to:

- make a new Shoebox project to act as a 'container' for your data
- set up a lexicon database and modify the settings to suit
- set up text databases and associate them with the lexicon
- start to build the lexicon while interlinearizing the texts
- experiment with alternative views of the data; browsing and sorting
- export lexicon data via the MDF to a Word/PDF/Lexique Pro file
- refine parsing – considerations for dictionary
- export texts to Transcriber (from Shoebox)
- process texts with audio in Transcriber
- import texts from Transcriber via Elan to Shoebox
- export to Elan

Whether you start from scratch or use our example projects will determine how many of these steps below you will need to take. If you are starting with our project you can skip to 9.3. You can refer to 1. through 9.2 when you want to significantly modify the project or start a brand new one.

Finally, a crucial step is getting the transcribed, marked up text file into Shoebox in an appropriate format (eg the format used in the text files in the example projects) in the first place. It is equally important to be able to move the data back and forth between Shoebox and Elan. The techniques we use involve the use of Transcriber as well as Elan and are described fully in the appendix at the end of this document, 'Moving data between Transcriber, Elan and Shoebox/Toolbox'.

Detailed Instructions

Bold_type indicates where you would substitute your own file-names for ours. (If you are doing an exercise using the files in the Some Shoebox Settings' folder do make sure you use your own names – otherwise you will simply overwrite the example projects and lose the information in them).

SMALLCAPS/SMALLCAPS indicates a computer action via the regular menus or right-click contextual menus.

Highlighted_text indicates an example file that you can inspect (all such files are in the 'data' folder)

1. Starting the first project

1.1 Open Shoebox. Select PROJECT/NEW. (If another project is already open it will simply close when you open another or create a new one, unless it has been modified in which case you will be prompted to save it first). Enter the name (**MyProject_1**). Choose a location for the project file. Usually this would be the 'My Shoebox Settings' folder created by the Shoebox installation procedure – this is suggested as the best place for project files as it gives all projects potential access to all default and user-created definition files such as database types and language encodings. You can place the project in any folder however – but you will have to copy all necessary files to that folder. (The 'Some Shoebox Settings' folder we are supplying here is an example of such a custom made folder. For the purposes of this guide you will find all the relevant definition files for the example projects in that folder and the data files in 'data'). Select PROJECT/CLOSE then 'Exit the Shoebox program'.

1.2 Optional: create a shortcut for this project (RIGHT CLICK ON THE PROJECT FILE ICON/CREATE SHORTCUT) and place it somewhere convenient, e.g. on the Desktop (you can't move the project file itself without moving all its associated files so this is a way to get to it a bit faster).

2. Establishing a database type for your lexicon compatible with the Multi-Dictionary Formatter

2.1 Make a copy of MDF.typ (which you will find in 'My Shoebox Settings'). Paste the copy back into 'My Shoebox Settings' (remember, for this exercise you can find everything in 'Some Shoebox Settings') and rename it **MyLanguageLexic.typ**. This is a good way to start using MDF compatible markers because by copying and renaming the MDF.typ file you automatically include not only all the relevant markers but also supporting information about their use.

2.2 Open MyLanguageLexic.typ in a text editor (e.g. WordPad) and change the first line from \+DatabaseType MDF 4.0 to \+DatabaseType **MyLanguage** Lexicon Save and close file. (This step is not strictly necessary, but will give you useful information when selecting database types within Shoebox).

3. Starting your lexicon

3.1 Double-click on the **MyProject_1.prj** icon or the shortcut if you created one. This opens Shoebox with the selected project open at the last saved stage. Select FILE/NEW then type **MyLanguage.Lex** and navigate to the folder where you wish to store it (in this case 'Shoebox Examples'; it is recommended that you do not store your data files in the 'My Shoebox Settings folder'). Click 'Save'.

3.2 A 'Select Database Type' dialog box opens. Select **MyLanguage** Lexicon from the list, then click 'OK'. A window will open with the newly created lexicon open with a sample set of markers supplied for the first record. This sample is based on a template. To see the names and hierarchy of these markers click on the new **MyLanguage.Lex** window to make sure it is active, then select VIEW/BOTH MARKERS AND NAMES then VIEW/MARKER HIERARCHY. To get more information RIGHT CLICK on any marker. To see the full range of markers currently available (which are all those used by the MDF process) select DATABASE/PROPERTIES. Note: if you are using a custom settings folder (such as our 'Some Shoebox Settings' folder which was originally empty), you might notice at this stage that various other files (required by Shoebox) such as Default.lng, Date.lng, etc. have been created automatically. If you are using the default settings folder these already exist.

3.3 Note that although you now have a functioning database type for your lexicon you do not yet have any special language encoding data for it. Depending on the nature of the language you are trying to describe it may be important to set something up at an early stage. However you can get started with the default settings provided and since this topic is not particularly crucial to our immediate needs yet is unfortunately rather complex we do not deal with it here.

4. Modifying the template to include further markers (e.g. the Homonym number)

4.1 With the **MyLanguage.Lex** window active, select DATABASE/PROPERTIES. Scroll through the list to find the /hm marker then click 'Modify'. This will give additional information about this marker including its position in the hierarchy: this field comes directly after the \lx field.

4.2 Return to the lexicon window ('Cancel', 'Cancel') and place the cursor in the right hand pane on the same line as the \lx marker. Press 'Enter' then type a backslash. Immediately the cursor will move to the left pane and a drop down list will appear. You can either pick the 'hm' marker from the list or simply type the letters. Press 'Enter' again - the marker is added in the correct position.

4.3 Select DATABASE/TEMPLATE, then click 'OK'. The template is now updated.

5. Adding alternate form (\a) and underlying form (\u) markers to assist in interlinearizing

5.1 These markers are not used by the MDF process, but they are useful for solving various parsing problems when interlinearizing. Refer to the ShParse.pdf file which comes with the installation (a copy is in the 'other documents' folder). With the **MyLanguage.Lex** window active, select DATABASE/PROPERTIES. Click 'Add'. On the 'General' tab move the cursor to the 'Marker' field and type the letter 'a'. Optional: in the 'Field Name' field insert 'Alternate Form'. Check that in the 'Hierarchy' field 'lx lexeme' (the default) is selected. Click 'OK'.

5.2 Add the \u marker in the same way, typing 'Underlying Form' if desired. Click 'OK' again to return to the lexicon window.

5.3 If you want you can add these fields to the template following the procedure in 4.2 and 4.3. You might place them directly after the \hm marker (\a then \u).

6. Establishing a database type for your transcription (texts) compatible with Elan

6.1 Copy the file ElanExportText.typ (which you will find in 'Some Shoebox Settings'). Paste the copy into your settings folder (e.g. 'My Shoebox Settings'). Note this database type is not only compatible with Elan for importing and exporting Shoebox files, it also reflects graphically the hierarchy of markers in both programs.

7. Opening a raw transcription for processing. Note: you will not need this step if you are bringing in your marked up texts via Elan, as described later in the appendix. It is only for processing 'raw' texts that consist of the words and punctuation only and have not been opened in Shoebox already.

7.1 Understanding the Change Table file (.cct) for text processing: When you open a raw text in Shoebox for interlinearizing you will need to specify a change table. Which one you choose will depend on how you wish to break down your text. If you choose the default table supplied for this purpose (TextPrep.cct – found, as usual, in the My Shoebox Settings folder) a new record will be made whenever the following punctuation marks are encountered: .?! We decided that we would like to break records on commas and colons too in order to make each record more equivalent to an intonation unit. To achieve this we made a copy of TextPrep.cct, (renamed TextPrepComma.cct for convenience), and edited it to include these marks. A copy of both files can be found in 'Some Shoebox Settings'. If you compare them in a text editor you can see how easy this modification was. Under the section `begin >` near the top of the file you will see the line `store(setSentPunc)` What follows are the punctuation marks that trigger a new record. With this in mind any other desired modification can be made. For instance if you wish to break records on hash marks simply add the # symbol to the list.

7.2 Select FILE/OPEN , navigate to where your text files are located then choose **MyText.txt** . An 'Import' dialogue box will open. Under 'Database Type' choose ElanExport (This is the ElanExportText.typ file from 6.1). Under 'Make Consistent Changes' check the box then click 'Browse' and from within 'My Shoebox Settings' choose one of the .cct files (i.e. TextPrep.cct or TextPrepComma.cct). It is a good idea to also check the box 'Backup up original as *.ORI'. Then click 'OK'. The processed file will open at the first record. (Note that if you checked the 'Backup ..' box Shoebox automatically makes a backup of the original unprocessed file with the extension .ori, i.e. **MyText.ori**. Some confusion can arise from this convention because Shoebox preserves the .txt extension for the new, processed file - so processed and unprocessed files can have the same .txt extension but quite different contents and status, while .ori files are the same as unprocessed .txt files in content).

7.3 Optional: now that you have two windows open (the lexicon file and the text file) you can quickly tidy up the layout by selecting WINDOW/TILE.

8. Setting up the interlinear processes

8.1 Shoebox has a convenient ‘Quick Setup’ facility which we shall take advantage of, with a few modifications along the way. Import a text, e.g. **MyText.txt**, into Shoebox as just described in 7.2. With the focus on this file, select DATABASE/PROPERTIES/INTERLINEAR TAB. Then click ‘Quick Setup’. A pop-up box will appear – accept the default text markers for now (you may decide to change these later to make them compatible with your own or the Dobes conventions). Ignore (i.e. click ‘OK’ to any warnings about markers already existing in the database). Another pop-up box will appear – in the ‘Available Databases’ pane highlight **MyLanguage.Lex** (you should already have this database file open in the project, if not follow steps 3. to 5.3) then click ‘Insert’ (or simply double click in the pane). This will place that database in the ‘Databases to Search’ pane. Probably the gloss marker default will be ‘g’ – change it to ‘ge’. Likewise for the Part of Speech marker – change it from ‘p’ to ‘ps’. These changes are necessary to make these markers conform to the MDF marker set. Leave the ‘a’ and ‘u’ markers unchanged. Click ‘OK’ – this will return you to the interlinear tab which will now indicate three lines: one ‘Parse’ and two ‘Lookup’ processes.

8.2 While this interlinear tab window is still open check each of the processes in turn. With the Parse line highlighted click ‘Modify’ and then ‘Lexicons’ – in the ‘Databases to Search’ pane you should see **MyLanguage.Lex**, ‘Markers to Find’ should contain ‘lx’ and ‘a’, ‘Marker to Output’ should contain ‘u’. Repeat for the two Lookup lines. In both cases the ‘Database to Search’ should be **MyLanguage.Lex**, and ‘Markers to Find’ should contain ‘lx’, while ‘Marker to Output’ should be ‘ge’ and ‘ps’ for the first and second process respectively. It is worth making a note of these settings as it is easy to lose them if, for example, you inadvertently change the path to the Lexicon by moving or renaming a folder (Shoebox is in general quite fragile in this respect). There are, as you will see, many other settings that you may wish to change, but this default setting is the one used by this example project.

9. Setting up the jump path and jump target

9.1 It is useful to set up a ‘jump path’, (which is an association between one or more markers in one database and one or more markers in other databases) for the text databases. The idea behind this is to allow you to ‘jump’ from any relevant item in the text to the matching item in the lexicon, or to allow you to insert that item in the lexicon if it does not. This is a quick way to build and edit a lexicon through the transcription of texts. With the focus on **MyText.txt**, select DATABASE/PROPERTIES/JUMP PATH TAB. Then click ‘Add’. A pop-up box will appear – type a name for this path (e.g. ‘path1’) then check the ‘Default Path’ box. (This is a slightly crude way to proceed as it simply adds all available markers in the ‘source’ to the ‘Fields to Jump from’ pane; alternatively you can uncheck the box and add only those markers that are relevant, e.g. ‘t’ and ‘m’). Under the Destination section highlight **MyLanguage.Lex** in the ‘Available Databases’ pane then highlight ‘lx’ under ‘Field to Search’ list then click ‘Insert’ to add this combination to the path. Click ‘OK’ twice.

9.2. Test the jump path as follows. With the focus on **MyText.txt**, right click on the first word in your text. Since at this stage in your lexicon you have no entries, a ‘No Matches’ pop-up will appear, Click ‘Insert’. A new window showing the Lexicon will appear with the selected word added as a record. Edit the record (e.g. trimming off suffixes, etc) and add information in other fields as desired. (If the word already existed as a record the window would appear with that record selected – you will see this next time you right click on the word in the transcript).

9.3 Each time you right click on word in this manner a new window will open, regardless of whether you are inserting a new or jumping to an existing record. This is not normally what you want since you simply end up with multiple views of the same lexicon. This is where the ‘jump target’ comes in. Close the newly created window **MyLanguage.Lex:2**, move the focus back to **MyLanguage.Lex** then select VIEW/JUMP TARGET. To test, right click again on the first word in **MyText.txt**. The focus will automatically move to the appropriate record in the **MyLanguage.Lex** window rather than in a new window. At this point you could open our example projects **MyProject_1.prj** (which is the result of following all the previous steps, with some of our example data, but before setting the jump target) and **MyProject_2.prj** (the same, but after setting the target).

9.4 ‘Jumping’ is an area where Toolbox differs from, and has more to offer than Shoebox. For example one can return to the point from where one jumped with the shortcut combination CTRL+R. Refer to the Toolbox Help file.

10. Building the lexicon by jumping, directly, and through interlinearizing

10.1 In the last step we added a record by using the jump path process. Of course one can add records simply by working directly in the lexicon: With the lexicon window in focus select DATABASE/INSERT RECORD – or more conveniently use the shortcut combination CTRL+N. Likewise one can edit or delete records. Every time a record is added or edited its date stamp marker (dt) is updated – this can be a very important way to keep track of the history of the lexicon, e.g. through the use of a filter targeting a date range.

10.2 Another way to build the lexicon is through the interlinear process itself. In a sense this is a reversal of the usual process of interlinearizing texts using a lexicon. For this to work one setting must be enabled in the interlinear process. With the focus on **MyText.txt**, select DATABASE/PROPERTIES/INTERLINEAR TAB. Then with 'Parse' highlighted click 'Modify'. Under 'If parse fails' check the 'Insert into Lexicon' box. Returning to the first text window, click on the first word, select TOOLS/INTERLINEARIZE or ALT+I or the interlinearize icon on the toolbar. The interlinear process will look up and fill in the data for the first word (since you have already added that to the lexicon) but will stop at the second. A 'Lookup Failure' box will appear. Click 'Yes'. The word is added to the lexicon and the focus moves to that record and the interlinear process stops. To move on return the focus to the text and click the interlinearize icon again. If you re-interlinearize the text any relevant information you have added to the lexicon will be added to the text.

10.3 Continue adding records to the lexicon by any of the above means - make sure you at least add data to the 'ge' marker in each case (because we are shortly going to sort by this field). Note that you can at any time select interlinearized lines in the texts (by dragging the mouse over them) and delete them in order to start over again if you wish.

11. Further considerations for interlinearizing

11.1 At this point you should be aware of one of the potential pitfalls of using Shoebox. Because it is not a 'relational database', but rather uses lookup processes to generate interlinearized text (as well as other output such as concordances), Shoebox does not keep the data in the interlinearized texts aligned with the data in the lexicon. In other words when you make changes to the lexicon these will not automatically be reflected in the interlinearized texts – therefore the texts very quickly become out of date. This is a real problem that requires meticulous attention to the order in which tasks are undertaken. Of course you can at any time re-interlinearize a text but this is a far from trivial task as, even with use of forced parsing and word formulas (discussed later) you will be faced with many ambiguity decisions – decisions which you have made before but which are not 'remembered' by the program, as it were. It is inevitably a time-consuming task. Moreover you have to actually remember to update your work – nothing is done automatically. Toolbox, has gone some way to address this problem with something called 'Interlinear Verify' which 'will verify that all interlinear annotations correspond to entries in the lexicons'. This performs a series of checks that look for discrepancies between the lexicon and the text. However, you still have to remember to actually perform this task!

11.2 Before going much further you should also give some thought to how you plan to work with Transcriber and/or Elan. There are several potential workflows but I would recommend two:

11.2.1 If you are starting with new audio/video data then work in Transcriber first in order to add the time code directly to the transcription as you make it (working with Transcriber is dealt with later). Then export to Shoebox via Elan (see the appendix 'Moving data between Transcriber, Elan and Shoebox/Toolbox'). Shoebox will read such a file as if it has already been processed – i.e. you will not have to specify database type or change table. Then interlinearize the text in Shoebox. Finally import this file with associated audio/video into Elan for viewing and possibly further editing.

11.2.2 If you have existing transcripts that relate to old audio/video data and which you now wish to mark up with time code, then import the transcripts into Shoebox first – using the techniques described earlier (7.) You could interlinearize them at this point but it is better to export via Econv to Transcriber format before interlinearizing (again, some text editing might be necessary– details later). Open in Transcriber and add the time code – this is relatively quick as the text has already been broken down into the desired 'chunks' as defined by your punctuation conventions and the change table used – all that remains is to align the borders - called breakpoints - of these chunks with the audio signal to create the time code data. Then re-export to Shoebox via Elan and continue as in the previous step, i.e. interlinearize the text in Shoebox.

11.2.3 One reason for interlinearizing later is that you don't have to bother about hiding the interlinearized stuff in Transcriber, although this is not difficult to do. (In Transcriber, after opening the .trs file, select OPTIONS/EVENTS/EVENTS DISPLAY – this opens a 'Configure Events' panel. Remove the contents, {%s}, of the 'Comment:' box – this will suppress the interlinearized lines) However there can be big problems with this method: what happens is that the interlinear alignment generally gets messed up. Therefore I repeat the recommendation that you DO NOT begin interlinearisation in Shoebox prior to exporting to Transcriber. (If you do, it's not a disaster - you will just have to re-interlinearise most of it when you bring it back to Shoebox = wasted work!).

12. Manipulating the look of the data – Browse View and Browse fields

12.1 With the focus on the **MyLanguage.Lex** window, select VIEW/BROWSE or ALT+R or click the ‘Browse View’ icon on the toolbar. The data is now presented as a list of records, sorted by the main marker and showing one or two other data fields. Click on another record (i.e. a different line in this view) then repeat the previous action, or simply double click, e.g. ALT+R to return to the single record view. With the focus again on the **MyLanguage.Lex** select VIEW/BROWSE FIELDS... This opens a box that allows you to add, remove and re-arrange the markers in the browse view. Try removing the ‘hm’ marker and adding the ‘ge’ marker instead. You will see the changes in the browse view. (Note that this does not change the sort order: even if you place the ‘ge’ marker first the lexicon is sorted by the ‘lx’ marker which is the default.). These actions work in all database types and provide a good way to move around the data – try the same thing with the focus on **MyText.txt**.

13. Sorting by other than the default marker (e.g. by English gloss)

13.1 With the focus again on the **MyLanguage.Lex** select VIEW/BROWSE FIELDS... again. Add the ‘ge’ marker again and this time it makes sense to place it at the top of the list as we are also going to sort by ‘ge’. Now select DATABASE/SORTING. This opens another, similar, box. Remove any existing markers from ‘Sort Fields’ and add ‘ge’. The browse view now gives you a useful view of your lexicon sorted by the English gloss field. Note that now the lexicon window doesn’t necessarily indicate the same number of records as before even though it shows the same lexicon. If one lexeme has more than one gloss, for example, it will now show up in **MyLanguage.Lex** more than once. Only the lexicon sorted by the lexeme marker (in this case ‘lx’) accurately reflects the number of discrete lexemes.

13.2 In Toolbox the technique just described (i.e. using DATABASE/SORTING) becomes redundant because one can sort the records in any window which is in Browse View simply by clicking on the column heading (i.e. a marker name) – very convenient. (Also, by holding CTRL while clicking you can sort any column by the right hand end of the string). Open example project **MyProject_3.prj** in Toolbox to test this. Note however that when the Lexicon window is not sorted by the \lx marker, jumping will open a new window. (In other words the jump target only works if the target window is sorted by a marker with the same language encoding as the word being jumped from).

14. Other Shoebox functions

14.1 Apart from sorting the data, one often needs to filter it, to search for particular types of information (Shoebox is particularly rich in this department having ‘Search’, ‘Find’ and ‘Jump’ commands, all subtly different), and to create word lists and concordances according to specific criteria. Some of these techniques may be touched upon in the sessions, but as they are not especially relevant to the production of the corpus we shall not deal with them here.

14.2 Another area where Toolbox goes further than Shoebox is its ability to associate media files (e.g. sound and graphics files) with a marker in a record and to cause them to play (usually using a helper application already on the computer such as a browser or media player) by pressing the F4 key. Navigate to the record ‘Basibasi’ in the lexicon window of **MyProject_3.prj** for an example of this. This record actually has two media files (each with marker \med) – a photo followed by a sound file; which file is played depends on where the cursor is – basically the first file found after the cursor position is played. The content of the marker is the path to the file – both absolute and relative paths are supported (absolute used here).

14.3 One feature of Toolbox that might be easily overlooked is its ‘Lock Project’ option that allows one to progressively limit the functionality of the program. One reason to use this is to provide a platform for ‘naïve’ users. At the extreme (level 10) Toolbox functions a mere viewer; this could be a good way to distribute your data to others. See the ‘Advanced Consultant Features - Reduced Menu Set and File viewer option’ topics in Toolbox help. Open **MyProject_4.prj** in Toolbox for an example of level 10 locking: this shows just the lexicon with gloss and semantic domain fields added in Browse View. Most of the advanced functions are missing or limited, e.g. one can apply an existing filter but not make a new one. Nor can one add, delete or edit a record, but one can still sort the records by any of the three columns in BrowseView.

15. Exporting the lexicon as a dictionary or wordlist via MDF – Word, PDF and Lexique Pro formats

15.1 With the focus on the **MyLanguage**.Lex window select FILE/EXPORT. A box will appear with, among other options, the MDF RTF process. Select that, make sure the box for ‘Automatically open document in word processor’ is checked and click ‘OK’. At the next box make sure that ‘Audience’ and ‘Format’ are set to English and Dictionary:Diglot respectively and that ‘Exported File Type’ is RTF, edit the title as appropriate and click ‘OK’. At the next box choose the name and location for your exported dictionary. When you click ‘Save’ the export process will try to begin. Several things are necessary for success. Firstly the required MDF settings files must be available. This means that if you are running your project from a folder other than the default ‘My Shoebox Settings’ (as we are on with this example project) you must have first copied the relevant files to that folder (the easiest way is to copy any file beginning with ‘MDF’). Secondly you will need to have ‘Word’ available for completing the process. Thirdly you will probably have to explicitly enable macros, particularly in recent versions (eg TOOLS/OPTIONS/SECURITY/MACRO SECURITY... in Word 2002). Assuming that you have the above, Shoebox will first produce the basic .rtf file and give you a reminder that you have to run a macro (supplied) in Word to complete the process. Word should then open with the rtf file – select (within Word) TOOLS/MACRO/MACROS... then find ‘TemplateProject.FinishExportingFromShoebox.MAIN’ (at least this is the procedure on Word 2002 – the details tend to change with the version!). This will run the macro to complete the formatting and save in .doc. format to the same location. You can test the above using, say **MyProject_3.prj**.

15.2 Repeat the process above but this time choose Gloss index (finderlist): English under ‘Format’ and edit the title appropriately. This is the other main ‘view’ of the data that the MDF process provides.

15.3 Note that Toolbox can produce MDF dictionaries using Unicode but you ‘must set all language encodings to Unicode. In particular, you must set "vernacular" and "national" to Unicode even if you don't use them’. Future plans for Toolbox include a version of MDF (to be called TDF) ‘to handle dictionaries with multiple scripts and to make it easier to customize’.

15.4 You can include pictures for inclusion in the dictionary by using the \pc marker that is part of the MDF set. You have to provide not only the path but also the prefix ‘.G.’, the size details (in inches), and the extension for the file type – again see the record ‘Basibasi’ for an example of the format.

15.5 You can, and probably will, use filters to limit the lexicon entries that get through to the dictionary. Also you may wish to exclude certain markers from the export process – this possibility is found under ‘Options’ on the ‘Multi-Dictionary Formatter’ dialog box during the export process.

15.6 If you have Adobe Acrobat, you will probably want to convert your Word .doc files to .pdf format for distribution. This can be a lengthy process. See **Dictionary.pdf** for the pdf version of an unfiltered, unlimited (i.e. all markers) MDF dictionary produced from MyProject_3.prj.

15.7 Lexique Pro is a new program from SIL that allows one to produce an interactive dictionary from a Shoebox/Toolbox lexicon. It is free to use and uses the MDF marker set, so it seems it would be simple to produce an electronic version of the dictionary, or rather the whole lexicon, which can be searched, filtered and sorted. Photos can be readily attached to records, and there are additional options for providing supplementary information. Text and web pages can be exported from the finished file and the file itself can be distributed with a setup program to make it easy for users to install and use. One can encrypt the contents so that only the markers chosen are displayed and that the source file itself is protected – this makes it straightforward to export an entire Lexicon, even one containing sensitive information in ‘private’ markers. On the face of it this would seem a good alternative to distributing ones work as a read-only Toolbox file (as described in 14.3). See www.sil.org

16. Refining parsing – considerations for dictionary

16.1 Open example project **ParsingTest.prj** for examples and a brief discussion of some techniques for refining the way the lexicon is parsed when interlinearizing. There is a useful discussion on parsing provided by Shoebox in ShParse.pdf – a copy is in the ‘Shoebox Examples’ folder – and in fact our project builds on and repeats some of the information found there.

16.2 The use of alternate and underlying fields can help to organize the data in logical ways but there are some implications for the MDF dictionary process. For example if you place an item only in an alternate field – in order to list it as variation of another entry and to return that other entry when interlinearizing - it won't show up in the dictionary. If on the other hand it has its own lexical entry, but with an underlying field attached that refers to another lexical entry or combination of entries then it will show in the dictionary even though when parsing it will be replaced by the other entry/ies. Examples 1 and 2.

16.3 A good application for alternate fields is for defining rules for returning reduplication parses. For this a special kind of lexical entry is required – see ‘dupCV-’. Examples 3 and 4. You may not wish a lexical entry such as this to print in the dictionary – how can you exclude it from the MDF process while still keeping it in the lexicon? One simple answer is to use a special filter that targets such entries. If you try the ‘exclude’ and the ‘exclusions only’ filters, available when the lexicon has the focus, you can see how this works on the marker ‘exc’ (which was made up by me for the purpose and is not part of the MDF set). Note that the way this filter is written the marker does not have to actually contain anything, it merely has to be present (so the content ‘yes’ is purely informational). When you export to MDF you simply have to select the appropriate filter in the ‘Filter records:’ field. (Filtering by certain markers is also a good way to quickly produce a thematic dictionary, e.g. one covering only animals, perhaps as a community resource).

16.4 The discussion given on compound roots in ShParse.pdf stops short of considering what happens to the parse process when you put not only the roots but the compound in the lexicon (as you might wish to do in order to provide that information to the dictionary). The ParsingTest.prj therefore tests various possibilities for returning different parse results under such conditions through the use of different combinations of underlying fields. It is quite straightforward: entering a compound as a lexical entry will cause the parser to return that compound rather than the separate roots; adding an underlying field to that entry (the compound) which contains the roots (separated by a space) will cause the parser to return the roots; adding a further underlying field containing the compound will cause the parser to return an ambiguity box with the alternatives shown. Examples on ‘white bird’ and ‘whitebird’, 5 to 8.

16.5 The use of forced parsing – where specific instances of homonyms can be referenced as part of an underlying field entry – can greatly assist in avoiding bad parses. They only apply of course to the specific instance and therefore a word formula – which operates at a more general level – can be a better choice when you are faced with many bad parses of the same type (e.g. wrong kind of suffix offered). On the other hand for the same reason they avoid the danger of over-generalisation that an inadequate word formula can lead to. Example 9.

16.6 The use of word formulas can assist in the speed of parsing – providing you can identify what an appropriate formula is in the first place. Keep in mind that an invalid formula can prevent potential ‘good’ parses from being offered. The last example shows what happens when you have no word formulas or they are not enabled - you have maximum flexibility to make good and, as here, bad parses. Examples 10 to 15.

17. Exporting texts to Transcriber from Shoebox using Econv

17.1 To allow Transcriber to open a Shoebox text file – whether interlinearized or not – it must first be converted to a .trs file that Transcriber can read. The tool for this is Econv. There is a Dobes guide document for its use (econv.pdf) but my use of it has been slightly different, as given below. (The chief difference is that I have not used the recommended ‘econv.prj’ and ‘econv.typ’ files - actually I could not locate them). My method requires a little post-editing in a text editor as indicated, but it does work satisfactorily.

17.2 Start Econv. Click ‘Shoebox == > Transcriber’. Navigate to your Shoebox text file, say **MyText.txt** (i.e. one that has already been opened and associated with a database type in Shoebox). A ‘Shoebox Options’ box will open. After ‘Name of speaker is’ select ‘Not contained. Use’. In the next box either leave the default ‘Speaker 1’ or, insert the identity of the speaker (if there is more than one, then use the identity of the main speaker – means less to edit later). After ‘Transcription is contained in tier’ change \trs to \t (if you have been following our conventions above; if not then change it to whatever marker name you have chosen, e.g. \tx, for the transcription tier. Click ‘OK’. Econv writes the file and notifies you of the name (same as the input but with .trs extension) and location. You can exit Econv.

17.3 If you try to open this file in Transcriber (at least in version 1.4.2 for Windows) you will probably get a ‘Validation impossible – no DTD found in document’. My solution to this failure is to edit the new file (e.g. **MyText.trs**). Open it in a text editor and save it as text to a slightly different name (e.g. **MyText_a.trs**) to preserve the original in case you need to go back. (Note, if you use WordPad the program will add a superfluous .txt extension which you will have to remove later in Explorer, e.g. change **MyText_a.trs.txt** back to **MyText_a.trs**).

17.4 Place the cursor at the end of the first line (which will be something like `<?xml version="1.0" encoding="UTF-8"?>`) Press enter a couple of times to make some space. Insert the following text in the gap (i.e. on a new line):

```
<!DOCTYPE Trans SYSTEM "trans-13.dtd">
```

```
<Trans scribe="(unknown)" audio_filename="AUDIO_FILENAME" version="1" version_date="040421">
```

where AUDIO_FILENAME is the name of the audio (.wav) file which you intend to link to the text. This name can be typed without the .wav extension, e.g. **MyText** rather than **MyText.wav**. The next line begins `<Trans><Speakers><Speaker....etc.` Delete the `<Trans>` tag, i.e. so that it reads `<Speakers><Speaker....etc.` Save and close the file. You should now be able to open it in Transcriber. Take a look, in a text editor to the example files **MyText.tr**s and **MyText_a.tr**s (located in the 'data' folder) to see how the changes look. Note that the file opens with the intonation units you have determined through your punctuation conventions and the cct (constant change table), however they have been arbitrarily assigned to one-second 'chunks'. The task that remains is just to re-align the borders of the chunks with the correct portions of the sound file and to make sure that the speaker identities are correct – this step has been completed in **MyText_b.tr**s.

18. Processing texts with audio in Transcriber - general notes

18.1 Transcriber works only with text and audio files – not video. This makes it more limited for our purposes than Elan as a transcribing tool. However this limitation is part of its key attraction: Transcriber is very easy to use and to teach to language consultants precisely because of its simplicity when used purely for marking up audio files with time-aligned text. Transcriber was not developed just for this task however and some of its functions are definitely incompatible with Elan. I repeat here the following warnings from the econv.pdf guide:

18.2 'Use exactly one section. In Transcriber you can define as many units (sections) as you want. In Shoebox, there is no concept of a section. Please use exactly one section.'

18.3 'Do not use "events" in Transcriber. Please do not use events. You should use a dedicated tier in shoebox when you want to denote something.'

18.4 'How to denote silence. Please use a turn without a speaker to denote silence. Please do not enter any text in a turn without a speaker.'

18.5 In addition I would warn against the use of overlapping speech in Transcriber, since Elan deals with it in a totally different way (because it assigns a new set of tiers for each speaker, unlike Transcriber). If you need to use overlapping speech it would be better to do that in Elan at a later stage. (A personal opinion).

18.6 I don't find it necessary to play around with the various menu options, e.g. playback mode, with the exception of suppressing the interlinear text, as described in 19.2., below.

19. Using Transcriber with an existing .trs file

19.1 Start Transcriber. It will by default begin with an 'Open transcription or audio file' box. Navigate to your .trs file (e.g. **MyText_a.tr**s) and click 'Open'. When you are opening a .trs file of the kind just made you may have to open both this file – the transcription and then the related audio file. This occurs if the program can't find a file of the name specified (by you) in the .trs (see 17.4 above) in the same folder as the .trs. Once you have opened the appropriate audio file and then saved (as another .trs) you should be able to open it again in future without specifying the audio file. For trouble free use, however I recommend keeping the audio and .trs files in the same folder.

19.2 If the file you open has not been interlinearized you will simply see the text, broken down into records (in two ways: as a series of editable lines and as the content for the blocks running along under the audio signal). If on the other hand you have already interlinearized the text (or simply added any other tiers in Shoebox) then this information will also display in a very unhelpful manner. As described above in 11.2.2, to suppress (not destroy) these unwanted lines select OPTIONS/EVENTS/EVENTS DISPLAY – this opens a 'Configure Events' panel. Remove the contents, { %s }, of the 'Comment:' box.

19.3 Playback and navigating. Simple playback is achieved by clicking the play arrow. However you will use the following shortcuts more often. Press TAB. If a particular part of the audio signal is highlighted (achieved by clicking in one or more of the blocks under the signal; the selection corresponds of course to the record/s selected) then just that selection will play. If no such selection has been made then playback will start at the position of the cursor (over the signal) and continue until TAB is pressed again. Press SHIFT+TAB. In this case just the record that is highlighted in the lines above the signal will play back. To move to the next or previous record you can either click on them or, more conveniently, use the UP ARROW and DOWN ARROW.

19.4 Adjusting breakpoints. There are two basic ways in which to move the breakpoints. Position the mouse over a breakpoint and LEFT-CLICK while pressing CTRL. This allows you to move just this breakpoint, back and forth for fine tuning without affecting other breakpoints – good for final editing. Position the mouse over a breakpoint and LEFT-CLICK while pressing CTRL and SHIFT. This has the effect of moving all the breakpoints to right or left when the one you are editing gets close to them – very useful in the early stages when you are basically working from left to right through the signal.

19.5 Deleting and adding breakpoints. If you need to delete a breakpoint then click in the block to the right of the breakpoint and press SHIFT+BACKSPACE. To insert a new breakpoint click on the signal roughly where you want the breakpoint to be then press ENTER (you can then fine-tune its position as in 19.4.) Warning: remember that each time you delete or add a breakpoint to an existing file you are altering the record count of that file!

19.6 Editing text. Enter and edit text by simply clicking – or moving with the up/down arrow keys - on the relevant line. Usual copy and paste techniques work.

19.7 Adding/deleting turns and speakers. Depending on how you imported your file you may already have a principal speaker defined. To add a speaker or change details just click on any blue ‘turn’ button – e.g. the one at the top of the text lines. To make a new turn highlight the first line of the new turn and press CTRL+T. A box will open allowing you to choose a speaker, or add a new one. Click ‘OK’ and the appropriate turn button will appear above the line. To delete a turn button click on it then click ‘Destroy’. These speakers will be written into any file you export to Elan and thus to Shoebox.

19.8 Changing the resolution. To see more or less of the signal in the window, RIGHT CLICK when the mouse is over the lower pane then choose a setting under RESOLUTION.

20. Starting a new transcription in Transcriber

20.1 Instead of opening a .trs file you will be simply opening the associated .wav file. Select FILE/NEW TRANS and you will be automatically prompted to open an audio file.

20.2 The general editing tasks in this case are exactly the same as in previous case. The only difference is that you must first create the records in which to add the text and then, of course transcribe the audio into these records. A quick way to get started is, with the audio playing, simply hit the ‘Enter’ key every time you think you discern a natural intonation unit boundary. You might make say five to ten such records and then go back and refine the boundaries adding the text in the lines above as you go. It becomes a very intuitive process after a short time which hardly merits further description. It can be useful to start at a rather broad resolution and change to a finer one as it becomes clearer where the boundaries should be.

20.3 Select file/save to save the work as .trs file. Next time you open this file both audio and text will load together, as with any other existing .trs file, unless you have since moved or renamed the .wav file.

20.4 It is often useful to be able to go back to where one left off when opening Transcriber (particularly as it doesn’t seem to be possible to associate a .trs file with the program such that it will open just by double clicking on the icon). The way to achieve that is to choose OPTIONS/SAVE CONFIGURATION before closing the program.

21. Importing texts from Transcriber to Shoebox using Econv

21.1 This is still possible but has, in my view, been superseded by the new ability of Elan to import Transcriber files and export/import Shoebox files. See the appendix topic below for details.

Appendices

1 Notes on Toolbox vs Shoebox

Some of the salient differences between the two programs have been discussed in context, above. There are few other issues that are worth knowing about before making a choice on which program to use. These may be summarized under the pros and cons of the newer program, Toolbox.

For: Toolbox is free whereas Shoebox 5 has a small license fee. The issue here is not so much the expense as the ease with which one can use the program. For example a University may object to a class in Shoebox being taught because it is concerned about license infringement issues; this is not an issue with Toolbox. Toolbox supports Unicode. Multiple instances of Toolbox can be run on one machine and 'network safety' is built in. Toolbox can compare two versions of the same file. Toolbox has the ability to do an 'external jump' to some other programs, and in general the built in jumping capabilities are more extensive.

Against: Toolbox is not available for Macs! Toolbox has no guaranteed support (on the other hand since it was built on Shoebox, and since Shoebox will not be developed further, it is debatable whether this a serious issue – in fact the signs so far are that Toolbox is being steadily improved). Toolbox projects using Unicode language settings are not safely readable in Shoebox – not a problem if you don't have to 'go back' to Shoebox.

In general Toolbox is so similar to Shoebox that the transition is easy. Most projects (but not Unicode ones as mentioned above) can be opened without problem in either program at this stage of Toolbox's evolution – the (only?) difference being the availability or otherwise of Toolbox features.

2 Moving data between Transcriber, Elan and Shoebox/Toolbox

We use Transcriber for the basic marking up of audio/video files (rather than Elan, mainly because it is easier to train field assistants to use it) and we use 'Shoebox/Toolbox' for lexicon and interlinear work. However we also use Elan – principally as a medium for viewing finished work, but also as an editing tool. We therefore need to be able to move the data effectively from one program to the other. Formerly we used Econv for some of this work and relied on 'Find and Replace' techniques within 'Word' for the rest. With Elan 2.4 and 2.4.1 however there is a very effective 'import from Transcriber' facility, and with this we can cut out the need for Econv. We still need to do some editing in Word, but once this is done we can move back and forth between Shoebox and Elan as much as we like without difficulty

While our method (which is based on Elan 2.4.1) is no doubt idiosyncratic, being adapted to our particular needs and preferences, it may be that other teams will be able to make use of it, with or without further adaptation. What follows is an overview, then a step by step account of the workflow we normally use. The files referred to are all in the same 'data' folder as used for the rest of this guide.

Overview:

1. We start our work in Transcriber and then import the data to Elan. This is a one way process: we don't need to go back to Transcriber (of course we complete as much as possible of the transcription in Transcriber before moving on).
2. We save this basic data as an Elan .eaf file and then export the same data from Elan to Shoebox format, making a .txt file as a result. (We might make some changes in Elan before saving and exporting, e.g. dealing with overlapping speech events).
3. We open this .txt file in Word and make a few changes and additions to suit the needs of our particular Shoebox setup, using 'Find and Replace'.
4. We open the amended file in Shoebox. Note that in order to open the file in Shoebox we must have an appropriate 'Database Type' file - see the detailed instructions that follow.
5. From this point we can edit the file in Shoebox, then re-import it into Elan and then export it again, without suffering changes, other than the ones we intend. So for instance, we might do some of the interlinearization in Shoebox, then make some changes to time boundaries in Elan, then reinterlinearize in Shoebox again, and so on.

Detailed instructions:

1. After saving a finished Transcriber file (.trs extension), open Elan and select File/Import/Transcriber file. We then right click on the tier called 'Sections' then click Delete 'Sections'. This will also delete 'Turns' for which 'Sections' is the parent. (We do this because we find no use for this Transcriber specific information in either Elan or Shoebox, but you may keep these tiers if you wish). Ref: [MyText_b.trs](#)
2. Select File/Save to make an Elan .eaf version of this file. Select File/Export As/Shoebox File to make a Shoebox .txt version: A dialog box will open. Select the tiers you want to export (if you followed our procedure there will be only one for each speaker). Then click the radio button next to 'Define field markers...' button so that is no longer greyed out, then click the button itself. This brings up another dialog box. You can either fill in each marker and its properties yourself (referring to the Elan manual), or – a lot easier at this stage – click on 'Load Markers...'. Elan will probably by default direct you to a folder called 'Elan_data', but since you haven't set up your own marker file already, this will be empty. Navigate instead to the 'Some Shoebox Settings' folder where you will find 'marker_file.mkr' (this file I made especially for this exercise and so is another file to make sure you copy to your 'My Shoebox Settings' folder, if you later want to use our setup). Choose this file then click 'Select' then 'Close', then 'OK'. – you will be prompted as normal for a file name and location to save. Refs: [MyText_b.eaf](#); [MyText_b.txt](#)
3. Edit in Word:
 - 3.1. Open the .txt file in Word, accepting the 'Windows (Default)' option if your version of Word suggests it. Immediately save (as text) to a different file name and work on this new file (to preserve the original in case of...) Refs: [MyText_b.txt](#); [MyText_c.txt](#) (this file is the result of the next steps, 3. -)
 - 3.2. Change all the markers provided for the text from the speaker name to 't' (which is our Text marker). For example in the example file there is one speaker, 'Speaker1' (these were default speaker identities provided by Transcriber, but we could, and normally would have changed these in Transcriber to something more meaningful) and so Elan has provided a text marker in the .txt file called '\Speaker1' – this is the marker we have to change. To change '\Speaker1' select Edit/Replace (Ctrl +H). In 'Find what' type \Speaker1 , in 'Replace with' type \t . Click 'Replace All' .(Do likewise for any other speaker markers you might have e.g. \Speaker2, \Joe, etc).
 - 3.3. Add markers for the free translation field, 'f' before each 'ELANBegin' marker: Select Edit/Replace (Ctrl +H). In 'Find what' type \ELANBegin , in 'Replace with' type \f ^p\ELANBegin . Click 'Replace All' . (The ^p combination inserts a paragraph break between the markers).
 - 3.4. Convert the marker supplied for a record number field, 'block' to 'ref' plus a meaningful prefix (if desired): Select Edit/Replace (Ctrl +H). In 'Find what' type \block (include a space after '\block'), in 'Replace with' type \ref MyText_ . Click 'Replace All' . (Note that 'MyText_' is the name of the file, i.e. it is merely an identifier attached to the record number - we find it useful but it is optional; you would substitute the filename or other useful string to suit your purposes).
 - 3.5. Save the file (as text) and close. Ref: [MyText_c.txt](#)
4. This is a 'finished .txt file which can now be opened in Shoebox or re-imported into Elan:
 - 4.1. Shoebox: Open this file in any Shoebox project. The only essential prerequisite is that both the project and the appropriate database type file are in the same folder (normally 'My Shoebox Settings'. To try this case open MyProject_5.prj, select File/Open then locate MyText_c.txt. The file will open through its association with the ElanExport database type (filename ElanExportText.typ located in the Some Shoebox Settings folder – make sure you copy the files in this folder to your own My Shoebox Settings folder if you want to use this set up). Refs: [MyProject_5.prj](#); [MyProject_6.prj](#) (has MyText_c.txt already open, and set up with more informative browse fields, etc).
 - 4.1.1. Normally your project would –as our examples here - also contain an appropriate Lexicon. This would be 'connected' via the database type properties to any text opened in this way; the example projects all exhibit such features. If you wish to learn more about these connections, click on the .txt file window in a project, then select Database/Properties to bring up the dialog box; the 'Jump Path' and 'Interlinear' tabs contain the relevant information for the connections made between Text files and Lexicon file/s).

- 4.1.2. Once a file has been interlinearized in Shoebox a number of new markers are added to each record. These markers and their corresponding data will be read correctly in Elan provided the correct 'Field Marker File' is loaded when importing it – see the next section. To make the changes clear we have saved a version of MyText_c.txt as MyText_d.txt, then started to interlinearize it. Ref: [MyProject_7.prj](#) (has MyText_d.txt open, as well as MyText_c.txt for comparison).
- 4.2. Elan: Rather than being 'opened', a Shoebox .txt file must be 'imported' into Elan. This must be done each time a new or revised .Shoebox file is brought into Elan, but each time it can then be saved as a native .eaf file which can then be simply 'opened' in Elan, for viewing and/or editing and subsequent re-exporting to Shoebox (if desired).
- 4.2.1. In Elan select File/Import/Shoebox File, a dialog box will open. For 'Shoebox file' navigate to the desired .txt file, e.g. MyText_d.txt.
- 4.2.2. The next step is similar to 2. above: Click the radio button next to 'Set field markers' button so that is no longer greyed out, then click the button itself This brings up another dialog box. You can either fill in each marker and its properties yourself (referring to the Elan manual), or – as previously described – click on 'Load Markers', navigate to the 'Some Shoebox Settings' folder (will already be the default folder if you followed step 2.), choose 'marker_file.mkr', then click 'Select' then 'Close', then 'OK'.
- 4.2.3. Elan will now import the file (this may take a little time) with the associated .wav or .mpeg files/s. If for some reason Elan doesn't find these media files it is not hard to restore them: Select Edit/Linked Files then click 'Add' then navigate to the audio/video files you want - if they are in the same folder this will be very easy You will have to choose each file in a separate step; make sure the right file is set under 'Master Media'. Then click 'Apply'. The files will load.
- 4.2.4. At this stage the interface may appear a bit messy because the tiers will be unsorted. To remedy this right-click over the tier labels (in the bottom left) then select Sort Tiers/Sort by Hierarchy (my favourite, or choose another option of course).
- 4.2.5. It may be that you don't see all of the tiers you expect; this sometimes happens e.g. after interlinearizing. In that case, right-click over the tier labels again and select Visible Tiers/Show All.
- 4.2.6. Now save this set up as an Elan .eaf file so that you can open it easier next time, and re-export it to Shoebox if desired – if you simply choose File/Save an .eaf file with the same name will be made (but this fact will not be reflected in the title bar at the top which will still say 'xxx.txt'), alternatively choose File/Save As and give the file a different name (in this case the title bar name will update). In both cases of course the extension .eaf will be added automatically. Ref: [MyText_d.eaf](#) (the Elan version of MyText_d.txt).
- 4.2.7. You can edit the .eaf file using the normal Elan methods, e.g. shifting time boundaries, adding or deleting annotations to tiers, etc. Ref: [MyText_e.eaf](#) (We have added a free translation to the first segment and shifted the first boundary a little to the right).
- 4.2.8. When you are done, save the .eaf file again, then export it to Shoebox again, exactly as in step 2. (you will probably want to export all the markers, which is the default). Open in Shoebox to verify that the changes have been made and that no corruption of the file has taken place. In our setup the re-exported file has a slightly different layout from the original, but this only seems to occur once and isn't a serious error (there may well be ways around this that I haven't yet discovered). The other curious effect is that another '\block' marker and record is added to most records (not the last one for some reason). The quick and dirty way to solve this 'problem' is to use Toolbox's 'Hide Fields' command: with the text window highlighted, select VIEW/FIELDS TO HIDE then insert 'block' into the 'Hide Fields' box then 'OK' - the offending field will disappear, (it can be shown again with the VIEW/HIDE FIELDS toggle). Ref: [MyProject_8.prj](#) (like MyProject_7.prj but includes MyText_e.txt, the Shoebox-exported version of MyText_e.eaf).
- 4.2.9. As far as I have been able to ascertain, there is no limit to the number of times one can move back and forth between Elan and Shoebox in this way, although there may well be limitations on the kinds of editing one can do which I have not yet encountered. Ref: [MyProject_9.prj](#) (compare MyText_e.txt with MyText_f.txt: the latter file was produced by importing MyText_e.txt to Elan, making minor modifications, saving as MyText_f.eaf then exporting again to Shoebox – no noticeable differences apart from the intended edit to the free translation in record MyText_002).