Exporting ..................................................................................................................................... 167
  Export: An Overview (File menu) .......................................................................................... 167
  Records to include .................................................................................................................. 168
  Export Process Type dialog box .......................................................................................... 169
  SF Export Process Properties dialog box ............................................................................ 170
  RTF Export Process Properties dialog box ........................................................................ 172
  Page Setup, Document tab (RTF and MDF export processes) ........................................... 176
  Paper ..................................................................................................................................... 176
  Margins ................................................................................................................................. 176
  Page Setup, Section tab (RTF and MDF export processes) ............................................... 177
  Header and footer ................................................................................................................ 177
  Columns ............................................................................................................................... 177
  Special symbols used in exporting data ............................................................................. 177
  Export Current Window dialog box .................................................................................... 179
  Select Fields dialog box (Exporting) .................................................................................. 180
  Exported File dialog box ..................................................................................................... 180
  Multi-Dictionary Formatter (MDF): An Overview .............................................................. 181
  Using Unicode with MDF ..................................................................................................... 181
  Multi-Dictionary Formatter (MDF) dialog box .................................................................. 181
  Multi-Dictionary Formatter (MDF) Audience Setup ......................................................... 183
  Multi-Dictionary Formatter (MDF) Options ....................................................................... 184
  MDF Reminder dialog box .................................................................................................. 186
  Direct Printing ..................................................................................................................... 186
  Overview ............................................................................................................................... 186
  Print (File menu) .................................................................................................................. 186
  Page Setup (File menu) ....................................................................................................... 187
  Printing codes for the header and footer .......................................................................... 188
  Print Preview (File menu) ................................................................................................... 188

Interlinear.......................................................................................................................... 190
  Interlinear topics .................................................................................................................. 190
  Interlinearizing: An Overview ........................................................................................... 190
  Text: Organizing and Preparing ......................................................................................... 191
  Interlinear Setup .................................................................................................................. 196
  Interlinear tab (Database Type Properties) ........................................................................ 197
  Quick Setup Interlinear Text Markers ................................................................................ 197
  Quick Setup Lexicon Markers (Interlinear) ......................................................................... 198
  Select Process dialog box (Interlinear) .............................................................................. 198
  Parse dialog box (Interlinear) ............................................................................................ 199
<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Formulas dialog box</td>
<td>202</td>
</tr>
<tr>
<td>Lookup dialog box (Interlinear)</td>
<td>202</td>
</tr>
<tr>
<td>Rearrange dialog box (Interlinear)</td>
<td>204</td>
</tr>
<tr>
<td>Generate dialog box (Interlinear)</td>
<td>204</td>
</tr>
<tr>
<td>Given dialog box (Interlinear)</td>
<td>205</td>
</tr>
<tr>
<td>Lexicons for Interlinear dialog box</td>
<td>206</td>
</tr>
<tr>
<td>Punctuation issues</td>
<td>206</td>
</tr>
<tr>
<td>Interlinearize (Tools menu)</td>
<td>207</td>
</tr>
<tr>
<td>Interlinearize (show all parses) (Tools menu)</td>
<td>207</td>
</tr>
<tr>
<td>Parsing: An Overview</td>
<td>209</td>
</tr>
<tr>
<td>Parsing: Roots and Affixes</td>
<td>209</td>
</tr>
<tr>
<td>Infixed (Interlinear)</td>
<td>210</td>
</tr>
<tr>
<td>Compound Roots</td>
<td>210</td>
</tr>
<tr>
<td>Reduplication in Interlinear Text</td>
<td>211</td>
</tr>
<tr>
<td>Reduplication (More)</td>
<td>212</td>
</tr>
<tr>
<td>Alternate Forms of Morphemes</td>
<td>213</td>
</tr>
<tr>
<td>Underlying Forms (Interlinear)</td>
<td>216</td>
</tr>
<tr>
<td>Underlying forms (More)</td>
<td>218</td>
</tr>
<tr>
<td>Ambiguity: Forced Values</td>
<td>221</td>
</tr>
<tr>
<td>Ambiguity Selection dialog box</td>
<td>222</td>
</tr>
<tr>
<td>Parsing: Preventing Invalid Parses</td>
<td>222</td>
</tr>
<tr>
<td>Morphophonemics</td>
<td>225</td>
</tr>
<tr>
<td>Context Sensitivity Preventing Incorrect Parses</td>
<td>226</td>
</tr>
<tr>
<td>Ambiguity: Recognizing</td>
<td>230</td>
</tr>
<tr>
<td>Word Formulas (Interlinear)</td>
<td>230</td>
</tr>
<tr>
<td>Examples of valid and invalid parses</td>
<td>231</td>
</tr>
<tr>
<td>Examples of word formulas</td>
<td>231</td>
</tr>
<tr>
<td>Enabling word formulas in the Parse dialog box:</td>
<td>232</td>
</tr>
<tr>
<td>Word Formulas: Selecting valid parses</td>
<td>233</td>
</tr>
<tr>
<td>Word Formulas: Symbols</td>
<td>233</td>
</tr>
<tr>
<td>Word Formulas: Patterns</td>
<td>234</td>
</tr>
<tr>
<td>Word Formulas: Recursion</td>
<td>236</td>
</tr>
<tr>
<td>Word Formulas: Troubleshooting</td>
<td>237</td>
</tr>
<tr>
<td><strong>Spelling and Interlinear Check (Checks menu)</strong></td>
<td>238</td>
</tr>
<tr>
<td>To check spelling</td>
<td>238</td>
</tr>
</tbody>
</table>
Toolbox Reference Manual

Introduction

This file is a reference manual for the Field Linguist's Toolbox program. It contains details on how to use all the tools in the Toolbox. It is not an introduction to Toolbox. For an introduction to Toolbox read the self-training document.

To find a topic in this manual, first look at the Table of Contents. If you don't see it there, try using Find.

Revision History of This Manual

This file is a growing document. New material is added as it becomes available. New sections are listed in the revision history below.

Jan 2020 Added MDF marker material that was previously available in Toolbox files.

Dec 2019 The original version of this manual was produced from the information in the Toolbox help file.

General Information

Markers

Markers: An Overview

Toolbox uses a system called Standard Format as a means of labeling fields within a record. A Standard Format marker indicates the beginning of a field. It is specified by a backslash followed by a short mnemonic label. Toolbox expects all fields to begin a new line. When a line in a standard format database begins with a backslash (\), this signifies the start of a new field. The marker is separated from the field contents by a return or a space. For example:

<table>
<thead>
<tr>
<th>Marker</th>
<th>Field Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>\lx</td>
<td>adar</td>
</tr>
<tr>
<td>\ps</td>
<td>n</td>
</tr>
<tr>
<td>\ge</td>
<td>statue</td>
</tr>
<tr>
<td>\sg</td>
<td>adarke</td>
</tr>
</tbody>
</table>
In a Toolbox database window, the markers are edited and displayed in the marker pane. The marker pane only displays the fields actually present in the current record. The entire list of available markers for the database is displayed in the drop-down box when you insert or change a field in a record and in the Database Properties dialog box.

Markers in Toolbox have options at two levels: the record level and the database level.

**Record level options**

**Modifications to the record**

*Editing options*
- Insert a field from the list of available fields (markers)
- Delete a field (or a marker) from a record
- Change a field's marker to a different existing marker

*Viewing options*
- View just the markers
- View just the descriptive field names
- View both markers and field names
- View the marker hierarchy
- Change the size of the marker pane

**Database level options**

**Modifications to the list of all available markers**
- Add a new marker
- Delete an existing marker
- Change the marker to a different marker throughout the database

*Editing Aids*
- Template for new record
- Following marker to be automatically inserted

*Consistency Checking*
- Range set (to ensure consistency of contents for fields with a Limited set of valid data, like the part of speech)
- Data properties (for checking references, allowing empty field, etc.)

**Marker Description**
- Descriptive field name
- Full description

**Language and font**
- Language encoding for the field (connecting to sort order, direction, font, etc.)
- Specify a different font or font properties (bold, color, size, etc.) than the standard font specified for the language encoding

**Other properties**
- Position in the hierarchy
- If exported as RTF, specify as character or paragraph style
**Insert Field (Edit menu)**

The Insert Field command will cause a new field to be created following the current field.

**To insert a new field**

Do the following:

1. Position the cursor anywhere in the field before the place you want the new field inserted.
2. From the Edit menu, choose Insert Field or use the shortcut keys, CTRL+E. A combo box will appear in the marker pane with the list already dropped.
3. Choose the field marker you wish from the list or create a new one if necessary.
4. Press space or enter to terminate the choice.

Note: Markers can be created "manually" by the following process. (It is almost as quick as using the shortcut.)

1. Place the cursor at the very end of the field preceding the point where you want the new field.
2. Press Enter.
3. Press backslash (\). At this point, the combo box will appear in the marker pane with the list dropped.
4. Choose the marker you wish from the list or create a new one if necessary.
5. Press space or enter to terminate the choice.

**Deleting a field or a marker from a record**

**To delete an entire field**

Choose one of the following:

A. Select the entire field by positioning the mouse cursor over the marker and double-clicking. Press Delete (or Cut if you wish to place the field elsewhere.)

B. Or, position the cursor at the end of the preceding field, hold down the Shift key, and down-arrow to the last line of the record you wish to delete. Press the End key to complete the selection of the last line. Press Delete or Cut.

C. Or, position the cursor at the end of the field and press backspace until the field contents are gone. Press backspace three times more until the marker itself is gone.

**To delete just a marker**

To delete just a marker from the beginning of a field, thus merging the contents of the field with the contents of the previous field, do the following:

1. In the data pane, position the insertion point at the beginning of the field with the marker you wish to delete.
2. Press backspace. A box will open with the marker highlighted.
3. Press backspace twice to delete the marker and the backslash.

Or, just cut and paste the contents, then select and delete the now empty marker.
Markers: Changing one in a Record

To change a marker in a record to a different marker, do the following:

1. In the data pane, position the insertion point at the beginning of the field with the marker you wish to change.
2. Press backspace. A box will open with the marker highlighted.
3. Type the desired marker or select one from the list. Press Enter. The new marker now appears in the marker pane of the database window.

Restricted characters

The following table includes a list of characters which should not be used.

<table>
<thead>
<tr>
<th>Location</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markers</td>
<td>\ { }</td>
</tr>
<tr>
<td></td>
<td>(backslash, curly brackets, vertical bar, comma)</td>
</tr>
<tr>
<td></td>
<td>Also, an underscore in the first position after the backslash is reserved for Toolbox's internal use.</td>
</tr>
<tr>
<td>Field names</td>
<td>\ { } , ;</td>
</tr>
<tr>
<td></td>
<td>(backslash, curly brackets, comma, semicolon)</td>
</tr>
<tr>
<td>Standard Format fields</td>
<td>Characters with codes below 32 (space), except newline</td>
</tr>
</tbody>
</table>

Note: If you are expecting to export database files in Rich Text Format or with the Multi-Dictionary Formatter option for use in Microsoft Word or some other software package, it would be wise to consider any additional characters which might be reserved for use by that software. An example would be the non-breaking space (d160) in Microsoft Word.
Markers (View menu)
This command allows you to display the markers in the left pane of the database window.

Note: Toolbox provides three different display options in the left pane of the database window:
- Marker only
- Field Name only
- Both Marker and Field Name
Only one of these options may be selected at any given time. Markers (only) is the default option.

To view the Markers
From the View menu, click on Markers. A check mark will appear beside the option on the menu.

To turn off viewing the Markers
From the View menu, click on Field Names. The check mark beside the Markers will be removed and Field Names will be checked.

Toolbox also allows you to display or hide the Marker Hierarchy.

Field Names (View menu)
This command allows you to display the descriptive field names, if defined, in the left pane of the database window.

If field names have not been specified in the Marker Properties dialog box, then Toolbox must default to displaying only the markers.

Note: Toolbox provides three different Marker display options in the left pane of the database window:
- Marker only
- Field Name only
- Both Marker and Field Name
Only one of these options may be selected at any given time.

To view the Field Names
From the View menu, click on Field Names. A check mark will appear beside the option on the menu.

To turn off viewing the Field Names
From the View menu, click on Markers. The check mark beside the Field Names will be removed and Markers will be checked.

Toolbox also allows you to display or hide the Marker Hierarchy.

Both Markers and Names (View menu)
This command allows you to display both markers and field names (if defined) in the left pane of the database window. If field names have not been specified in the Marker Properties dialog box, then Toolbox must default to displaying only the markers.

Note: Toolbox provides three different display options in the left pane of the database window:
- Marker only
• Field Name only
• Both Marker and Field Name

Only one of these options may be selected at any given time.

**To view the Both Markers and Field Names**

From the View menu, click on Both Markers and Names. A check mark will appear beside the option on the menu.

**To turn off viewing Both**

From the View menu, click on Markers (only) or Field Names (only). The check mark beside the Both Markers and Names will be removed and the option you chose will be checked.

Toolbox also allows you to display or hide the Marker Hierarchy.

## Marker Hierarchy *(View menu)*

This command allows you to display or hide the relative hierarchical structure of the markers in the marker pane of the database window.

**To view the hierarchy**

From the View menu, click on Marker Hierarchy. A check mark will appear beside the option on the menu.

**To turn off the view of the hierarchy**

From the View menu, click on Marker Hierarchy. The check mark beside the option will be removed.

The marker hierarchy is defined in the Marker Properties dialog box for each marker, by specifying which marker the current marker is under in the hierarchy.

Note: Viewing the Marker Hierarchy is compatible with viewing any of the other view options
• Marker only,
• Field Name only, or
• Both Markers and Names.

### Marker Hierarchy: An Overview

In order for some functions of the Toolbox program to work in a more effective manner, it is necessary to specify a hierarchical structure for markers. This is especially true when there are multiple occurrences of fields within a record. The Marker Hierarchy feature allows you to define the relationships between markers—to tell Toolbox that data within the specified fields is related.

The need for a hierarchical structure is especially true in a lexical database containing subentries *(se)* and parts of speech *(ps)* used to organize sections within an entry. An example of a hierarchical structure is as follows:

```
\lx lexeme
  \ps part of speech
    \sn sense number
      \ge gloss (English)
      \de definition (English)
      \xv example (vernacular)
```
The example above places:
- `se` under `lx` in the hierarchy `ps`
- under `se` in the hierarchy `sn`
- under `ps` in the hierarchy `ge`
- under `sn` in the hierarchy `de`
- under `sn` in the hierarchy `xv`
- under `ge` in the hierarchy `xe`
- under `xv` in the hierarchy

The power of using Marker Hierarchy becomes clear when using the **Browse** feature of Toolbox. Records containing multiple occurrences of the same field, such as parts of speech, present a special problem. For example:

When Marker Hierarchy **is not defined**, and you **Sort** by the `gl` field and **Browse** on the `gl`, `ps` and `lx` fields, only the first `ps` in the record is displayed.

```
  gl     ps     lx
  not dirty   adj   clean
  purify     adj   clean
```

However, when Marker Hierarchy **is defined**, the same Sort and Browse, will result in the appropriate `ps` being displayed.

```
  gl     ps     lx
  not dirty   adj   clean
  purify     v     clean
```
Markers tab (Database Type Properties)
This tab allows you to manage the list of Standard Format markers which are available for use in a database type. This tab offers the following options:

**Add**
Adds a new marker to the list available for the database type.

**Copy**
Adds a new marker along with the properties of the marker currently highlighted in the list.

**Modify**
Changes the properties of the marker currently highlighted in the list. Double-clicking on any marker is the same as choosing the Modify option.

**Delete**
Deletes an unused marker from the list.

**Default Language Encoding**
Specifies a default language for any new markers added to the database. New markers are automatically assigned to this language, unless specified otherwise in the Marker Properties dialog box.

Choosing Add, Copy, Modify or double-clicking on any marker causes the Marker Properties dialog box to appear.
Choosing Cancel does not undo Add, Copy, Modify or Delete.

**Note:** The list displayed in this dialog box includes:
- The Marker
- The Field Name
- Language
- Under—Shows hierarchical structure
- SFR—Style (displays C for Character or P for Paragraph), Font (displays an F if default language font is not used) and Range Set (displays an R if defined).

*It is not necessary to specify a list of markers in the Markers tab. When you import a file into Toolbox, the markers used in that file are automatically added to the list of markers and flagged with an asterisk in the field name.

**Markers: Changing one throughout the database**
See also: Markers: An Overview

Global changes can be made to a marker within a particular database. To change a marker, do the following:

1. From the Database menu, choose Properties.
2. In the Database Type Properties dialog box, select the marker you wish to change and click Modify (or double click on the marker).
3. The Marker Properties dialog box* will open. Type the new marker in the Marker edit box.
4. Choose OK.
5. At this point a message box appears asking if you want to replace the former marker with the new marker you just typed. If you do, choose Yes. If you do not, choose No or Cancel.
6. If you chose Yes, choose OK in the Database Type Properties box.
The marker is now changed in every open database assigned to that particular database type. If for some reason, you are unhappy with this change, the original marker can be restored by repeating the steps above and typing the original marker in the Marker edit box.

If the marker you have changed is specified in interlinear settings, the jump path or a filter, it will not be changed automatically in this version of Toolbox. To change these settings, you will need to manually type in the new marker in the appropriate tabs within the Database Type Properties dialog box.

*Note: You can also access the Markers Properties dialog box by right clicking on the marker in the marker pane.
Template for New Records (Database Menu)

The template provides a basic framework of fields (with or without data) for shaping the contents of a record. If specified, the template is inserted automatically when a new record is created. The template is a feature of the database type.

The template is not limited to field markers. Data might be included in some or all of the fields. For example, when inserting a group of nouns, you might want to go ahead and type n in the ps (part of speech) field.

To specify a template
Do the following:

1. Set the contents of the current record to reflect the framework for a database record. This can be done, by choosing an existing record or by inserting a new record and adding the desired fields (with or without data).
2. From the Database menu, choose Template.
3. The Set Template dialog box opens with the following options:
   - OK: Sets the contents of the current record as the template.
   - Cancel: Aborts the command with no changes to the previous template, if set.

Include field contents
This check box allows you to decide whether or not to include the data in the active record. When turned off, the template will include only the field markers with no data. When turned on, includes all the data also.

It is recommended that you include all frequently used fields in the template.

The Multi-Dictionary Formatter suggests a basic set of fields for most records in the lexicon, including:

- lx: lexeme/headword
- ps: part of speech
- pn: part of speech—national language
- ge: gloss—English
- re: reversal—English
- de: definition—English
- gn: gloss—national language
- dn: definition—national language
- rf: reference
- xv: example—vernacular
- xe: example—English translation
- xn: example—national language translation
- ee: encyclopedic information—English
- en: encyclopedic information—national language
- If: lexical function (lexical network)
- le: lexical function gloss—English
- ln: lexical function gloss—national language
- mr: morphology
- bw: borrowed word
- cf: confer/cross-reference
- ce: cross reference gloss—English
cn   cross reference gloss—national language
sd   semantic domain
st   status of entry
so   source
dt   date entry—date stamp

Note: Blank lines between fields are considered field contents. To preserve blank lines between fields in the template, choose the Include Field Contents option.
**Marker Properties: General tab**

This tab allows you to specify general properties for a particular field marker.

Access this tab in one of the following ways:

- Right-click the marker (in the marker pane).
- From the Database menu, choose Properties. From the Markers Tab select the marker and choose Modify, or choose Add to create a new marker. The Marker Properties dialog box opens displaying this tab.

By either method, the Marker Properties dialog box opens displaying this tab.

The following options are offered:

**Marker**
A short name, or code, that is the standard format marker.

It is not necessary to type the backslash. For example, you might type `ps` for part of speech.

**Field Name**
Specifies the field name for the Marker—usually describing the function. For example, **Part of Speech** might be used as the field name for the `ps` marker. (Field names can be viewed in the marker pane of the database window through the Field Names command in the View menu. See also Restricted Characters.)

**Under what in the hierarchy**
Selects the Marker which defines the appropriate hierarchical structure.

**Marker for following field**
If there is a field which will typically follow the field you are defining, Toolbox can automatically insert it in your data. This feature works much like the "Style for following paragraph" feature in Microsoft Word. Any marker is allowed with the exception of the **record marker**.

*Example*: Let's say you are defining the properties for the `ps` marker and you select the `ge` marker in this box. Now when you enter the `ps` field in the marker pane of your database you have only to type in the data and press Enter and Toolbox will automatically insert a `ge` field in your database.

If you need to make an exception, you can change or delete the field that Toolbox enters. However, it is recommended that no following field marker be specified if you must make frequent exceptions.

**Language Encoding**
Sets the language encoding for the marker. By default, this is the language encoding specified on the Markers tab (the tab you see when you do Database, Properties). If you find that you are usually having to select a different language encoding for new markers, you should select a more appropriate default there.

**Language Properties button**
Accesses the Language Encoding Properties dialog box where the language name, sort order, case, variables, characters, keyboard and font specifications are set for the language used in the field.

**Description**
Describes the field represented by this marker. This description provides a way for the user to distinguish between various data fields.
Use Language Font
This check box allows you to use the font specified for the particular language. When turned on, the Choose Font button is disabled. It is recommended that you use the default font for the language when possible. The language font is set in the Options tab of the Language Encodings Properties dialog box.

Choose Font
This button is enabled when the Use Language Font option is turned off. It accesses the Font dialog box, allowing you to change the font, style (such as bold and italic), font size and color for the particular marker. This feature overrides the Language font settings.

Style to Export
When exporting a database in Rich Text Format for use in Microsoft Word, it is helpful to have the markers assigned to a style which can be readily used by the template in Word.

Character style
Affects only the text included in this field and applies any of the formats of the Font command (font, font size, bold, italic, etc.). Some markers that might be assigned a character style include the \ps (part of speech) and \ge (English gloss). Character styles, if different from the paragraph style they occur in, will override any formatting set for that paragraph.

Paragraph style
Controls all aspects of a paragraph's overall appearance (text alignment, tab stops, line spacing, borders, formatting, etc.). Some markers that might be assigned a paragraph style include the \lx (lexical entry) and \se (subentry).

Marker Properties: Range Set tab
Some data fields have only a limited set of valid values. When those can be listed, they can be used to check that your data is consistent. Such fields are the part of speech, the semantic domain, and possibly other similar fields. We call this limited set of values a "Range Set". This tab allows you to specify the set of valid values for data fields of a particular marker.

Access this tab in one of the following ways:

- Right-click the marker (in the marker pane) and choose the Range Set tab in the Marker Properties dialog box.
- From the Database menu, choose Properties. Select the marker and choose Modify. The Marker Properties dialog box opens allowing you to select the Range Set tab.

Creating a New Range Set
There are three ways to create a Range Set.

- One is to manually enter the values you anticipate using.
- A second is to turn on the Range Set option and the Consistency Check for the Range set of that field. Then as you add new data elements in that field, Toolbox will regard them as inconsistent,
giving you the opportunity to decide on a case-by-case basis whether to add the new data into the range set. This approach is best used at the beginning of a new database.

- The third is to use the “Build Range Set” option. This is the most useful if you are starting the Range Set after the data has a number of records.

It is acceptable to leave the range set empty, particularly when you are just starting.

Adding a Range Set Element

1) Enter the new element in the Range Set Element box. If the element you enter is not already in the range set, the Add button will be available.
2) Choose Add. The new element will appear in the Range Set list.
3) You may continue adding additional elements or setting other marker properties. When you are finished, choose OK to save any changes.

Modifying / Editing a Range Set Element

1) Select the element you wish to modify in the Range Set list. The element will appear in the Range Set Element box.
2) In the Range Set Element box, make the needed changes. If the modified version of element is not already in the range set, the Replace button will be available.
3) Choose Replace. The modified element will appear in the Range Set list, replacing the original one. (NOTE: This does not replace the data in your database.)
4) You may continue making additional modifications or setting other marker properties. When you are finished, choose OK to save any changes.

Building a Range Set from the Existing Data

If you have an existing field that already has data in one or more databases, you may wish to scan the existing data and build a new range set (or add to an existing one) based on that data. To do this, follow these steps:

1) Before proceeding, confirm that the data properties selected for this marker are correct. These settings will affect the way Toolbox interprets the existing data. See the discussion of Data Properties, below, for more details.
2) It may be helpful to begin by removing any existing range set elements that are invalid (see below for specific steps).
3) Choose Build range set from the data. Toolbox will add the field contents of all open databases of this database type to the range set. This may take a few minutes. When it is finished, Toolbox displays a message with some statistics for your information.
4) Choose OK.
5) You may need to edit the list Toolbox produces to delete or modify invalid elements or to add additional valid elements that didn’t occur in the data. Or you may need to remove the list, modify the Data Properties, and remake the list.
6) You may continue setting other marker properties. When you are finished, choose OK to save any changes.

Deleting a Range Set Element

- Select the element you wish to delete in the Range Set list.
- Choose Delete. The modified element will be removed from the Range Set list.
- You may continue making additional changes to the range set or setting other marker properties. When you are finished, choose OK to save any changes.
Deleting all Range Set Elements

If the elements of the range set are all or mostly wrong, it may be easiest to delete all of them and start from scratch. To do this, choose Clear All. Toolbox will remove all elements from the Range Set list. Note that this does not disable range set checking for this marker (see additional information above under Creating a New Range Set).

You may now proceed to add elements or set other marker properties. When you are finished, choose OK to save any changes.

Eliminating a Range Set

If you no longer want to constrain field data to the values in the range set, click to clear the Use Range Set check box. This will make the other buttons on this tab unavailable. The elements in the range set will be preserved until you choose OK, in case you change your mind, but then they will be discarded. If you are unsure, select and copy the list to a text file.

Note that clearing all the elements from the range set does not disable range set checking for this marker (see above).

Relationship between Data Properties and Range Sets

When you choose OK, the values entered into the range set will be checked to make sure they do not violate the data properties for this marker. Specifically, if you indicate that field data consists of a single word or of multiple one-word items, then range set elements must be single words. If, however, the marker data consists of a single item (possibly consisting of multiple words), then range set elements can be multiple words. If you are having difficulty getting these settings to work together properly, see Troubleshooting Range Sets.

Marker Properties: Data Properties tab

This tab allows you to specify properties that affect the way Toolbox treats the data in a particular field. The settings you select will affect data validation (i.e., consistency checking) and a number of other features, including jumping, searching, and reshaping.

Access this dialog box in one of the following ways:

- Right-click the marker and choose the Range Set tab in the Marker Properties dialog box.
- From the Database menu, choose Properties. From the Markers Tab select the marker and choose Modify. The Marker Properties dialog box opens allowing you to select the Data Properties tab.

The following options are offered within this tab:

Field data consists of

- **a single word (i.e. no embedded spaces)**
  
  Tells Toolbox to treat the data in this field as a single item and to not allow any white space. This option is useful for fields such as part of speech or sense number, where it would be invalid to have more than one "word" in a given field.

- **multiple one-word items (default)**
  
  Tells Toolbox to treat each word in this field as a separate item. If this option is selected, jumping, searching, etc. will use the current "word" as the text to look for.

- **a single item (possibly consisting of multiple words)**
Tells Toolbox to treat the data in this field as a single item. If this option is selected, jumping, searching, etc. will cause Toolbox to look for the entire contents of the current field. This option is useful for fields containing filenames, idiomatic expressions, compound words, etc.*

Allow empty field contents
This option is selected by default. If you want Toolbox to help you avoid leaving a field in your database empty, clear this check box. Requiring data for a field is often useful for a marker with a range set or a marker which is set up as the source field of a data link. (See Consistency Checks.)

No Word Wrap
By default, Toolbox automatically inserts line breaks as needed to make the contents of a data field fit within the margin as you type. If you select this option, wrapping and reshaping are disabled so that you can control the line breaks. This option is not generally useful if your data consists of a single item (either a single word or a multiple-word expression).*

After you have edited a record, the data are checked automatically against the data properties.
Toolbox can also check consistency of all records in a window. If it finds a data property inconsistency, it displays the Data Property Consistency Check dialog box.

* If you tell Toolbox to treat the data in a field as a single item, then jumping, searching, etc. will use the entire contents of the current field. However, it will only match up to 100 characters. If the marker has the No Word Wrap property selected, then only the first line of the field will be used.

Data Properties and Range Set
The following shows the way Toolbox will interpret existing field data when building a range set (on the Range Set tab).
If your data properties settings indicate that field data consists of:

• a single word...
  Toolbox will scan existing data to build a range set whose elements consist of a single word. Any fields of the given marker that have multiple words in the existing data will be skipped. This means that if you check consistency immediately after building the range set from data, there could be fields which will have inconsistencies.

• multiple one-word items...
  Toolbox will scan existing data to build a range set whose elements consist of a single word. Any fields of the given marker that have multiple words in the existing data will have those words entered as separate elements of the range set.

• a single item (possibly consisting of multiple words)...
  Toolbox will scan existing data to build a range set whose elements consist of one or more words. Any fields of the given marker that have multiple words in the existing data will have their entire data contents added to the range set as a single element.

If you are having difficulty getting these settings to work together properly, see Troubleshooting Range Sets.

Character-based Range Sets
See also: Markers: An Overview
Character-based range sets allow Toolbox to check field data one character at a time to determine if each character is included in the set of acceptable characters. This feature is no longer deemed to be necessary for most databases. While Toolbox will still correctly perform consistency checks on data fields
with character-based range sets, it is no longer possible to create this type of range set within Toolbox. The following information will help you determine whether you should continue to use an existing character-based range set or use a different feature of Toolbox to achieve the desired result. This help topic may also be useful for determining what kind of consistency checking will best meet your needs if you are creating a new marker or setting up consistency checking for an existing database.

**When to convert to an item-based range set**

1) If your data consists of single-character codes, you can automatically convert to an item-based range set. Then if you also set the data properties so that Toolbox knows to limit the data in that field to a single word, you can achieve even greater control over your data. For example, if your character-based range set contained valid codes a, b, and c and you had a field that incorrectly had the value ab, Toolbox would have determined that the data was valid. However, after converting to an item-based range set and setting the marker's data properties to indicate that data should consist of a single word, Toolbox will correctly detect ab as an inconsistency because the data item ab is not in the range set. 2) If your data consists of a finite set of numbers. One of the typical uses of character-based range sets is for guaranteeing that numeric data consists only of numerals. One obvious use of a numeric field is for keeping track of sense numbers (\(\text{sn in MDF}\)). However, as in the example above, limiting the data to any combination of numerals means that 1242345 would pass as a valid sense number, which is highly unlikely, even for a very thorough lexicon. It is better to convert to an item-based range set, and then add a few two-digit numbers if necessary (if per chance you should have sense numbers higher than 9).

**When to use data links instead of a range set**

If your data consists of a finite set of codes that refer to items in another database, then you should consider no longer using a range set at all. Instead, create a jump path from this source field to the destination(s) that comprise the master list of valid codes. Establish this jump path as a data link and Toolbox will help you keep your codes consistent with the list.

**When to continue using a character-based range set**

If you are using a character-based range set to impose restrictions on the set of characters that can be used in a particular field, but the number and order of the occurrences of the characters is unbounded, then character-based range sets are probably still your best option. This would include fields such as phone numbers, counts, or other numbers that are not a reference to other data. Note that even character-based range sets cannot fully validate most data of this type (e.g., Toolbox would accept (1 as a valid phone number because all the characters are in the range set).

**Range Sets: Troubleshooting**

Because the data properties selected for a marker affect the way Toolbox interprets the data in a field of that type, it is possible to have a range set that appears to include the needed elements, when it may not. This can result in Toolbox reporting unexpected inconsistencies. It is also possible to build a range set that accidentally includes extra elements, which could result in Toolbox quietly allowing data that should be identified as inconsistent. Following are some questions to help you identify the cause of your problem and some instructions for fixing it.

**Before you begin**, make sure that the inconsistency you are seeing or that you expect to see is a range set inconsistency, and not a data property or data link inconsistency. Also make sure that for the field in question, the Use range set check box is selected on the Range Set tab of the Marker Properties dialog box.

1) Is Toolbox identifying a range set inconsistency for data that you believe to be consistent with the range set?
   a) Is Toolbox identifying a single character as inconsistent?
      Please see the help topic about Character-based Range Sets.
   b) Does the data in each field consist of a single word?
      i) Is "Field data consists of a single word" selected on the Data Properties tab of the Marker Properties dialog box?
      Look at the data that Toolbox is identifying as inconsistent. Make sure that it does not contain any spaces or line breaks. If it does, Toolbox will not be able to find it in the
range set because the elements of this range set are single words. To have Toolbox accept data consisting of multiple words, choose one of the other options on the Data Properties tab.

If the field truly contains only a single word, it must be missing from the range set. Go to the Range Set tab of the Marker Properties dialog and examine the range set. Check for differences in case (range sets are case-sensitive) or punctuation. If the data does not appear in the range set, you can add it manually. If there are several different elements that appear to be missing and you are reasonably certain that your data doesn't contain data that you don't want in the range set, you might want to build the range set from the existing data. For more information about adding range set elements and building range sets from existing data, see the help topic about the Range Set tab.

ii) Is "Field data consists of multiple one-word items" selected on the Data Properties tab of the Marker Properties dialog box?

The data must be missing from the range set. Go to the Range Set tab of the Marker Properties dialog and examine the range set. Check for differences in case (range sets are case-sensitive) or punctuation. If the data does not appear in the range set, you can add it manually. If there are several different elements that appear to be missing and you are reasonably certain that your data doesn't contain data that you don't want in the range set, you might want to build the range set from the existing data. For more information about adding range set elements and building range sets from existing data, see the help topic about the Range Set tab.

Note that if you really expect the field data to consist of a single word, you can have Toolbox check this by selecting "Field data consists of a single word" on the Data Properties tab of the Marker Properties dialog box.

iii) Is "Field data consists of a single item" selected on the Data Properties tab of the Marker Properties dialog box?

The data must be missing from the range set. Go to the Range Set tab of the Marker Properties dialog and examine the range set. Check for differences in case (range sets are case-sensitive) or punctuation. Also check to make sure that the word exists by itself as a range set element. If it exists as a word that is part of a multi-word element, Toolbox will not consider it to be a match. If the data does not appear in the range set, you can add it manually. If there are several different elements that appear to be missing and you are reasonably certain that your data doesn't contain data that you don't want in the range set, you might want to build the range set from the existing data. Note, however, that if you build a range set from data, Toolbox will add range set elements based on the entire contents of each existing data field, not as individual words. For more information about adding range set elements and building range sets from existing data, see the help topic about the Range Set tab.

Note that if you really expect the field data to consist of a single word, you can have Toolbox check this by selecting "Field data consists of a single word" on the Data Properties tab of the Marker Properties dialog box.

c) Do you expect Toolbox to look in the range set for each individual word in the data field?

i) Is "Field data consists of a single word" selected on the Data Properties tab of the Marker Properties dialog box?

If the field data consists of multiple words, Toolbox will identify the entire field as inconsistent because the data property setting indicates that the field should allow only a single word. To have Toolbox accept data consisting of multiple words and look in the range set for each individual word in the data field, select "Field data consists of multiple one-word items" on the Data Properties tab of the Marker Properties dialog box.

ii) Is "Field data consists of multiple one-word items" selected on the Data Properties tab of the Marker Properties dialog box?
The data must be missing from the range set. Go to the Range Set tab of the Marker Properties dialog and examine the range set. Check for differences in case (range sets are case-sensitive) or punctuation. If the data does not appear in the range set, you can add it manually. If there are several different elements that appear to be missing and you are reasonably certain that your data doesn't contain data that you don't want in the range set, you might want to build the range set from the existing data. For more information about adding range set elements and building range sets from existing data, see the help topic about the Range Set tab.

iii) Is "Field data consists of a single item" selected on the Data Properties tab of the Marker Properties dialog box?

This data property setting tells Toolbox to treat the entire field as a single item and to look for it as a multiple-word element in the range set. Even if the individual words of the data field exist as elements in the range set, Toolbox will identify the entire field as inconsistent. To have Toolbox look in the range set for each individual word in the data field, select "Field data consists of multiple one-word items" on the Data Properties tab of the Marker Properties dialog box. Note that if you change this setting, Toolbox will not allow you to keep in the range set any elements consisting of multiple words. If you previously built the range set from the existing data, there may be many such elements. It will probably be easiest to choose Clear All on the Range Set tab and then rebuild the range set from the data (after you change the data property setting). This will generate a new range set consisting of one-word elements. If some of the existing data contain words that you do not want Toolbox to accept as consistent, you can delete the unacceptable elements.

d) Do you expect Toolbox to look in the range set for the entire contents of the data field (even though it may consist of multiple words)?

i) Is "Field data consists of a single word" selected on the Data Properties tab of the Marker Properties dialog box?

If the field contains multiple words, Toolbox will identify the entire field as inconsistent because the data property setting indicates that the field should allow only a single word. To have Toolbox accept data consisting of multiple words and look in the range set for the entire contents of the data field, select "Field data consists of a single item" on the Data Properties tab of the Marker Properties dialog box.

ii) Is "Field data consists of multiple one-word items" selected on the Data Properties tab of the Marker Properties dialog box?

This data property setting tells Toolbox to look up each word of the field data individually in the range set. To have Toolbox look in the range set for the entire contents of the data field, select "Field data consists of a single item" on the Data Properties tab. Note that if you change this setting, you will probably need to rebuild the range set since all the existing elements will be single words rather than whole data items. It will probably be easiest to choose Clear All on the Range Set tab and then rebuild the range set from the data (after you change the data property setting). After generating a new range set, if some of the existing data contain items that you do not want Toolbox to accept as consistent, you can delete the unacceptable elements.

iii) Is "Field data consists of a single item" selected on the Data Properties tab of the Marker Properties dialog box?

The data must be missing from the range set. Go to the Range Set tab of the Marker Properties dialog and examine the range set. Check for differences in case (range sets are case-sensitive) or punctuation. Note that even if the range set contains elements that have one or more words that match parts of the field data, Toolbox will not regard it as consistent. There must be a single element which matches the data exactly. If the data does not appear in the range set, you can add it manually. If there are several different elements that appear to be missing and you are reasonably certain that your data doesn't contain data that you don't want in the range set, you might want to build the range set from the existing data. For more information about adding range set elements and building range sets from existing data, see the help topic about the Range Set tab.
e) If you have followed the diagnostic steps above and the troubleshooting hints did not help you resolve your problem, this help topic does not address your problem. If you need further assistance, please contact technical support.

2) Is Toolbox failing to identify a range set inconsistency for data that you believe to be inconsistent with the range set?

a) Is Toolbox failing to identify a range set inconsistency when doing automatic checking following data entry?

i) Is Check Consistency When Editing turned off?

   It is possible to turn off consistency checking when editing so that Toolbox will not perform automatic checking. If Toolbox is failing to identify range set inconsistencies in newly entered data, it may be that this feature is turned off. Make sure that there is a check mark beside Check Consistency When Editing on the Checks menu. ii) Is Toolbox skipping all checking after you edit this record (if you aren’t sure, read this point for further diagnosis)?

   If you are just looking at a record but don’t make any changes, Toolbox will not alert you to any inconsistencies. Also, if you have an existing database and add or change a range set or alter other settings, Toolbox will not automatically re-check unedited data based on those changes. Automatic consistency checking only happens when you actually edit a record, followed by an attempt to perform an action that could potentially cause the current record to change or that would close the window in which the edited record is displayed.

   If Toolbox does not appear to be doing any automatic checking or you aren’t sure, check to make sure that you are triggering the check by temporarily making a change that you think should cause data to become inconsistent (preferably in a field other than the one where you are currently seeing a problem). Then choose Database, Next Record (Alt-N) to move to the next record. If you have in fact entered inconsistent data, Toolbox should identify it as such. Regardless of the results, make sure you return your data to a proper state when you have completed this test.

iii) If your problem is still not solved, proceed to c below.

b) Is Toolbox failing to identify a range set inconsistency when you choose Check Consistency on the Checks menu?

   When you choose to check the consistency of a database, Toolbox displays the Check Consistency dialog, where you have the option to select which marker(s) to check and which check(s) to perform. Toolbox remembers your selections, so you need to make sure that the selections you are now using include the field you wish to check and that the “Check Range Sets” check box is selected. If this does not solve your problem, proceed to c below.

c) After considering both a and b above, is Toolbox failing to identify a range set inconsistency in both cases (if you are only trying one of the cases but you have gone through the material above, you can probably assume it would fail in the other case as well)?

   i) Is there an element in the range set that shouldn’t be there?

      If someone else created this range set or if it was built based on existing data or if you accidentally chose Add to Range Set at some point when Toolbox displayed an inconsistency, it is possible that the range set contains one or more unexpected elements, corresponding to data that you regard to be inconsistent. Carefully, examine the range set (on the Range Set tab of the Marker Properties dialog box) to make sure that it does not include the data that Toolbox is erroneously accepting as consistent.

   ii) Is “Field data consists of multiple one-word items” selected on the Data Properties tab of the Marker Properties dialog box?

      This data property setting tells Toolbox to look up each word of the field data individually in the range set. One reason why Toolbox could be accepting data as consistent when you don’t think it should is that you may be expecting Toolbox to look in the range set for the entire contents of the data field, when it is actually looking up (and finding) each individual word.
If looking up each word is not what you intend, but rather you really want it to treat the whole field as a single item, select "Field data consists of a single item" on the Data Properties tab. Note that if you change this setting, you might need to rebuild the range set since all the existing elements will be single words, and so now Toolbox will regard as inconsistent any data items consisting of multiple words. To start over and rebuild the range set, choose Clear All on the Range Set tab and then rebuild the range set from the data (after you change the data property setting). After generating a new range set, if some of the existing data contain items that you do not want Toolbox to accept as consistent, you can delete the unacceptable elements.

iii) Is "Field data consists of a single item" selected on the Data Properties tab of the Marker Properties dialog box?

This data property setting tells Toolbox to treat the entire field as a single item and to look for it as a multiple-word element in the range set. One reason why Toolbox could be accepting data as consistent when you don't think it should is that you may be expecting Toolbox to look in the range set for each individual word, when it is actually looking up (and finding) a single range set element that matches the entire field contents.

If looking up the entire field contents as a single item is not what you intend, but rather you really want it to treat each word as a separate item, select "Field data consists of multiple one-word items" on the Data Properties tab. Note that if you change this setting, you will probably need to rebuild the range set since all the existing elements will be single words rather than whole data items. It will probably be easiest to choose Clear All on the Range Set tab and then rebuild the range set from the data (after you change the data property setting). After generating a new range set, if some of the existing data contain items that you do not want Toolbox to accept as consistent, you can delete the unacceptable elements.

d) If you have followed the diagnostic steps above and the troubleshooting hints did not help you resolve your problem, this help topic does not address your problem. If you need further assistance, please contact technical support.

3) Is Toolbox preventing you from adding elements to the range set?

a) Is “Field data consists of a single word or multiple one-word items" selected on the Data Properties tab of the Marker Properties dialog box?

Toolbox will not allow you to save a range set containing multiple-word elements if the data property setting indicates that the field should contain a single word. To have Toolbox accept data consisting of multiple words and look in the range set for the entire contents of the data field, select "Field data consists of a single item" on the Data Properties tab of the Marker Properties dialog box. Then Toolbox will let you add multiple-word range set elements.

b) If you have followed the diagnostic steps above and the troubleshooting hints did not help you resolve your problem, this help topic does not address your problem. If you need further assistance, please contact technical support.

4) Is Toolbox failing to generate certain elements when building a range set from existing data?

a) Is “Field data consists of a single word" selected on the Data Properties tab of the Marker Properties dialog box?

With this data property setting, Toolbox will not add any range set elements based on fields whose data consist of more than one word. For example, if you are building a range set for the \texttt{code} field and it occurs only twice in your database:

\begin{Verbatim}
\texttt{t 101 202 and t 303}
\end{Verbatim}

then the only element that Toolbox will add to the range set is 303. To have Toolbox accept data consisting of multiple words (and build the range set accordingly), choose one of the other options on the Data Properties tab. If the field really is only supposed to have a single word, then the correct resolution may be to split the first field in two:

\begin{Verbatim}
\texttt{t 101}
\end{Verbatim}
b) Is the data that you wish to add to the range set in a closed database?
   Toolbox only searches open databases when building a range set. Open the database and retry this operation.

c) Is the data that you wish to add to the range set in a database of a different type?
   Toolbox only searches databases of the same type when building a range set. It is possible to have both a lexicon and an interlinear text with a `ps` field, for example. But since these two databases will not be of the same type, Toolbox will not search the interlinear text database when building a range set for the `ps` field of the lexicon. In this example, chances are you wouldn’t really want or need the data from the `ps` field of the interlinear text database to be included as valid parts of speech for your lexicon, but if you do find yourself needing to have identical range sets for two fields in different types of databases, you will need to create them separately.

d) If you have followed the diagnostic steps above and the troubleshooting hints did not help you resolve your problem, this help topic does not address your problem. If you need further assistance, please contact technical support.

5) If you have followed the diagnostic steps above and the troubleshooting hints did not help you resolve your problem, this help topic does not address your problem. If you need further assistance, please contact technical support.

Whole Windows

*Things that can be done with whole windows in Toolbox*

**Arrange Icons (Window Menu)**
This command organizes the display arrangement of the "bricks" (document buttons) which appear when a database window has been minimized.

Note that you can arrange the bricks yourself by doing a single click on a brick and dragging it (hold down the left mouse button) to where you want it. Toolbox will remember the location of the bricks and place them there again when you minimize the same window.

**Cascade (Window menu)**
This command automatically arranges the display of multiple windows where they are positioned to overlap one another, with each subsequent window located slightly below and to the right of the previous one.

**Duplicate command (Window menu)**
This command enables you to open multiple windows on the same database. The exact number depends on the amount of available memory. This allows you to concentrate on different aspects or to change your focus on the database without leaving your current position. Each duplicate of the same database can be filtered and/or sorted independent of the other(s).

Choosing the Duplicate command opens another window on the active database. Any changes made within the database are global—they are reflected in all duplicate windows of the same database.

*Note:* It is important to remember that the duplicate command does not open a "copy" of the database, but instead opens another window on the database.
If the current database window is maximized, the duplicate will also be maximized. The only obvious difference between the two is the number after the database filename on the title bar. From the Window menu, choose Tile to see both database windows.

The Window menu allows optional ways of organizing multiple databases within the application window. These options include:

- Cascade
- Tile
- Tile Side by Side

When closing, it is important to remember that the Close command (File menu) closes all duplicate windows on the active database. To close only the active database window allowing duplicate windows of the same database to remain open, use the Close box in the upper right corner of the database window.

**Large Controls (View menu)**

This Toolbox feature was developed to make the screen easier to read when using larger fonts. Characters with stacked diacritics are more distinct when viewed at a larger point size.

This command allows you to change the size of the status bar and the portions of dialog boxes where text is entered.

The large controls are displayed when a check mark appears beside the name in the View menu. To obtain more standard-sized controls, choose the command again.

**Note:** This feature will also benefit those whose eyesight is poor or failing.

**Maximizing a window ***

**Maximize Button**

This feature of Windows expands the Toolbox window to fill the entire screen or the database window to fill the Toolbox window.

**To maximize the database window**

Do one of the following:

- From the database Control menu box, choose Maximize.
- Double-click on the title bar of the database.
- Click the maximize button on the right end of the title in the database window.

**To maximize the Toolbox window**

Do one of the following:

- From the application Control menu box, choose Maximize.
- Double-click on the title bar.
- Click the Maximize button on the right end of the title in the application window.

**Note:** After you expand a window, the Maximize button changes to the Restore button. To restore the window:

- Choose Restore from the Control menu box
- Click the Restore button on the title bar of the database.
If the current database window is maximized, a duplicate window opened with the Jump To or the Duplicate command will also be maximized. At times the only obvious difference between the two is the number after the database filename on the title bar.

**Minimizing a window ***

**Minimize Button**
This feature allows you to reduce the application or the database window. When minimized, the application or database stays in memory, but its window does not take up space on the screen or in the Toolbox window.

**To minimize the database window**
Do one of the following:
- From the database Control menu box, choose Minimize.
- Click the Minimize button on the right end of the title bar in the database window.

**To minimize the Toolbox window**
Do one of the following:
- From the application Control menu box, choose Minimize.
- Click the Minimize button on the right end of the title bar in the application window.

**Note:** After minimizing a window, the minimize button changes to the Restore button. To restore the window:
- Choose Restore from the Control menu box
- Click the Restore button on the title bar of the database.

**Restoring a window**

**Restore Button**
To restore a window to its original size, do one of the following:
- Double-click on its "brick" (document button) or click once on the brick and choose Restore from the application Control menu.
- Click the Restore button on the title bar of the database.

**Sizing a Window**
You can make the database window smaller so that you can view more than one window at a time, or larger to see more of the contents of the database.

You can leave Toolbox temporarily to work in another application by minimizing the application window. You can quickly restore the program by clicking on the Toolbox taskbar button ("brick") on the taskbar.

**To size a window with the mouse**
Do one of the following:
- Click the Maximize button on the title bar to expand the window to its maximum size.
- Click the Minimize button on the title bar to reduce the window to a taskbar button.
- Click the Restore button in the upper-right corner of the window to restore the window to its original size.
To change the size of the window, position the mouse pointer along the border of the window you want to size. When the mouse pointer changes to a double-headed arrow, drag the border until the window is the size you want.

To size a window with the keyboard

Do the following:

1. If the window is enlarged to its maximum size, restore it to its original size by pressing ALT+HYPHEN to open the document Control menu or ALT+SPACEBAR to open the application Control menu, and then choosing Restore.
2. From the database Control menu or application Control menu, choose Size.
3. When the mouse pointer changes to a four-headed arrow, use an arrow key to select the border you want to move. For example, press the DOWN ARROW key to select the bottom border.
4. When the mouse pointer changes to a double-headed arrow, press the arrow keys until you move the borders to the desired size. If you need to return the window to its original size, press ESC.
5. When you finish changing the size of the window, press ENTER.

Note: You cannot move or size a window that has been enlarged to its maximum size. Double-clicking on the title bar itself, toggles between restored and maximized for the application window.

Tile (Window menu)

This command automatically arranges the display of multiple windows where the workspace is divided horizontally in equal sizes, so you can see the work area of each database.

Tile Side by Side (Window menu)

This command automatically arranges the display of multiple windows where the workspace is divided vertically in equal sizes, so you can see the work area of each database.

Window List (Window menu)

Toolbox lists the names of up to nine open windows at the bottom of the Window menu. If more than nine windows are open, the More Windows command is displayed on the menu. Choose More Windows to display a dialog box listing all open windows.

When databases are maximized within the database window, you can easily change from one database to another by clicking on the file name, or typing the number associated with the file (the number on the left). Toolbox displays a check mark next to the active window.

Duplicate windows of the same database show a number (on the right) after the file name which relates to the actual number of windows currently open of that particular database.

Note: Use CTRL+F6 to cycle between the various windows.
MDF Multi-Dictionary Formatter

New: Jan 2020 MDF Information

Introduction

New: Standard Lexical Database Field Markers (for MDF)

Introduction

MDF is a set of markers designed for multi-lingual dictionaries. It was created by Grimes and Coward and documented in a book named "Making Dictionaries". A PDF file of that book can be downloaded from the MDF page of the SIL website.

MDF is also a printing (export) process. The letters MDF stand for Multi-Dictionary Formatter because the printing process can produce a variety of dictionary formats with no changes to the data.

Because the MDF printing process was built into the Windows version of Shoebox (now Toolbox), both have spread -- each feeding the popularity of the other. The MDF marker set is now used by linguists worldwide.

Organizing the Markers

One problem faced by users of MDF is that of "too many markers". The following summary is an attempt to make the number seem less threatening.

... by Language

A large number of markers come in sets of three or four. MDF attempts to generalize the need for multiple languages by recognizing four possible languages and providing markers for them. Most of the MDF markers are two characters long. The second character is often (not always) a language character.

- e indicates English language markers
- n indicates National language markers
- r indicates Regional language markers
- v indicates Vernacular language markers

Some markers have the language assumed. So the lexeme (lx) and subentries (se) are always vernacular language, even though they don't end with "v" (and even though subentry ends with "e"). Some of these, like the variant (va), have related markers (comments in this case) for the other three languages.

There is no explicit provision for the compiler’s language if it’s other than English. This can be handled by using either the set of “e” markers or the “r” markers.

... by Function

In a real sense, everything in the entry relates to the headword. But some fields are more tightly related than others. Some are more related to the definition, some are your personal notes on different topics, etc. This is an attempt to organize the approximately 100 MDF markers.

Note: The following does not give the order in which these fields normally appear in a
Note: A marker like de* means that there is a set of language markers beginning with d: dv, dn, dr, de.

Information Relating Directly to the Headword

<table>
<thead>
<tr>
<th>lx lexeme</th>
<th>hm homonym number</th>
<th>lc lexical citation form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ph phonetic (pronunciation)</td>
<td>ps part of speech</td>
<td>pn national part of speech</td>
</tr>
<tr>
<td>sn sense number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Information about the Headword

<table>
<thead>
<tr>
<th>va variant form (also ve* comments)</th>
<th>mr morphology</th>
</tr>
</thead>
<tbody>
<tr>
<td>It literally</td>
<td>sc scientific name</td>
</tr>
<tr>
<td>sd semantic domain</td>
<td>is index of semantics</td>
</tr>
<tr>
<td>th thesaurus</td>
<td>pd paradigm class</td>
</tr>
<tr>
<td>pdl paradigm label (also pdv* paradigm form &amp; glosses)</td>
<td></td>
</tr>
</tbody>
</table>

Other forms of the Headword

<table>
<thead>
<tr>
<th>va variant form (also ve* comments)</th>
<th>mr morphology</th>
<th>pd paradigm class</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdl paradigm label (also pdv* paradigm form &amp; glosses)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Related entries

<table>
<thead>
<tr>
<th>cf cross reference</th>
<th>mn main-entry cross reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>If lexical function (like synonym, causal, generic, deverbal noun, locative, etc.)</td>
<td></td>
</tr>
<tr>
<td>(also lv* lexical function form and glosses)</td>
<td></td>
</tr>
</tbody>
</table>

Origins

<table>
<thead>
<tr>
<th>bw borrowed word</th>
<th>et etymology (also eg gloss, es source, ec comment)</th>
</tr>
</thead>
</table>

Definition and Range of Meaning

<table>
<thead>
<tr>
<th>de* definition</th>
<th>ee* encyclopedic information (expanded definition)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ue* usage</td>
<td>oe* &quot;only&quot; (restrictions)</td>
</tr>
<tr>
<td>ge* gloss (for interlinear)</td>
<td>re* reversal (for finder lists)</td>
</tr>
<tr>
<td>we* word-level gloss</td>
<td></td>
</tr>
</tbody>
</table>

Examples

<table>
<thead>
<tr>
<th>rf source of example (reference)</th>
<th>xv* examples and translations</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc picture</td>
<td>snd sound</td>
</tr>
<tr>
<td>vid video</td>
<td></td>
</tr>
</tbody>
</table>

Dialects

There are no specifically "dialect" markers. Various alternative approaches can be used. The revision of Chapter 2 discusses the following as possibilities.

| va* (variant) | ue* (usage) | oe* (only / restrictions) |
If you are unsure which marker to use, first look under 'Summary_of_Fields' for the code you want, and then search for that field marker.

Three Types of Field Markers

In this concept, each field marker is described by of the following categories:

<Basic>
This marks all of the "basic" field markers. These are the field markers which will be useful to the most people. Consider including all of the basic field markers in your Toolbox lexical database template, so they will be inserted automatically into each of your new records. Leaving a basic field empty is okay.

<Reserved>
This marks the "reserved" fields, meaning, these fields serve a crucial function in the MDF formatting process and so should be used only if needed. These are the \lc, \hm, \se, and \sn fields. Empty reserved fields should not be included in your records, i.e. if present, they should contain data.

<Optional>
This marks the "optional" fields. These are the more technical fields given to cover just about every topic. You may want to include some of these in your database template as well. And no doubt, some of these optional fields will be of no use to you. Like with basic fields, leaving an optional field empty is okay.

Multiple Fields

The number of occurrences of any given field is not restricted. But in printing through the standard MDF export process, multiple occurrences of a given field will be concatenated within a hierarchical section, e.g. with a sense, or if there is no sense, then within a part of speech section, or subentry section, etc. Multiple occurrences within the same record but in separate sections (sense, part of speech, or subentry) are kept separate.

Order_of_Fields

Recommended Field Order

The following are all of the fields in their recommended order for a lexical entry. The order shown here assumes you are using the standard hierarchy (the alternate hierarchy only varies in the hierarchical order of the \sn, \se, and \ps fields). For more information on the alternate hierarchy, see:

Alternate_Hierarchy

The field order shown here is nearly identical to the order in which MDF outputs the fields when formatting your lexicon. This needs to be stressed that when MDF formats your lexicon, the fields are rearranged to (nearly) the following order. This is to provide a consistent field sequence to each and every lexical entry (a much desired goal of any
The main differences between the recommended order given here and the order in which MDF outputs the fields are as follows:
1) the \lx and \lc fields -- in the database, the \lx field must come first, but for printing, the citation (\lc) field is the first field out (if it exists).
2) the gloss and definition fields -- the gloss field prints only if there is no definition field.
3) the reverse fields and the word gloss fields -- these generally do not print; but if you request them, they print grouped together after the definitions.

It is recommended that you use the database template feature of Toolbox to help you maintain a consistent ordering of your fields.

Deviating from this order (especially for the \se, \ps, \pn, and \sn fields) could foul MDF formatting, since these fields also serve to mark section boundaries. For more on this, see:

Sections_in_a_Lexical_Entry

In the following list, the following codes are used:

'B' marks 'Basic' fields (17 total)
'R' marks 'Reserved' fields (4 total)
(all other fields are 'Optional')

\lx (lexical entry) ONLY ONE PER LEXICAL ENTRY
R \hm (homonym number)
R \lc (lexical citation)
  \ph (phonetic/phonemic form)
R \se (subentry)
B \ps (part of speech)
B \pn (part of speech-national)
R \sn (sense number)
  \gv (gloss-vernacular)
  \dv (definition-vernacular)
B \ge (gloss-English)
B \re (reversal form-English) [default set not to print]
  \we (word gloss-English) [default set not to print]
B \de (definition-English)
B \gn (gloss-national)
B \rn (reversal form-national) [default set not to print]
  \wn (word gloss-national) [default set not to print]
B \dn (definition-national)
  \gr (gloss-regional)
  \rr (reversal form-regional) [default set not to print]
  \wr (word gloss-regional) [default set not to print]
  \dr (definition-regional)
  \lt (literal meaning)
  \sc (scientific name)
B \rf (reference to notebooks, etc.)
B \xv (example-vernacular)
B \xe (example-English)
B \xn (example-national)
\xr (example-regional)
\uv (usage-vernacular)
\ue (usage-English)
\un (usage-national)
\ur (usage-regional)
\ev (encyclopedic info.-vernacular)
\ee (encyclopedic info.-English)
\en (encyclopedic info.-national)
\er (encyclopedic info.-regional)
\ov (only [restrictions]-vernacular)
\oe (only [restrictions]-English)
\on (only [restrictions]-national)
\or (only [restrictions]-regional)

\lf (lexical function label)
\lv (vernacular lexeme referenced by the lexical function)
\le (lexical function-English gloss of referenced lexeme)
\ln (lexical function-national gloss of referenced lexeme)
\lr (lexical function-regional gloss of referenced lexeme)
\sy (synonyms)
\an (antonyms)
\mr (morphemic representation)

B \cf (confer/cross-reference)
B \ce (cross-reference-English gloss)
B \cn (cross-reference-national gloss)
B \cr (cross-reference-regional gloss)
\mn (main entry form)
\va (variant forms, e.g. dialect, etc.)
\ve (variant forms comment-English)
\vn (variant forms comment-national)
\vr (variant forms comment-regional)
\bw (borrowed word)
\et (etymology)
\eg (etymology-gloss)
\es (etymology-source) [default set not to print]
\ec (etymology-comment) [default set not to print]

\pd (paradigm set)
\pdl (paradigm label)
\pdrv (paradigm vernacular form)
\pde (paradigm form-English gloss)
\pdn (paradigm form-national gloss)
\pdr (paradigm form-regional gloss)
\sg (singular noun form)
\pl (plural noun form)
\rd (reduplication forms)
\ls (1st person singular verb form)
\ls (2nd person singular verb form)
\ls (3rd person singular verb form)
\ls (4th person singular verb form)
\ld (1st person dual verb form)
\ld (2nd person dual verb form)
\ld (3rd person dual verb form)
\ld (4th person dual verb form)
\lp (1st person plural-generic verb form)
\le (1st person plural-exclusive verb form)
\li (1st person plural-inclusive verb form)
\lp (2nd person plural verb form)
\lp (3rd person plural verb form)
To continue this discussion, see:

*Summary_of_Fields*

**Sections_in_a_Lexical_Entry**

*Sections in a Lexical Entry*

*(Understanding the hierarchical structure of an entry)*

MDF has two built-in hierarchical structures that should be flexible enough to meet most needs. (See "Alternate_Hierarchy" for a discussion of MDF's other hierarchy.) The field codes that mark the boundaries to lexical subsections are `\lx`, `\ps` (\pn), `\sn`, `\se`. Each of these sections or subsections can take a full set of field markers (except `\lc` and `\hm`, which should only occur at the top of the record).

*Multiple parts of speech*

Multiple parts of speech (`\ps`) are used to organize sections within an entry. A lexeme that fills more than one syntactic slot (as a noun, verb, etc.) should not be handled as homonyms. This is because the different syntactic functions (e.g. 'shower' (n) 'shower' (v) are still clearly related to each other in meaning.

```plaintext
\lx shower
\ps n
\de a light rain
\ps vt
\de to bestow special things on someone
```

MDF's standard printing function starts new `\ps` fields within an entry on a new line, preceded by an em dash to show that the lexeme form has not changed but its function has.

*shower*  *n.* a light rain.

—  *vt.* to bestow special things on someone.
Multiple Senses

MDF allows sense numbers (sn) to be used as another level of hierarchy within an entry. In the standard hierarchy, the sense number is lower than part of speech, and so multiple senses should be grouped under the relevant parts of speech to denote related but distinct meanings within a particular part of speech. (See "Alternate_Hierarchy" for a discussion of MDF's other hierarchy.) Multiple senses in each separate part of speech should start with '1' (as seen in the following, more complete entry for 'shower'):

```
\lx shower
\ps n
\sn 1
\de a light rain
\sn 2
\de a man-made device for dispensing water in droplets on a person; used for bathing
\sn 3
\de an event in which gifts are given to someone; as in baby showers and wedding showers
```

```
\ps v
\sn 1
\de raining lightly
\sn 2
\de to bathe using a device which causes water to dispense in droplets on a persons head; usually done standing up
\sn 3
\de to bestow special things on someone
```

This complex record would print as:

shower n. 1) a light rain. 2) a man-made device for dispensing water in droplets on a person; used for bathing. 3) an event in which gifts are given to someone; as in baby showers and wedding showers.

— v. 1) raining lightly. 2) to bathe using a device which causes water to dispense in droplets on a persons head; usually done standing up. 3) to bestow special things on someone.

Some lexicographers want to make fine distinctions between subsenses. These can be handled in MDF in the \sn field with a, b, c, etc. subcategorization.

```
\lx lexeme
\ps n
\sn 1a
\ge gloss
\de definition
\sn 1b
\ge gloss
\de definition
\sn 1c
\ge gloss
\de definition
\sn 2
\ge gloss
```
Which would have the general printed structure of:

lexeme n.
   1a) definition.
   1b) definition.
   1c) definition. 2) definition. 3) definition.

Using Subentries
Subentries (\se) provide a further level of hierarchy. These are commonly built around polymorphic forms in a root-based dictionary (see the MDF field manual (Coward and Grimes, 1995) for an extended discussion).

\lx bren
\ps vi
\ge play
\ee Implies lack of focus or purpose.

\se brenak
\ps vt
\ge play_s.t.
\de play a game, or play with s.t.

\se inabren
\ps n
\ge recreation ; entertainment

\se rabrenak
\ps n
\ge toy
\dt 17/Jun/92

This would print like the following:

bren vi. play. Implies lack of focus or purpose.
   brenak vt. play a game, or play with s.t.
   inabren n. recreation, entertainment.
   rabrenak n. toy.

A more complete example using subentries:

\lx bersih
\ps adj
\sn 1
\ge clean
\de be clean, not dirty or messy
\sn 2
\ge innocent
\de be innocent, without fault

\se kebersihan
\ps n
\ge cleanliness

\se membersihkan
\ps vt
\sn 1
\ge clean_up
A common question that all lexicographers must face when wrestling with complex (polymorphemic) lexemes is whether to incorporate them in the lexicon as subentries under the lexical entry of their root lexeme, or to place them in the lexicon as independent lexical entries. This is not an easy question to answer, but it is addressed in greater detail in Chapter 4 (section 4.6) of the MDF field manual (Coward and Grimes, 1995). For this database, we will summarize the two options below:

1. Using Subentries and Minimal Entries

If you want to keep all related (root and polymorphemic) lexemes together, use the subentry method shown below (this is a simplified example):

\lx bren
\ps vi
\ge play
\ee Implies lack of focus or purpose.

\se brenak
\ps vt
\ge play_s.t.
\de play a game, or play with s.t.

\se inabren
\ps n
\ge recreation ; entertainment

\se rabrenak

For information on MDF’s alternate hierarchy, see:

Alternate_Hierarchy

To continue with the general discussion, jump to:

Using_Subentries_or_Lexical_Entries
This would print as:

***bren*** *vt.* play. Implies lack of focus or purpose.

***brenak*** *vt.* play a game, or play with s.t.

***inabren*** *n.* recreation, entertainment.

***rabrenak*** *n.* toy.

This method allows you to see all of the related pieces close together. But you have to know where to look. And this is a major problem; outsiders will not know that to find a discussion of "inabren" they must look under "bren". To make this easier for new users, we recommend you also make extensive use of "Minimal Entries" (essentially the same as "minor entries") -- these are lexical entries that have only the most basic fields (maybe a \ps field and a gloss field), but they include the \mn "main entry" field which provides the sign-post back to the lexical entry where more information can be found.

The minimal entry for "inabren" might be:

```
\lx inabren
\ps n
\ge recreation
\mn bren
```

When printed the \mn field typically has the label "See main entry:"

That is all there is to this minimal entry. For this reason, it is expected that the subentry for "inabren" would look very much like a full lexical entry; it would contain verbose definitions, have example sentences, multiple senses (if needed), cross-referencing, and any encyclopedic information, notes, or comments that are needed to elucidate this lexeme, i.e. it would be as though a full lexical entry were stuck right inside another lexical entry. (The above example of subentry use obviously falls short of this expectation -- it is merely an example.)

But practically speaking, many find it difficult to actually put the same amount of effort into embellishing a subentry as they would into a main lexical entry. For this reason, and for the fact that such embellished records soon become enormous, you might find the alternative way better for you:

### 2. Lexical Entries and Cross-referencing

The alternative is nearly the exact opposite: make all lexemes (roots or polymorphemic forms) main lexical entries. But in order to see the relationships between the scattered pieces, you will need to use an extensive cross-referencing system. MDF provides the \cf and the \lf field bundles for this.

Again using the above example, the "bren" lexical entry would include the following fields:

```
\cf brenak
\ce play s.t.

\cf inabren
\ce recreation

\cf rabrenak
\ce toy
```
These `cf` fields reference full lexical entries for both "inabren" and for "rabrenak". If you wish to provide even further information regarding the relationship between the "bren" lexeme and these referenced lexemes, use the `lf` field (the `lf` labels shown here are not certain):

```
\lf Vgoal
\lv brenak
\le play s.t.

\lf Ngoal
\lv inabren
\le recreation

\lf Ninst
\lv rabrenak
\le toy
```

This strategy keeps the individual records smaller and still provides a way to see what all the related forms are and their general meanings and relationships. But for more detailed information about a related lexeme requires searching through the lexicon (easy in Toolbox, but tedious when in book form).

You will need to decide which strategy will work best for you.

For more information on sections in a lexical entry, see:

- `Sections_in_a_Lexical_Entry`
- `Alternate_Hierarchy`

For information on related fields, see:

- `mn` main entry form
- `cf` general cross-reference
- `lf` lexical function cross-reference

### Free-form_Fields

#### Free-form Fields

In the standard MDF printing, the following fields are "free-form" fields, meaning that MDF will not add periods, etc. to them; they are basically formatted as you entered them. Therefore, when using these fields, you are encouraged to use capitalization and proper punctuation. Free-form fields may be as long as you need, e.g. several sentences.

```
\rf reference
\vx vernacular example
\xe English example
\xn national example
\xr regional example
\ve variant form (English)
\vn variant form (National)
\vr variant form (Regional)
\ur usage (regional)
\or only (regional)
\er encyclopedic (Regional)
\tb table
```

The following fields are also free-form, except that the printing process will add
If a field is not explicitly marked in its description as being a "free-form" field, then it is not one.

If a field is not a free-form field, you do not need to use any terminating punctuation; MDF adds the needed punctuation to the end of each non-free-form field.

* The picture reference field is free-form so long as the field does not begin with .G. (which marks it as a graphics link paragraph). See \pc for more information on the picture reference field.

Range_Sets

Range Sets

Through the Range Set feature, Toolbox provides you with a powerful way to maintain consistent labeling on any field where you are using a relatively small set of categorizing codes. The most common fields for this are:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ps</td>
<td>part of speech</td>
</tr>
<tr>
<td>pn</td>
<td>part of speech (national)</td>
</tr>
<tr>
<td>pd</td>
<td>paradigm set</td>
</tr>
<tr>
<td>sd</td>
<td>semantic domain</td>
</tr>
<tr>
<td>is</td>
<td>index of semantics</td>
</tr>
<tr>
<td>th</td>
<td>thesaurus</td>
</tr>
</tbody>
</table>

You are strongly encouraged to use Range Sets for these fields also:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lf</td>
<td>lexical function</td>
</tr>
<tr>
<td>pdl</td>
<td>paradigm form label</td>
</tr>
</tbody>
</table>

Here are other fields which could benefit from the Range Set feature:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hm</td>
<td>homonym number</td>
</tr>
<tr>
<td>sn</td>
<td>sense number</td>
</tr>
<tr>
<td>bw</td>
<td>borrowed word (source of a loan word)</td>
</tr>
<tr>
<td>es</td>
<td>etymology source</td>
</tr>
<tr>
<td>so</td>
<td>source of data/information</td>
</tr>
<tr>
<td>st</td>
<td>status of the lexical entry</td>
</tr>
</tbody>
</table>

The easiest way to start a Range Set for a particular field, is to Right-Click on the field marker (option-click on the Macintosh) and select the "Range Set" tab from the Marker Properties dialog box that pops up. You can enter the labels directly (separated by a <space>) or use the Build Range Set feature. This will build a Range Set on the current
field marker using the contents of all occurrences of this field in all of the currently opened databases of this type. This is a great way to start a new Range Set on an existing database or to compile a new list at any time of what is currently in use (to see how consistent you are being, e.g. am I using : or _ to concatenate complex labels, did I add a new label when I already had one that is spelled almost the same, etc.).

*Do not build a Range Set on a field which does not have a closed set of possible contents.*

*Changing Range Set Labels (for the ps, lf, and pdl fields)*

MDF standard printing provides a way (for certain fields) to list English and National Audience substitutes for each label in the Range Sets. (The purpose of the English substitute allows users to consistently change the formatted output labels from the labels they use in their database, e.g. one-letter codes in the paradigm label fields could be changed to more readable abbreviations).

*This ability to make Range Set label substitutions is limited to the ps, lf, and pdl fields.*

*(If more fields need to be covered, contact Toolbox@sil.org.)*

*Note: with this feature, you no longer need to use the pn field to output national parts of speech. (But if you ever want to sort or browse on the national part of speech, you will still need to use pn.)*

These Range Set label substitutions are handled by the same CC tables that change the default MDF field labels. For more information on these CC tables, see:

*Printed_Field_Labels*

The areas in these CC tables that need "tweaking" are the special "group" sections, e.g. group(gPartOfSpeech), group(gParadigmLabel), etc. Be sure to follow the instructions given in these groups carefully. It is best to include a CC rule for every label in the field's Range Set. To make sure you don't miss any:

1. In Toolbox, Right-click on the field marker, to go to its properties sheet.
2. Click on the "Range Sets..." button
3. Click on "Build Range Set"
4. Use the mouse to select all of the Range Set labels in the edit window on the left.
5. Copy them to the Clipboard (<Ctrl-C>)
6. Open Notepad (or any text editor) and Paste in the labels (that is <Ctrl-V>),
7. Now you can print the labels or save them to a file.
8. Use Notepad (or whatever) to edit the appropriate CC table and add a CC rule for each label in your Range Set.

*Character_Style_Codes*

*Character Style Codes*

Character style codes can be used in any field (usually the information or discussion type fields) to uniquely mark the specific language of the text that you are discussing. For example, 'fv:' marks the following word as vernacular text:
This usage is a bit blunt. Most people don't say this directly, rather they refer to where they are going, e.g. fv:keta to refer to 'outside the village', but fv:-dai is used in familiar settings.

Character style codes are applied only to the following word. A space or any punctuation (except the '-') terminates the style. The style codes must be in lower case, and there must not be any space between the colon and the following word, e.g.

fv:sabun is okay
fv: sabun won't work (the fv: is treated as normal text)

Use the underline ( _ ) character to continue the style and link together words in a phrase, e.g.

In the phrase fv:Mala_sai_desy_de? there is a sense of anger or rudeness.

MDF will format this whole phrase with the vernacular character style (the underlined characters are converted to spaces). The question mark will not be included in the vernacular style.

Character codes should be placed with the word, inside parentheses, quotes, brackets, etc., otherwise the opening brackets will receive the character style along with the word.

**Basic Language and Character Codes**

fv: (font--vernacular)
fe: (font--English)
fn: (font--national language)
fr: (font--regional language)
fs: (font--standard)
fb: (font--bold)
fi: (font--italic)
uc: (underline character)
ub: (underline--bold)
ui: (underline--italic)

**Other Character Codes**

sc: (underline a scientific name--not required in the \sc field)

The uc: code is able to detect which type of field it is used in. If the field is a vernacular field uc: will underline with bold characters (following the vernacular character style); if the field uses the national language, uc: will underline italic characters; and if the field uses English, uc: will underline normal characters (to follow the character style for English). If you want to specifically control the underlining character style, use ui: and ub:.

If you need to tweak your formatted dictionary or reversed finderlists and apply a character style where you had forgotten to place a code, simply use the same character style that MS-Word uses to format other such text (e.g. the vernacular font in MS-Word uses the "f_vernacular" style). Just be sure to also add the missing character style code in your lexical database too, otherwise you'll have to "tweak" the output every time.

**Explicit Font Codes**

To explicitly control the character style or font of any phrase, word, or part of word, use
the underlying bar codes used by MDF. For example, to underline only a part of a word, use the \u bar code:
\xv Aulopo\u{a} au lae weidu.

The \u{} underlines the bracketed character ('a') only. Note that the curly braces {} can contain any string of letters, numbers, spaces, or punctuation.

The complete list of explicit font codes follows:

| fv( ) | Vernacular font |
| fe( ) | English font |
| fn( ) | National font |
| fr( ) | Regional font |
| fs( ) | Standard font (for general punctuation) |
| fl( ) | The font for field labels |
| fb( ) | Bold font |
| fi( ) | Italic font |

Still supported, but not encouraged, are the older ("Manuscripter") style codes. Underlining is more complex and so provides the following:

| ui( ) | General underlined character font (as shown above) |
| uc( ) | Specific regular underlined character font |
| ub( ) | Specific underlined, bold font |
| ui( ) | Specific underlined, italic font |

To format a homonym number in a discussion field you can use the explicit code:

| hm( ) | Homonym number font |

For example:
\ec This lexeme is now a homonym with fv:asw|hm(1) 'dog'
because proto *l ~ s in Selaru resulting in a loss of contrast.

*If you choose to use these explicit style codes, you must not forget the closing brace '}'*. If you do, the rest of the field will be considered a part of the specified font!

Actually, any field marker can become a character style code and transmit its formatting to the data enclosed in the curly brackets. This is mostly useful for the fields mentioned already (sc, hm) but any field code can be used.

### Punctuation_and_Special_Codes

**Punctuation**

Generally, leave off all punctuation at the end of bookkeeping fields and basic data fields (\hm, \sn, \ge, \lf, etc.). The only places where you need to include punctuation is in and at the end of free-form (discussion type) fields (\ue, \ee, \nt, etc.). In all other data fields, final punctuation is added by MDF automatically. For a list of free-form fields, see:
Free-form_Fields

Special Codes
Toolbox allows you to hard-code the following special characters directly into your lexical database (this works for both the RTF and the MDF exporting process):

- Inserts a Tab character
- Inserts an En-dash character
- Inserts an Em-dash character
  - Inserts a non-breaking space
  - Inserts an optional hyphen
- Inserts a non-breaking hyphen

For an example which uses the special code for Tab, see:

\tb                     Tables

For information on special codes for embedding character styles, see:

Character_Style_Codes

Printed_Field_Labels
Printed Field Labels
MDF adds field labels to many of the fields as it formats your lexicon for printing. These labels are explicitly noted in the discussion of each of the field markers. MDF allows for these field labels to be changed for either English or National audiences.

By default, MDF uses the following two CC tables to make the label substitutions:

MDF_eng.cct       (for English labels)
MDF_inz.cct       (for national (Indonesian)language labels)

MDF also comes with other label tables. For example, for French and Spanish:

MDF_frn.cct       (for French labels)
MDF_spn.cct       (for Spanish labels)

These CC tables are in the MDF folder, just under the Toolbox program folder.

To change a field label:

1. IMPORTANT: First, using File Manager or Explorer, to copy the CCT file you want to change to a new filename.
2. Next open the new CCT file in Notepad.
3. At the end of this file you will find a section marked as "group(gLabel)"; under here are all of the field labels that MDF adds to the beginning of fields. The following are a few
of the label-changing lines from one of these files:

"Ant:" > "Lawan:" c \an Antonym
"Read:" > "Baca:" c \bb Bibliography
"From:" > "Pinjaman:" c \bw Borrowed word (loan)

4. Carefully change each label in the "Lawan:" column that needs changing. Be careful you do NOT delete the " marks!
5. Save the file to the new name (be careful that Notepad doesn't stick on a .TXT extension on the file --if it does, rename it).
6. Now go back to Toolbox and the MDF Export-Options-Audience box to give the name of the new CC table you created.

In these CC tables there is also options for making consistent changes to your lexical function (\lf) labels, part of speech (\ps) labels, and paradigm form (\pdl) labels. This allows you to keep your Range Sets labels for these all in one language, but when the dictionary is printed for another audience, the labels are changed appropriately.

For a detailed description of the MDF formatting process, see:

*Formatting_and_Printing*

For related information on how MDF can also substitute Range Set labels, see:

*Range_Sets*

**Unknown_Fields**

*Unknown Fields*
Any field marker in a lexical database that is not recognized by MDF is considered an unknown or Non-MDF Field.

Normally MDF discards unknown fields from the formatted output.

While it is always best to use available fields where possible (for the sake of functionality and compatibility), if you should need to create a field not covered by the 102 MDF fields (say, for some special bookkeeping or analysis chore), you may do so without worry. As a general rule, any field marker of three or four letters is a non-MDF field. The only exceptions to this rule are the new paradigm fields: \pdl, \pdv, \pde, \pdr, and \pdn.

*If you want MDF to include non-MDF fields in the formatted output, simply check "Include non-MDF fields" under Options in the MDF Export dialog box.*

All unknown fields are collected together and placed at the end of the section they occur in. They are bracketed with [ ] and marked with the label: ??.

**Formatting_and_Printing**

*Formatting and Printing Your Dictionary and Finderlists*
So your lexicon is ready, and you want to print it. This assumes you've set up the database to be an MDF type database. (An MDF type database not only uses the MDF marker set but has checked the box in Database-Properties-Options labeled "Multi-Dictionary Formatter").
Specifying the Options

To print your lexicon, choose "Multi-Dictionary Formatter" from the File-Export menu. Here you will be able to select the intended audience (either English or national), the format (diglot, triglot, or a reversal finderlist), specify text for the footer for each of these choices, select a filter (to limit the formatted document to a smaller set of records), and specify whether you want Word opened automatically (for Windows users).

This dialog box also has an Options button which will allow you to set the basic parameters the standard MDF print feature uses to create the formatted dictionary or finderlists. These options include selecting specific fields to include or exclude and some basic page formatting.

Clicking Export gives you a chance to specify a filename and location for the new document. Clicking OK again starts the whole process. MDF goes through several steps to format the document (detailed below). The final product is a Rich Text Format (RTF) file.

Open the RTF file in Word

If you checked "Automatically open document in word processor", Toolbox causes the word processor to automatically open the exported document; otherwise you must open the RTF file yourself.

Finish the process of exporting from Toolbox

To make the final formatting touches, choose the menu option "Finish exporting from Shoebox" from under the Tools menu in Word. This runs the FinishExportingFromShoebox macro in the document template. It updates style formatting and does a few other special formatting operations. It also saves the exported file as a Word DOC file.

Note: Word processor programs other than Microsoft Word may not be able to use the Word macro, and may even require a different template than is supplied with the standard MDF printing process.

Note: Probably the most significant thing the Finish exporting from Shoebox macro does is to load all the pictures.

If you need to change any style formatting, click on the text that needs changing, choose Format, Style, and Modify. Make the changes needed, and remember to check the "Add to Template" box before you choose OK so the style changes will be available the next time you export.

Print it

If the document looks fine, choose File Print.

Note: Errors in the data should be corrected in Toolbox (and export again). If you correct errors in the Word document, they will still be there in the data when you export the next time.

This concludes the general discussion of how to use MDF.
### Summary_of_Fields

#### Summary of Fields (not ordered)

The following are all of the fields MDF recognizes, listed in their suggested groups:

('f' marks free-form fields)

---

**RECORD MARKER**

\lx lexical entry (only one allowed per record)

**BASIC FIELDS**

- \ps part of speech
- \pn part of speech (natnl)
- \ge gloss (English)
- \gn gloss (national)
- \re reversal (English)
- \rn reversal (national)
- \de definition (English)
- \dn definition (national)
- \f rf reference to notebooks
- \f xv example (vern.)
- \f xe example (English)
- \f xn example (national)
- \cf cross reference
- \ce cross ref. (English)
- \cn cross ref. (national)
- \f nt notes, etc.
- \f dt datestamp

**OPTIONAL FIELDS (Cont.)**

- \li 1st plural incl. form
- \2p 2nd plural form
- \3p 3rd plural form
- \4p pl. non-human form
- \ph phonetic form
- \cr cross ref. (regional)
- \mr morphemic form
- \rd reduplication form
- \va variant form
- \ve variant (English)
- \vn variant (national)
- \vr variant (regional)
- \mn main entry form
- \lf lexical function
- \lv lexeme ref’d by lexical fnct
- \le lexical fnct (English)
- \ln lexical fnct (national)
- \lr lexical fnct (regional)
- \uv usage (vernacular)
- \ue usage (English)
- \un usage (national)
- \ur usage (regional)
- \ov only (vernacular)
- \on only (national)
- \or only (regional)
- \ev encyclo. (vern)
- \ee encyclo. (Engl)
- \en encyclo. (Natnl)
- \bw borrowed word
- \lt literal meaning
- \eg etymology (gloss)
- \es etymology (source)
- \bc etymology (comment)
- \sd semantic domain
- \is index of semantics
- \th thesaurus
- \bb bibliographic ref.
- \sc scientific name
- \tb table/chart
- \np notes on phonology
- \ng notes on grammar

**RESERVED FIELDS**

- \hm homonym number
- \an antonyms
- \lc lexical citation
- \f uv usage (vernacular)
- \f un usage (national)
- \f ur usage (regional)
- \ov only (vernacular)
- \on only (national)
- \or only (regional)
- \ev encyclo. (vern)
- \ee encyclo. (Engl)
- \en encyclo. (Natnl)
- \bw borrowed word
- \lt literal meaning
- \eg etymology (gloss)
- \es etymology (source)
- \bc etymology (comment)
- \sd semantic domain
- \is index of semantics
- \th thesaurus
- \bb bibliographic ref.
- \sc scientific name
- \tb table/chart
- \np notes on phonology
- \ng notes on grammar
To continue this discussion, see:

Sections_in_a_Lexical_Entry

Alternate_Hierarchy

A Sense-Oriented Hierarchy
The alternative hierarchy that MDF supports differs from the standard hierarchy in its ordering of these three crucial field markers: \sn \se \ps. (The standard hierarchy expects the order \ps \sn \se.) The alternate hierarchy’s ordering can be outlined as follows:

\lx lexeme
  \sn sense number
  \se subentry
    \ps part-of-speech
    \ps part-of-speech
    \se subentry
    \ps part-of-speech
    \ps part-of-speech
  \sn sense number
  \se subentry
    \ps part-of-speech
    \ps part-of-speech
  \se subentry
    \ps part-of-speech
  \se subentry

The main purpose of this alternate hierarchy is to allow the user to group or base subentries on a given sense. A simplified example of this would be:

\lx ad\ge throw ; discard
\sn 1
\de throw
\se mongad\ps vt
\ge throw away ; discard
\de throw something away; to discard
\se iad\ps vt
\de throw something away
\se paad\mr po-ad \ps vCAUS
\de drop
\se puad\mr pu-ad\ps v
\de throw away (thoughtlessly or irresponsibly)
\sn 2
\de divorce
\se muad\mr m-pu-ad\ps vREC
\de divorce smb (lit. throw each other)

\se puad\mr pu-ad\ps vREC
\de be divorced; separated (used for divorce and separation of man and wife)

\sn 3
\de defecate

\se paad\ps vCAUS
\de defecate

This example would typically print as follows (shown without the character styles here):
ad\ through, discard.
   1) throw.
      mongad\vt. throw something away; to discard.
      iad\vt. throw something away.
      paad\Morf: po-ad\vCAUS. drop.
      puadstd\v. throw away (thoughtlessly or irresponsibly).
   2) divorce.
      muad\Morf: m-pu-ad\vREC. divorce smb (lit. throw each other).
      puad\Morf: pu-ad\vREC. be divorced; separated (used for divorce and separation of man and wife).
   3) defecate.
      paad\vCAUS. defecate.

Note that by judiciously using the definition (\de) field you can give labels to the sense groupings.
This method of organization was urgently requested by people working in languages that were highly polymorphemic, because this hierarchy allows them to group complex polymorphemic lexemes together based on a sense (rather than stand alone -- as subentries do in the standard MDF hierarchy).

Currently MDF can only support one method of organization per lexical database. Therefore you

must choose which hierarchical method will best meet your needs and use only that one method for all lexical entries in you dictionary. Mixing methods in Shoebox and MDF can produce inaccurate interlinear choices and browse views, as well as odd formatting in the output.

To help with the decision, note that there are advantages and disadvantages to using this alternate method:
• One main advantage to this structure is that you can see all the related polymorphemic forms that carry a particular sense.
• One disadvantage of this structure is that you do not readily see the actual range of senses that a particular polymorphemic form can express.

For important information on how MDF handles sections in a lexical entry as well as
related information on the standard hierarchy, please see:
Sections_in_a_Lexical_Entry

For information on the order in which fields are formatted for printing, see:
Order_of_Fields

To continue on with the general discussion, jump to:
Using_Subentries_or_Lexical_Entries

Also see the document "MDF-ch2 revised.doc", pages 55-63 for a detailed discussion of the two different hierarchies.

The_MDF_Documentation

The MDF Documentation
The original MDF manual, "Making Dictionaries: A guide to lexicography and the Multi-Dictionary Formatter", by Coward and Grimes, was published in 1995 and gave instructions for a set of independent CC tables running in the DOS operating system. While still an extremely valuable document, there are parts which are no longer relevant, and some which have been changed. The manual itself is only available as a PDF file, though the revision of chapter two is a Word document. They can be obtained from the MDF page of the SIL website.

The following are suggestions for reading the manual:
Chapter 1 - Skip (specifics of installation of the original DOS version)
Chapter 2 - see revised version: MDF-ch2-revised.doc. Has a description of all MDF markers. Ignore references to "older approaches". The section on gloss vs reversal vs definition is useful.
Chapter 3 - Skip. Specifics of running the original DOS version.
Chapter 4 Theory, which needs to be considered.
Chapter 5 - Read section 5.3 about categories of markers. The rest is mostly about why to use Shoebox instead of Word, and about sorting, which is done differently in Toolbox.
Chapter 6 discusses: principles for choosing the headword, example sentences, homonym vs different senses, semantic categories, dialect information. Note, sometimes gives alternative ways to handle the same information. Check with a dictionary consultant if you have doubts.
Chapter 7 Various possible lexical function relationships. Usefulness of semantic relationships as an approach for eliciting data.
Chapter 8 Lots of good theory on making good entries; good questions to ask. Special considerations for activities & events, states & processes, loans & etymologies, ritual speech, etc.
Chapter 9 Discussion on part of speech.
Chapter 10 A dictionary is more than a list of words. Discusses preface material, appendices, etc.
Appendices A-E include lists of markers, recommended order of fields, lists of semantic domains and lexical functions, and some commonly used abbreviations.
Appendix F-I - Skip. Specific to the original version. Not relevant now.
**Old_and_Changed_Markers**

**Old and Changed Markers**

Within a couple years of the publication of the MDF manual in 1995, as the number of users grew and became more widespread, there were recommendations which modified the MDF marker set. These changes were reflected in the modification to the MDF manual, "MDF-ch2 revised.doc".

In this database, some of the long-since un-recommended markers have been removed from prominent display. (The MDF tables have not been changed.) In addition, the database type distributed with the Toolbox Training Materials and the Toolbox New Project Installer no longer includes these markers. These should all be replaced by the paradigm markers.

\1s \2s \3s \4s \1d \2d \3d \4d \1p \1i \1e \2p \3p \4p

For example,

\1s yoban

should be replaced by

\pdl 1s
\pdv yoban

In this example, the label which will be used in printing is placed in the \pdl (paradigm label) field. The paradigm form itself is placed in the \pdv (paradigm vernacular form) field. The paradigm approach has much greater flexibility, both in the choice of labels and in allowing glosses in other languages.

In addition, the following are still included in the MDF database type file but could be replaced by the paradigm approach.

\sg Singular form \pl Plural form \rd Reduplication form

Similarly, these markers could use the lexical function approach.

\an Antonym \sy Synonym

For examples and more details, see

\lf the lexical function label
\pd paradigm set
\pdl the paradigm label

**When_MDF_Fails_to_Meet_Your_Requirements**

**When MDF Fails to Meet Your Requirements**

This topic is still under construction. Terse notes on common problems and answers are below. Contact Toolbox@sil.org for more details or other problems.

**Typical problems**

appearance of the printout -- modify the template in Word, check "add to template" to keep changes for the next export; check Page Setup under Options for columns, margins, odd-even -- these are not included in the template

labels in the wrong language -- check the set of files mdf_XXX.cct to see if the language you want is available. if not, modify the mdf_eng.cct and save as an appropriate name, then select that file by Modifying the Audience options.

labels in the wrong script -- modify the template in Word, check "add to template" to keep changes for the next export

label needed after the field, not preceding -- contact Toolbox@sil.org. A more flexible
system is coming
punctuation is not what is wanted -- punctuation is very difficult to modify. Contact Toolbox@sil.org. A more flexible system is coming.
some fields not included in the printout -- check list of markers to exclude; check Options settings for various "include..." options; check Audience selection; check diglot / triglot selection; check Finderlist selection
sorted order of the printout -- check language encoding assignment of \lx and \lc in MDF.typ and also in MDF_RTF.typ, which MDF printing uses for final sorting. (Do Project, Database types in order to check MDF_RTF.typ.)
fields are printed in the wrong order -- MDF standard printing reorders all fields according to a specific pattern; a version which doesn't reorder is available. contact Toolbox@sil.org for help on this. (Soon to be made widely available.)
marker NOT to be printed, not available -- no problem, create a new marker and use it for whatever notes or questions you are wanting it for. Also, read the NOTE ON ADDING MARKERS at the end.
marker to be printed but NOT FINAL printing, not available -- create the new marker, check "include non-MDF field" under options and be sure it is in the list of fields included (Select Fields to be Excluded button in Options). Note, such fields will be printed at the end of the entry. Also, read the NOTE ON ADDING MARKERS at the end.
marker for final printing not available -- be sure you understand what *is* available (read the descriptions of the markers). MDF standard printing puts all non-MDF fields at the end. If that's OK, be sure it's included for printing ("Select Fields to be Excluded" button in Options). Also, read the NOTE ON ADDING MARKERS at the end.
need markers for more scripts than are available, multiple scripts for the same field (eg, lexeme) -- if it's only the lexeme (perhaps entered in Roman but you want to print in script) AND you are not using the lexical citation option (\lc field), then the \lc field can be used for the script form of the lexeme. Similarly, if you don't have a regional language, you can use the various regional markers to provide another script (Actually, you may have to shuffle the meanings of the various "languages" in order to get the order you want.) -- these tricks are not really encouraged, but can be made to work. If these don't meet your needs, contact Toolbox@sil.org; a more flexible system is being developed. Also, read the NOTE ON ADDING MARKERS at the end.

NOTE ON ADDING MARKERS:
The MDF marker set is a standard recognized by various organizations. Additional markers should not be added without good motivation. On the other hand, it is also recognized that the marker set does not cover all possible reasonable needs. If you are going to publish electronically or if you are going to archive your dictionary through some organization, be sure they understand and approve the marker set you are using. Also, include your database type file with the archive and be sure your new markers have a good description. If you have legacy data, be sure that the archiving or publishing organization has the appropriate font(s) and language encoding file(s).

References
New: Jan 2020 MDF Fields Alphabetical

an
\an antonym
This field is used to reference an antonym of the lexeme.
MDF's standard printing adds the label "Ant: " to this field.
Using the \lf (lexical function) field for this is a better practice. For example,
\an sal

could be done instead as
\lf Ant.
\lv wrong, false

The latter gives more information to the outside reader by providing a gloss. (National and regional glosses are available also.)
For more detailed information on this, see:
\$sy synonym
\$lf lexical function

<Optional>

bb
\bb bibliographic reference
This is used to record any bibliographic information pertinent to the lexeme. MDF adds the label 'Read:' to this field.
<Optional>

bw
\bw borrowed word
This is for denoting a borrowed word or the source language. There is no standard way such information should be encoded, but generally the following is most common:
\bw Arabic

Which would typically print as:
From: Arabic.

<Optional>
**ce**

\ce cross-reference (English gloss)
This gives the English gloss for the vernacular lexeme referenced by the preceding \cf field.
For more information, see:

\Ye \cf confer/cross-reference

Identical national and regional language gloss fields are:
\Ye \cn and \Ye \cr

**<Basic>**

**cf**

\cf confer/cross-reference
This is a generic reference marker used to link together any two related entries in the lexicon. For example, in Selaru, '-aswasw kaha' means 'high water mark' and needs to cross-reference 'manahma' (the entry for 'rising tide') and visa-versa.
The \cf field is bundled with the \ce, \cn, and \cr gloss fields, so this example would be encoded as:
\cf manahma
\ce rising tide
\cn air naik
\cr

and would typically print in a triglot dictionary as:

See: manahma 'rising tide'; air naik'.

Multiple cross reference bundles are usually concatenated with semicolons ';' as seen in:
\cf manahma
\ce rising tide
\cn air naik
\cr
\cf mety2
\ce low tide
\cn
\cr
\cf rean
\ce high tide
\cn
\cr

This would typically print in a diglot dictionary as:

See: manahma 'rising tide'; mety2 'low tide'; rean 'high tide'.

(Note that actually the vernacular text is usually be printed as bold, and the '2' for the homonym number of 'mety' as subscripted, but these can't be displayed that way in Toolbox.)
One major short coming of using the cross-reference fields (rather than the lexical
function fields) is that the semantic relationship between the two related lexemes is not made explicit. But it is often the case that a researcher will know there is some kind of relationship between two lexemes but is unclear as to the nature of that relationship. This is what the \cf field is good for. And once the relationship is determined, the cross-reference information could then be transferred to an \lf field bundle.

It is not uncommon for there to be many lexemes that are interrelated. This creates a myriad of cross-references, where each lexeme of a given type is cross-referenced to all other lexemes of that type. Adding new entries to the group can also be very tedious. A simple solution to this is to choose one lexeme as the focal point and have all other related lexemes refer only to that one. That focal lexeme will then contain a listing of all other lexemes in the group, by using either a series of \lf (lexical function) bundles, to describe the relationships between that focal lexeme and each of the other lexemes, or by using the \tb (table) field to make an actual list of the other lexemes and their meanings.

For more on grouping cross-references or mapping related lexemes, see:

\lf lexical function label
\tb table

For the English, national and regional glossing fields, see:

\ce, \cn and \cr

<Basic>

\cn cross-reference (national gloss)
This gives the national language gloss for the vernacular lexeme referenced by the preceding \cf field.
For more information, see:

\cf confer/cross-reference

Identical English and regional language gloss fields are:

\ce and \cr

<Basic>

\cr cross-reference (regional gloss)
This gives the regional language gloss for the vernacular lexeme referenced by the preceding \cf field. MDF formats the \cr field with quotes inside square brackets.
For more information, see:

\cf confer/cross-reference

Identical English and national language gloss fields are:
ce and cn

<Optional>

de
\de definition (English)
This field is used to fully express the semantic domains of each sense of a lexeme in English. For related information, see:
  \ge gloss (English)

*Hint*
If an entry has many English glosses, none of which are vitally important to differentiate while glossing texts, give only one gloss in the \ge field (to simplify interlinearizing) and then give all of the glosses in the \de and \re fields. When the dictionary is formatted, the \de field will be used and the \ge field will be discarded. Also, because there are \re fields, the \ge field will be ignored for reversing the database.
For example, the Indonesian morpheme '-nya' means 'his' 'hers' or 'its', but for glossing '3sPOS' may be adequate (if not preferable), but '3sPOS' is awkward for both a dictionary printout and an entry in a reversed English finderlist. So, do something like this:

\ge 3sPOS
\re his ; hers ; its
\de his, hers, its

This will simplify interlinearizing by keeping Toolbox from asking which gloss ('his', 'hers', or 'its') is appropriate for each occurrence of '-nya' in a text, and yet will allow your lexical database to be more complete in its description of the meaning or semantic domain of the lexeme.
Identical vernacular, national and regional fields are:
  \dv, \dn, and \dr

Closely related fields are:
  \ue, \oe, and \ee

<Basic>

dn
\dn definition (national)
This field is used to fully express the semantic domains of each sense of a lexeme in the national language.
For more information, see "Hint" under:
  \de definition English
See also:
  \gn gloss national
For the related definition fields, see:
  ¥dv, ¥de, and ¥dr

Closely related national language fields are:
  ¥un, ¥on, and ¥en

**<Basic>**

**dr**

*dr definition (regional)*

Often early in the project the definitions we receive are actually in the regional language, not the national language; such definitions could be stored here. If this field is included in the output, MDF adds a label and brackets "[ ]" around it and it is treated as part of the national field.

For more information, see "Hint" under:
  ¥de definition (English)

See also:
  ¥gr gloss (regional)

For the related definition fields, see:
  ¥dv, ¥de, and ¥dn

Closely related regional language fields are:
  ¥ur, ¥or, and ¥er

**<Optional>**

**dt**

*dt datestamp*

A Toolbox field to help you keep track of the last time you edited an entry. There need only be one of these in a record (usually the last field) and is usually inserted automatically by Toolbox. The field is set up under the Toolbox menu option: Database-Properties-Options tab. The field marker must be defined first before it can be selected as the datestamp field.

*This field does not normally print.*

**<Basic>**

**dv**

*dv definition (vernacular)*

This is the hardest of all fields to fill in, because it requires the researcher to explain, in
the vernacular, what the salient concepts are (i.e. the domain) that this "unit of meaning" captures -- from a native speaker's perspective. Obviously not easy and something usually left to much later.
For the related definition fields, see:
$de, dn, and dr$

This field is also used for creating a monolingual dictionary. For related monolingual fields, see:
$gv, uv, ov, and ev$

<Optional>
**ec**

(ec etymology-comment)
Any comments the researcher needs to add concerning the etymology of the lexeme can be given here.
For more information, see:
$et$

*This field does not normally print.*

*This is a "free-form" field. Punctuation and capitalization should be used as needed.*

<Optional>
**ee**

(Encyclopedic information (English))
A field researcher has an incredible opportunity to assimilate intuitive knowledge of the language and culture of the people with whom he/she works. This type of information is invaluable to others who have no access to the language area (or at least no extended contact with the language community).
This knowledge is often never codified. Now Toolbox provides a simple means by which a researcher can "put on paper" this kind of information. Once added, the lexical database becomes more than a "dictionary" but a "knowledge-base" of the language.
This field crosses over with the $de, ue, and oe$ fields, but is intended for more verbose explanations of the lexeme (headword, subentry or sense). Basically, the researcher should use this field to encode any additional information needed by a non-native speaker to understand and use this lexeme properly. For example, the lexeme "hatw" (in Selaru) can be described with the following fields:

\de a fired, earthen, baking form used for making $fv$:skyerker
\ee Often today this is replaced by a cast iron griddle, much like a waffle iron, which is frequently found in the shape of four hearts.

MDF does not add any label to this field, but simply formats it as entered.
Tip
Since MDF does not format or add any label to this field, you can actually use this field to include additional information about a lexical entry which is not handled by any of the other fields that MDF supports. You can even add your own label to this information. It’s not pretty, but it works. For example, if you wanted to create a new field to keep track of the weather (a hypothetical example to be sure), you could enter your observations thus:
\ee |fl{Weather: }The sun is hot with scattered showers expected.

The special character code " |fl{ }" tells MDF that the data contained in the curly braces needs to be formatted as a label. The rest of the field is treated just like a normal encyclopedic field.

*Note: Using the encyclopedic fields in this manner should not be a common practice, as this will make it impossible for Shoebox or any other program to know exactly what kind of data is contained in your encyclopedic fields. But in a pinch, it does allow for this kind of flexibility.*

Identical national and regional language fields are:

\en and \er

Related fields include:
\de, \ue, and \oe

For more on special character formatting codes, see:
Character_Style_Codes

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

**eg**
\eg etymology-gloss
The published gloss for the etymological reference is given here.
For more information, see:
\et

<Optional>

**en**
\en encyclopedic information (national)
The national language equivalent to the \ee field. This field should cover information that provides a more complete knowledge-base on the lexeme.
For more information, see:
Closely related fields are:
¥dn, ¥un, and ¥on

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

¥er
¥er  encyclopedic information (regional)
The regional language equivalent to the ¥ee field. This field should cover information that
provides a more complete knowledge-base on the lexeme.
MDF adds the brackets "[ ]" around this field. If included in the output, it is treated as
part of the national field.
For more information, see:
¥ee

Closely related fields are:
¥dr, ¥ur, and ¥or

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

¥es
¥es  etymology-source
The reference or source abbreviation for etymology of the lexeme is given here.
For more information, see:
¥et

This field does not normally print.

<Optional>

¥et
¥et  etymology
The etymology for the lexeme is put here, e.g.:
¥et  *babuy

This field is bundled with the following fields:
¥eg  etymology-gloss
etymology-source [doesn't print normally]
etymology-comment [doesn't print normally]

A full example would be:
\et *ha[?]ng
\eg breathe loudly
\es PANDYMC
\ec *s > /h/? Not a likely cognate.

Which would print as either:
*ha[?]ng 'breathe loudly'.

Or:
*ha[?]ng 'breathe loudly' PANDYMC (*s > /h/? Not a likely cognate.)

Depending on whether you have changed the default print settings to include the \es and \ec fields or not.
You must explicitly include the asterisk (*) if you want it printed. The print tables will not add it.

<Optional>

encyclopedic information (vernacular)
This field contains the vernacular description of any pertinent encyclopedic information related to the lexeme or headword, subentry, or sense. This is intended for use in a monolingual dictionary, but can be used in diglot and triglot dictionaries as well.
For more information, see:

Related fields are:

ge gloss (English)
This is intended for interlinear morpheme-level glossing. It is used for reversing the dictionary if an \ev field is not present (or is present but empty). It is also used as an English definition in the printed dictionary if there is no \de field (or it is present but empty).
The user may enter data either with each gloss getting its own field:

ge put_out
ge move
The underline character '_' in the example glosses above is to force Toolbox to treat the multiple word gloss 'put out' as a single gloss. When printing a dictionary, the underlines normally will be converted to spaces.
Similarly, a dot ("period" or "full stop") can be placed between the words instead of the underline:
\ge put.out
\ge move

An older style allows you to list the glosses strung together, separated by a semicolon with a space on either side of it (so as to be unlike any normal punctuation):
\ge put_out ; move

Toolbox can recognize either format and when interlinearizing will give the user the choice of both glosses in either case. The sequence ' ; ' is also converted to ', ' by MDF printing when formatting a dictionary or finderlist.
The advantage of keeping your glosses in separate fields is that if you setup Toolbox to sort on the \ge fields (to get a type of finderlist), it will sort on all of the glosses; but if the glosses are all concatenated, Toolbox will only sort on the first form.
For important related information, see:
¥de definition (English)
¥re reverse form (English)

<Basic>

\gn
gn gloss (national)
This is intended for interlinear morpheme-level glossing. It is used for reversing the dictionary if an \rn field is not present (or is present but empty). It is also used as a national language definition in the printed dictionary if there is no \dn field (or it is present but empty).
The user may enter data either with each gloss getting its own field:
\gn keluarkan
\gn pindahkan

Or with the glosses strung together, separated by a semicolon with a space on either side of it (so as to be unlike any normal punctuation):
\gn keluarkan ; pindahkan

Toolbox can recognize either format and when interlinearizing will give the user the choice of both glosses in either case.
MDF will convert the ' ; ' sequence to ', ' when formatting the dictionary. (MDF will also automatically convert any underline character '_' in a gloss field to a space when formatting.)
For important related information, see:
¥dn definition (national)
¥rn reversal form (national)
<Basic>

**gr**

\textit{gr gloss (regional)}

This helps clarify the national language gloss in the prominent regional language of the area. This is also used as a regional definition in the printed dictionary if there is no \textit{dr} field present (or it is present but empty).

The user may enter data either with each gloss getting its own field:

\begin{verbatim}
\gr kasi_pinda
\gr kasi_pulang
\end{verbatim}

Or with the glosses strung together, separated by a semicolon with a space on either side of it (so as to be unlike any normal punctuation):

\begin{verbatim}
\ge kasi_pinda ; kasi_pulang
\end{verbatim}

Toolbox can recognize either format and when interlinearizing will give the user the choice of both glosses in either case.

The underline character \texttt{'_'} in the example glosses above is to force Toolbox to treat the multiple word gloss 'kasi pinda' as a single gloss. MDF will convert underline characters in gloss fields to spaces automatically. The sequence \texttt{' ; '} is also converted to \texttt{', '} by MDF when formatting a dictionary or finderlist.

If this field is included in the output, MDF adds a label and brackets "[ ]" around it and it is treated as part of the national field.

For important related information, see:

\begin{verbatim}
¥dr definition (regional)
¥rr reversal form (regional)
\end{verbatim}

<Optional>

**gv**

\textit{gv gloss (vernacular)}

This field is available for the development of a monolingual dictionary. And while this may seem somewhat meaningless (a vernacular gloss for a vernacular word), it actually can be a useful place to store the simple explanations the researcher is given by a language assistant concerning the meaning of the headword or lexeme. These can then be formulated into a more exact definition and transfer to the \textit{dv} field. It could also be used for those lexemes which can actually be defined with short glosses. It could also serve as a subset to the information covered in \textit{dv} field.

See:

\begin{verbatim}
¥dv
\end{verbatim}

<Optional>
hm

\hm  homonym number
This is field is used to differentiate homonym entries (lexemes that sound or are spelled the same but have no semantic relationship). This field generally comes directly after the \lx field and is simply followed by a 1, 2, or 3, etc.:
\lx asw
\hm 2
\ps n
...
(note the lack of punctuation in the \hm field)
Any cross-reference to one of these entries should also include the homonym number, e.g.:
\cf asw2

When the lexical database is converted to MS-Word format for printing, the homonym number for both the entry and the cross-reference will be subscripted by MDF using the homonym number character style.
For more on character styles, see:
   Character_Style_Codes

<Reserved>

is

\is index of semantics
"Beyond key terms: a lexicon useful for translation" (Rick Nivens, manuscript) discusses the use of Louw and Nida's (1988) Greek-English semantic domain categories. While Nivens proposes another way of doing this in his paper, it is also possible to use this \is field to catalog lexical entries according to these semantic domains. Reversing on this field would then yield semantically related entries (in relation to the New Testament). One word of caution: this is an etic approach, i.e. Greek semantics will rarely line up exactly with the vernacular domains. This field is to be a tool, and no attempt should be made to "force-fit" lexemes into pre-defined domains.
If selected for output, MDF adds the label "Semantics: " to this field.
For related fields see:
   ¥sd  semantic domain
   ¥th  thesaurus

For references see:
   References

This field does not normally print.

<Optional>
**lc**

**lc lexical citation**
This should be added only if the lexical entry form is inappropriate for the printed dictionary, and you want to substitute another form for the printed entry, e.g. you might want the entry "lewat" printed as "l趡t" if its stress pattern were not predictable. Roots are commonly used as the lexeme form in Toolbox lexical databases, but if the language affixes prepositionally, a dictionary printed with root forms as the entry form can be very confusing to native speakers. In this case, you could choose a consistent conjugated form (e.g. 3s-verb form) for a citation form. MDF will always replace the `lx` field with the `lc` field, if present, and then resort the dictionary according to these fields. (You can choose to restrict MDF to sort the database only by the `lx` field, even for those entries with an `lc` field. This restriction will not effect the `lc` field though, an `lc` field will still print as the entry form for its record. Choosing to restrict MDF to the `lx` field may cause some citation forms to appear out of sequence.)

<Reserved>

**le**

**le lexical function (English gloss)**
This is for giving the English gloss of the vernacular lexeme referenced by the lexical function.
For more information, see:

- ¥lf lexical function label
- ¥lv vernacular lexeme referenced by the lexical function

Related fields are:

- ¥ln and ¥lr

For generic cross-referencing, see:

- ¥cf confer/cross-reference

<Optional>

**If**

**If lexical function label**
For encoding the semantic networks of a language. The `If` field bundle includes the following fields:

- ¥lf the lexical function or relationship label
- ¥lv vernacular lexeme referenced by that lexical function/relation
- ¥le English gloss of the vernacular lexeme
- ¥ln national gloss of the vernacular lexeme
- ¥lr regional gloss of the vernacular lexeme
Here is a typical example:
\lf Syn
\lv -dew
\le chop on end
\ln
\lr

(Note that empty fields are okay.)
This lexical function would be typically print as:

   Syn: -dew 'chop on end'.

If the other glosses were filled in, they would be included in the printout (if a triglot dictionary was requested). Multiple lexical function bundles are concatenated with a semicolon ';', e.g.:

   Syn: -dew 'chop on end'; Syn: -sin 'split'.

Note: In the original MDF manual and even in the revision of Chapter 2 you will see examples in which the label and the vernacular word were both in the \lf field. For example, \lf Syn = -dew. This approach is now strongly discouraged. The user is strongly encouraged to use the \lv field as shown in the examples in this record.

The recommended MDF method for handling lexical functions (using the \lv field) has some distinct advantages over the old way:
1) Since the lexical function label sits by itself in the \lf field, you can use the Toolbox Range Set feature to maintain consistent labeling.
2) Using \lf and \lv fields with the sort and jump features of Toolbox is much easier.
3) Browsing on a database sorted by the \lv and \lf fields and displaying the \lv, \lf, and \le fields will give a virtual \lv reversal view of your data -- allowing you to see all of the lexical functions that any given lexeme has been categorized under.
4) New Toolbox features like integrity checks, range-set building, and word lists will not work properly on the \lf field if it contains mixed data.
5) If the vernaculalr language is written in a non-Roman script, the lexeme and lexical function label need to be in different fields since they have different language encodings.

See also:
\lv vernacular lexeme referenced by the lexical function label

For more information on Range Sets, see:
   Range_Sets

For an alternative way to create a table of related lexical items, see:
\ytb table

For generic cross-referencing, see:
\ycf confer/cross-reference
For more detailed information on language analysis through lexical functions, see the MDF field manual (Coward and Grimes, 1995) and the article "Mapping a culture through networks of meaning" (Grimes 1987). For referential information on these, see:

References

<Optional>

In

In lexical function (national gloss)
This is for giving the national gloss of the vernacular lexeme referenced by the lexical function.
For more information, see:

\$lf\  \text{lexical function label}

\$lv\  \text{vernacular lexeme referenced by the lexical function}

Related fields are:

\$le\  \text{and}  \$lr

For generic cross-referencing, see:

\$cf\  \text{confer/cross-reference}

<Optional>

Ir

Ir lexical function (regional gloss)
This is for giving the regional gloss of the vernacular lexeme referenced by the lexical function.
For more information, see:

\$lf\  \text{lexical function label}

\$lv\  \text{vernacular lexeme referenced by the lexical function}

Related fields are:

\$le\  \text{and}  \$ln

For generic cross-referencing, see:

\$cf\  \text{confer/cross-reference}

<Optional>
**lt**

**Literal meaning**

Used to elucidate the distinct meanings of the parts of an idiom or complex phrase in a lexical entry (\lx) or subentry (\se).

\lx setengah mati
\ps idm
\ge exhausting ; tiring ; frustrating
\de used to describe a situation, activity, or person as being exhausting, tiring, or frustrating
\lt half dead

MDF adds the label "Lit: " to this field and adds single quote marks around the data.

<Optional>

**lv**

**Vernacular lexeme referenced by the lexical function**

This field is used in MDF for encoding the vernacular lexeme in a lexical function network. The \lv field "points to" the vernacular lexeme (morpheme, word, or phrase) that is semantically related to the current headword as mapped or cataloged by the label in the \lf field. An example of its use would be:

\lx feten <-- the headword/lexeme
\ps n
\ge millet
...
\lf Gen <-- the lexical function/relationship
\lv agat <-- the related lexeme
\le grain <-- the English gloss of 'agat'
...

In this example, 'agat' ('grain') is in a Generic relationship to 'feten' ('millet'). Note that the lexeme 'agat' should also appear in the lexicon.

For more information, see:

￥lf lexical function label

Glossing fields for this vernacular lexeme are:

￥le,￥ln, and￥lr

For generic cross-referencing, see:

￥cf cross-reference

<Optional>

**lx**

**Lexeme or headword of the lexical entry**

This is the record marker in Toolbox and is the field by which the database is normally sorted. When formatted, MDF resorts the dictionary based on this field (or the \lc field, if present).

This field contains the lexeme or headword, which is commonly mono-morphemic in a
Toolbox lexical database. But such a lexeme form may not be very accessible for vernacular speakers if printed. To provide a more readable form for vernacular speakers, use the \lc field.

For a discussion of "lexical citation", see: \lc

\ Since this is the record marker, it cannot be added inside any record. It is discussed here simply for completeness. \n
\mn  main entry form
This is used to reference a minimal entry back to its main entry. Minimal or "minor" entries consist of abbreviated information, e.g.:
\lx tado
\ps adj
\ge calm
\mn teduh

This would typically print as:
tado  adj. calm; See main entry: teduh.

In this example, the minor entry "tado" has minimal information and contains a reference back to the main semantically related entry "teduh". This cross-referencing is done through the \mn field:
\mn teduh

The main entry "teduh" would be more verbose, containing all of the information normally included in a lexical entry. The "teduh" entry would also have the field "\va tado" to reference the variant (dialect) form.

For related information, see:
\va variant forms

For a detailed discussion of the issues involved in using minor (or minimal) entries, see:
Using_Subentries_or_Lexical_Entries

<Optional>

\mr  morphemic representation
This can be used to show the underlying morphemic structure for complex lexemes. MDF gives this field the label 'Morph:'

In Buru, the lexeme 'agat' ('grain') is a complex morpheme. Its internal structure should be encoded as:
\mr aga-t

This will typically print as: 
Morph: aga-t.


Optional

na

na notes on anthropology
This field is for any ethnographic note that is pertinent to the lexeme that you wish to keep separate.
If selected for output, MDF adds the label "Anth: " to this field and brackets it with "[ ]".
For generic notes, use:


Optional

nd

nd notes on discourse
This is a place for your notes on discourse/text analysis, should you wish to keep them separate.
If selected for output, MDF adds the label "Disc: " to this field and brackets it with "[ ]".
For generic notes, see:


Optional

ng

ng notes on grammar
This is a place for your grammar notes, should you wish to keep them separate.
If selected for output, MDF adds the label "Gram: " to this field and brackets it with "[ ]".
For generic notes, see:


Optional
np
\np  notes on phonology
This is a place for your phonology notes, should you wish to keep them separate.
If selected for output, MDF adds the label "Phon: " to this field and brackets it with "[ ]".
For generic notes, see:
  ¥nt

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

nq
\nq  questions
This is a place for your questions, should you wish to keep them separate.
If selected for output, MDF adds the label "Ques: " to this field and brackets it with "[ ]".
For generic notes, see:
  ¥nt

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

ns
\ns  notes on sociolinguistics
This is used for encoding sociolinguistic notes. Dialect information (i.e. which villages use this lexeme form, etc.) should be explained in the following fields:
  ¥ue, ¥un, ¥ur (Usage)
  ¥ee, ¥en, ¥er (Encyclopedic)
  ¥oe, ¥on, ¥or (Only -- restrictions)

If selected for output, MDF adds the label "Socio: " to this field and brackets it with "[ ]".
For generic notes, see:
  ¥nt

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

nt
\nt  notes, etc.
This is a generic dump for all your personal notes about an entry, subentry, or sense.
More specific note fields are provided for "splitters", i.e. those who desire a finer
differentiation to their notes. These are:

- \ynp \ notes on phonology
- \yng \ notes on grammar
- \ynd \ notes on discourse
- \yna \ notes on anthropology
- \yns \ notes on sociolinguistics
- \ynq \ questions

If selected for output, MDF adds the label "Notes: " to this field and brackets it with "[ ]".

*This is a "free-form" field. Punctuation and capitalization should be used as needed.*

\<Basic\>

\oe

\oe only [restrictions] (English)

This is for encoding any semantic and/or grammatical restrictions pertinent to the lexeme. In Buru, the lexeme 'anafina' is glossed as 'female'. But this needs to be restricted to only 'human' references, which could be encoded as:

\oe Human

and would print as:

Restrict: Human.

Since this is a free-form field, you can be as verbose as needed, but in many cases, such as this one, a simple code may suffice.

Closely related English fields are:

- \yde, \yue, and \yee

Identical national and regional language fields are:

- \yon and \yor

*This is a "free-form" field. Punctuation and capitalization should be used as needed.*

\<Optional\>

\on

\on only [restrictions] (national)

The national language equivalent to the \oe field. This field is for clarifying semantic and grammatical restrictions pertinent to the lexeme. See the \oe field for more information. MDF initially adds the label "NatRestrict: " to this field. This is later changed to whatever national language label is specified for this field in the national audience CC table. For more information on this, see:
Printed_Field_Labels

For more information, see:
  ¥oe

Closely related national fields are:
  ¥dn, ¥un, and ¥en

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

Or

\or only [restrictions] (regional)
The regional language equivalent to the \oe field. This field is for clarifying semantic and grammatical restrictions pertinent to the lexeme.
If included in the output, MDF adds brackets "[ ]" around this field, and it is treated as part of the national field.
For more information, see:
  ¥oe

Closely related regional fields are:
  ¥dr, ¥ur, and ¥er

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

OV

\ov only [restrictions] (vernacular)
This field contains the vernacular description of any semantic and/or grammatical restrictions pertinent to the lexeme or headword. This is intended for use in a monolingual dictionary, but can be used in diglot and triglot dictionaries as well.
MDF initially adds the label "VerRestrict: " to this field. This is not changed by either the English or national audience CC table. But you may add a rule to change this if you wish.
For more information on how to do this, see:
  Printed_Field_Labels

For more information, see:
  ¥oe
Related vernacular fields are:
\texttt{gv}, \texttt{dv}, \texttt{ev}, \texttt{uv}

<Optional>
\textbf{pc}

\texttt{pc} picture
This contains either the book and page number reference of a relevant picture (you've either sketched in a notebook or found in a picture book), or a graphics link for a PCX file of a picture you want to include in your printed dictionary.
If the field does not begin with \texttt{.G.} as in a graphics link, then the field is treated as "free-form". The field is output near the end of an entry paragraph, and the print tables put parentheses ( ) around the whole field (to set it off from other types of information).
If the \texttt{pc} field begins with a graphics link mark \texttt{(.G.)}, then it is NOT a "free-form" field and should follow the basic form below:
\texttt{pc .G.sago.pcx;1.5";1";PCX}

The \texttt{.G.} marks this as a graphics link; next follows the filename of the picture; next is the width (here 1.5") and then the height (1") and finally the graphics format (PCX). Each bit of information is separated by a semicolon.
When the dictionary is formatted, the graphics information is moved to the beginning of the entry, subentry, or sense in which the \texttt{pc} field is found. This will cause the text to flow around the picture. Sizes much larger than 1.5"x1.5" are not recommended. In double column format the picture is placed flush right in the column; in single column format the picture is flush right to the right margin.

This is a "free-form" field, if it is not a Graphics Link paragraph. Punctuation and capitalization should be used as needed.

<Optional>
\textbf{pd}

\texttt{pd} paradigm set
This is used for specifying the noun or verb class, gender, or other paradigm set that the lexeme or headword is associated with. These classes are generally given labels or numbers to differentiate them. It is assumed that the classes are explained elsewhere (like the introduction to the dictionary) and all that is needed here is the class label or number, e.g.:
\texttt{pd 3}

Mnemonic codes are actually better than numbers (since what the number represents is very difficult to remember), e.g.:
\texttt{pd met:vr}

might refer to a "verb root which metathesizes with its subject prefix". This is much easier to decode than "3". For consistency be sure to use Toolbox's Range Set feature on
this field. For more information, see:
  Range_Sets

This would typically print as:
  Prdm: met:vrt.

To give the actual vernacular form for a paradigm (especially needed where the paradigm
is incomplete or is irregular in form), MDF provides the following fields:
  \$pdl         paradigm label
  \$pdv         paradigm vernacular form
  \$pde         paradigm form-English gloss
  \$pdr         paradigm form-regional gloss
  \$pdr         paradigm form-national gloss

<Optional>

pde
  \pde paradigm form (English gloss)
This is used for glossing the vernacular paradigm form in English, e.g.:
  \pdl Pl
  \pdv asure
  \pde dogs
  \pdr anjing-anjing

This will typically print as:
  Pl: asure 'dogs' 'anjing-anjing'.

where the paradigm label is italic, the vernacular form is in vernacular font, and the two
glosses are in their appropriate language fonts.
For a more detailed explanation of this set of fields, see:
  \$pdl paradigm label
  \$pdv paradigm vernacular form

For the related glossing fields, see:
  \$pdr paradigm form-regional gloss
  \$pdr paradigm form-national gloss

For more information on the paradigm fields, see:
  \$pd paradigm set

<Optional>
The paradigm field (pd) field is used to define the general paradigm set a headword or lexeme is associated with. But it is also useful to give the actual forms for a paradigm, especially where a paradigm set is incomplete or irregular. For example, the third-singular genitive marker normally possesses nouns with the form '-na', except with certain nouns, where its form is irregular. Any noun that takes an irregular 3sGen form should be marked:

\lx ow
\ps n
\ge jackfruit
\gn nangka
\pd n-fd
\pdl 3sGEN
\pdv owan
\pde his/her jackfruit
\pdn nangkanya

These paradigm label fields would typically print as:

3sGEN: owan 'his/her jackfruit' 'nangkanya'.

(where the "3sGEN" is formatted as italic and the vernacular form and the glossing are formatted with the appropriate language fonts)

Note: The paradigm label (pdl) fields can be used with or without the paradigm set (pd) field.

You can use any label you need to mark a paradigm. Use Toolbox's Range Set feature to insure consistent labeling. For more information, see:

Range_Sets

Note: All the fields in this bundle are made up of markers using three-letter codes.

The related fields to this set are:

\pdv paradigm vernacular form
\pde paradigm form-English gloss
\pdn paradigm form-national gloss
\pdr paradigm form-regional gloss

For more information on the paradigm fields, see:

\pd paradigm set

<Optional>
pdn
\pdn paradigm form (national gloss)
This is used for glossing the vernacular paradigm form in the national language. For the related glossing fields, see:
  \pde paradigm form-English gloss
  \pdr paradigm form-regional gloss

For a more detailed explanation of this set of fields, see:
  \pdl paradigm label
  \pdv paradigm vernacular form

For more information on the paradigm fields, see:
  \pd paradigm set

<Optional>
pdr
\pdr paradigm form (regional gloss)
This is used for glossing the vernacular paradigm form in the regional language. For the related glossing fields, see:
  \pde paradigm form-English gloss
  \pdr paradigm form-regional gloss

For a more detailed explanation of this set of fields, see:
  \pdl paradigm label
  \pdv paradigm vernacular form

For more information on the paradigm fields, see:
  \pd paradigm set

<Optional>
pdv
\pdv paradigm vernacular form
This field is used to give the vernacular paradigm form specified by the label in the \pdl field, e.g.:
\pdl 1s
\pdv koban

where "koban" is the first-singular verb form of the verb root "-oban". MDF will format the '1s' label as italic, and the contents of the \pdv field as vernacular text.
For a more detailed explanation, see:

¥pdl paradigm label

For the related glossing fields, see:

¥pde paradigm form-English gloss
¥pdn paradigm form-national gloss
¥pdr paradigm form-regional gloss

For more information on the paradigm fields, see:

¥pd paradigm set

<Optional>

ph

\ph phonetic/phonemic form
This can be used as needed to retain the phonetic information that is lost when an orthographic spelling is used for an entry. The print process can place square brackets around the data. For example:

\ph apa

would typically print as:

[apa]

The field is formatted with the character style "Phonetic form (pronunciation)", so that you can specify a unique font for this field in the document template for MS-Word.

<Optional>

pl

\pl plural form
This is a special field used to give the plural form of the lexeme, e.g. in Selaru, the plural form of the lexical entry 'asw' ('dog') is entered as:

\pl asure

MDF adds a "Pl: " label to this form. The data is formatted as vernacular text.
For more flexible labeling and to allow for glossing, MDF provides the following set of paradigm fields:

¥pdl paradigm label
¥pdv paradigm vernacular form
¥pde paradigm form-English gloss
¥pdn paradigm form-national gloss
¥pdr paradigm form-regional gloss

For related fields, see:
sg     singular form
pd     paradigm set

<Optional>

pn
pn part of speech (national)
This field is used to classify the part of speech with labels found in national language
dictionaries. Consistent labeling is important. Use Toolbox's Range Set feature for this
field.
MDF requires that the \pn field follow the \ps field:
\ps n   (noun)
\pn kb  (the national abbreviate for 'noun')

If the order is reversed, MDF will not function properly.

MDF will format the \pn field only if you specify that the output is for a national
audience. When a national audience is specified, the \pn field will replace the \ps field.
But if there is no \pn field or it is empty, the \ps field will be output for the national
audience as for an English audience.
For more information, see:
   ¥ps part of speech

For important information on Range Sets and on MDF's label substitution feature, see:
   Range_Sets

<Basic>

ps
ps part of speech
This field is used to classify the part of speech for the vernacular form (not the national or
English gloss), i.e., "fat" may be an adjective in English, but that does not mean the
vernacular form can be classified as such. Consistent labeling is important. Use Toolbox's
Range Set feature for this field. For more information, see:
   Range_Sets

To specify a national part of speech label, see:
   ¥pn part of speech (national)

<Basic>
reduplication form(s)
A repository for reduplication forms for later study (conceivably these forms will be moved to subentries, etc. as more is learned about them). For example, the reduplicated form for 'yoban' ('hit') in Selaru is:
\rd ioboban

MDF adds a "Redup: " label to this form. The data is formatted as vernacular text.
For information on subentries, see:
\ys se subentry (a polymorphemic form or a phrase)

For information on paradigm fields, see:
\yp pd paradigm set

For more flexible labeling and to allow for glossing, MDF now provides the following set of paradigm fields:
\yp pdl paradigm label
\yp pdv paradigm vernacular form
\yp pde paradigm form-English gloss
\yp pdn paradigm form-national gloss
\yp pdr paradigm form-regional gloss

<Optional>
re
reversal form (English)
This gives the English word or phrase to be use to reverse the dictionary for an English index. If an \re * is present, the relevant entry, subentry, or sense will be ignored (i.e. not included in the reversed index). Like the \ge field, the data for this field can be kept each in its own field or concatenated in one field, separated by ';'.
It is often the case that there are several translation equivalents for a single vernacular term. 'huma' might mean 'house', 'hut', 'shack', 'dwelling', 'lean-to', etc. Each of these equivalents would be good to have in the reversed index, e.g.:
\re house ; hut ; shack ; dwelling ; lean-to

They could also be entered in separate \re fields. For more information on this, see:
\yg ge gloss (English)

The advantage of keeping them in separate fields is that if you setup Toolbox to sort on the \re fields (to get a type of finderlist), it will sort on all of them, but if the reversal glosses are all concatenated, Toolbox will only index on the first form.
This field does not normally print in the dictionary.
rf
\rf reference to notebooks, texts, etc.
This field keeps the notebook reference for the following example sentence. This will enable you to validate the example at a later date (e.g., if it comes from an early notebook, it may be suspect).
The rf, xv, xn, and xe fields are considered "bundled" (or grouped) together. If you include multiple examples for a single entry, subentry, or sense, be sure to include the fields grouped together.
\rf nbk1.23.12
\xv example1
\xn ...
\xe ...
\rf nbk1.54.09
\xv example2
\xn ...
\xe ... etc.

Usually these are grouped with the rf field at the beginning of each bundle, but if you don't want to use the rf field, then the xv field will mark the beginning of example sentence bundles.
You may use as many different example sentence bundles as you need for each sense, part of speech, and/or subentry in a record. Within a given section (e.g. sense), multiple examples are printed one after the other.
MDF adds the label "Ref: " to this field.
See also:
\xv example sentence (vernacular)

This is a "free-form" field. Punctuation and capitalization should be used as needed.

rn
\rn reversal form (national)
This gives the national language word or phrase form to be use to reverse the dictionary for a national language index. If an \rn * is present the relevant entry, subentry, or sense will be ignored (i.e. not included in the reversed index). Like the gn field, the data for this field can be kept each in its own field or concatenated in one field, separated by ';'.
For more information, see:
\re reverse English
Also see:
\gn gloss-national
This field does not normally print in the dictionary.

<Basic>

**rr**

\rr reversal form (regional)
This gives a regional language form which could be used to reverse the dictionary (to make a regional language list), or to be included in a national language list marked as the regional language (to aid recall).
The ability to reverse on a regional form is not currently supported in MDF. This field does not normally print in the dictionary.

<Optional>

**SC**

\sc scientific name
Providing a scientific name for a lexeme can be very useful if accurate. The gloss "a type of tree" is nearly worthless (but often the best we can do at the time). Having the scientific name allows us to eventually find the appropriate English gloss for such a species. Getting the scientific name requires access to high quality books (color pictures help).
The data given is automatically underlined and italicized by MDF; no formatting is needed.

<Optional>

**sd**

\sd semantic domain
This is the English version of \th and probably the one to use first. Here you try to catalog and differentiate the semantic compartments of an entry, being careful to not let the English force or mask the vernacular relations. Moving to the vernacular terms (given in \th field) as early as possible is best.
If selected for output, MDF adds the label "SD: " to this field.
For related fields see:

- \¥th thesaurus
- \¥is index of semantic

This field does not normally print.

<Optional>

**Se**

\se subentry (a polymorphemic form or a phrase)
This is like the \lx field except it occurs within the record, marking the word (or phrase) as a form derived from the root. Following this marker would be all the fields that
comprise a typical lexical entry. There can be several of these subentries within a record. Subentries can also have multiple senses. A simple example using subentries would be:

```latex
\texttt{\textbackslash lx bren}
\texttt{\textbackslash ps vi}
\texttt{\textbackslash ge play}
\texttt{\textbackslash ee Implies lack of focus or purpose.}

\texttt{\textbackslash se brenak}
\texttt{\textbackslash ps vt}
\texttt{\textbackslash ge play\_s.t.}
\texttt{\textbackslash de play a game, or play with s.t.}

\texttt{\textbackslash se inabren}
\texttt{\textbackslash ps n}
\texttt{\textbackslash ge recreation ; entertainment}

\texttt{\textbackslash se rabrenak}
\texttt{\textbackslash ps n}
\texttt{\textbackslash ge toy}
\texttt{\textbackslash dt 17/Jun/92}
```

This would typically print like the following:

```
bren vi. play. Implies lack of focus or purpose.
brenak vt. play a game, or play with s.t.
inabren n. recreation, entertainment.
rabrenak n. toy.
```

But note that the subentries in this example are far too simplistic; they lack much of the information that should be provided for these polymorphemic lexemes (definitions, example sentences, cross-references, notes, etc.). For a more detailed discussion of the issues involved in using subentries, see:

Using_Subentries_or_Lexical_Entries

For information on the topic of sections in a lexical entry, see:

Sections_in_a_Lexical_Entry

<Reserved>

**sg**

`\sg  singular form`

This is a special field used to give the singular form of the lexeme, e.g. in Selaru, the singular form of the lexical entry 'asw' ('dog') is entered as:

`\sg askwe`

MDF adds a "Sg: " label to this form. The data is formatted as vernacular text. For more flexible labeling and to allow for glossing, MDF provides the following set of paradigm fields:

```
\Y pdl paradigm label
\Y pdv paradigm vernacular form
```
pde              paradigm form-English gloss
pdn              paradigm form-national gloss
pdr               paradigm form-regional gloss

For related fields, see:
\pl               plural form
\pd               paradigm set

<Optional>

\sn
\sn  sense number
Where an entry has more than one sense, this code gives the number and marks the beginning of each sense, e.g.:
\sn 3             (no punctuation)

Generally each \sn section contains a full basic set of field markers (especially example sentences as these help to validate the distinctions between the senses). Depending on the hierarchy you choose, the sense number \sn is considered either below the \ps field (in the standard hierarchy) or is superior to all fields but the \lx, \lc, and \hm fields (in the alternate hierarchy).
For more information on hierarchy, see:
   Sections_in_a_Lexical_Entry
   Alternate_Hierarchy

For information relating to examples, see:
\rf

Do not forget to include \sn 1 in records that have more than one sense.

<Reserved>

\so
\so  source of data
This is a place to indicate the name and village of the informant who gave you the data in the current entry. There is no standard way such information should be encoded.
If selected for output, MDF adds the label "Source: " to this field and brackets it with "[ ]".
This field does not normally print.

<Optional>
\textbf{st}

\texttt{\textbackslash st \ status}

This is used to indicate how complete or thoroughly checked an entry is, e.g.:

\texttt{\textbackslash st \ OK}

\texttt{\textbackslash st \ no-print}

\texttt{\textbackslash st \ check}

Later you could filter the database to select only "check" records, export this filtered database through MDF, and print the formatted output from Word. This makes it easy for a language assistant to systematically check these entries.

If selected for output, the label "Status: " is usually added to this field and the whole field is bracketed with "[ ]".

\textit{This field does not normally print.}

\textbf{sy}

\texttt{\textbackslash sy \ synonym}

This and the \texttt{\textbackslash an} (antonym) field are helpful for those who want to keep track of such information without using the \texttt{\textbackslash lf} structure.

But you are encouraged to use the \texttt{\textbackslash lf} fields to handle "synonym" and "antonym" cross-referencing (instead of the \texttt{\textbackslash sy} and \texttt{\textbackslash an} fields). This is because the \texttt{\textbackslash lf} fields allow for glossing of the reference, whereas \texttt{\textbackslash sy} and \texttt{\textbackslash an} do not. Glossing has the advantage of giving the outside reader an idea of the meaning of a referenced lexeme without actually having to go and look it up directly. For example, a synonym of '-haw' ('to pound with a pestle') is '-tutu' ('to pound with a rock'). This could be encoded in the '-haw' entry as:

\texttt{\textbackslash sy -tutu}

And would print as:

\texttt{Syn: -tutu.}

But this tells the reader nothing really. Whereas if this were encoded as:

\texttt{\textbackslash lf \ Syn}
\texttt{\textbackslash lv -tutu}
\texttt{\textbackslash le \ to \ pound \ with \ a \ rock}

This would print as:

\texttt{Syn: -tutu \ 'to \ pound \ with \ a \ rock'.}

Which, to the outsider using your dictionary, is far more helpful.

For related information, see:

\texttt{\textbackslash lf \ and \ \textbackslash cf}

\textbf{See also:}

\texttt{\textbackslash an}
This marks the following text as unformatted. Line-breaks and hard-coded tab characters will be retained. (Multiple spaces are also be retained, but this is only useful if you define the "Table" font in MS-Word to be a fixed width font.)

A table, or list, of "cutting verbs" might look like this:

\begin{verbatim}
<table>
<thead>
<tr>
<th>Verb</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>akrina</td>
<td>split in two lengthwise</td>
</tr>
<tr>
<td>boras</td>
<td>cut s.t. in small pieces with a knife</td>
</tr>
<tr>
<td>dew</td>
<td>chop s.t. into smaller pieces while standing it on end</td>
</tr>
<tr>
<td>het</td>
<td>chop or hack with a machete</td>
</tr>
<tr>
<td>kety</td>
<td>slice open and clean an animal</td>
</tr>
<tr>
<td>lary</td>
<td>slice (like chilies, etc.)</td>
</tr>
<tr>
<td>lilit</td>
<td>shave or carve</td>
</tr>
<tr>
<td>mair</td>
<td>to adz wood</td>
</tr>
<tr>
<td>simat</td>
<td>pop out or cut out coconut meat</td>
</tr>
</tbody>
</table>
\end{verbatim}

In Toolbox, the table field is marked as a "No Word Wrap" field (in the Marker-Properties dialog box). This allows you direct control over line-breaks and spacing. Note the bar-code in the above example. This tells Toolbox to insert a Tab at each of these places, when the file is formatted by MDF or exported to an RTF file. (Currently Toolbox does not support typing the Tab character directly.)

Inevitably, your tables will require some "tweaking" in MS-Word before you print the dictionary. To do this, first convert the file to MS-Word format using MDF, and then search for the "Table" style. After some tweaking, these could then be converted into a Word table (with the Word menu command: Table-Convert Text to Table).

For a more powerful and in-depth approach to mapping the relations of lexical items, see:

\begin{verbatim}
¥lf
\end{verbatim}

For more on special codes supported by Toolbox, see:

\begin{verbatim}
Punctuation_and_Special_Codes
\end{verbatim}

This is a "free-form" field. Punctuation and capitalization should be used as needed.

This is a field for developing a vernacular-based thesaurus. It is to be labeled with the vernacular term governing the semantic domain of the entry. Reversing on this field (within Toolbox) would yield a vernacular thesaurus.

If selected for output, MDF adds the label "Thes: " to this field.

For related fields see:

\begin{verbatim}
¥sd
¥is
\end{verbatim}
This field does not normally print.

<Optional>

ue

\ue usage (English)
This field should cover such information as common usage, or restrictions in usage, (such as taboos), or any other information that is needed so a non-native speaker can use this lexeme properly. For example, the Selaru lexical entry for "wai" contains the field:
\ue A kin term of address which is used for same-sex siblings or for marriable non-kin; but may not be said to siblings of the opposite sex.

MDF adds the label "Usage: " to this field.
Identical national and regional language fields are:
 ¥un and ¥ur

Closely related fields are:
 ¥de, ¥ee, and ¥oe

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

un

\un usage (national)
The national language equivalent to the \ue field. This field should cover information such as common usage, or restrictions in usage. For more information, see:
 ¥ue usage (English)

MDF initially adds the label "NatUsage: " to this field. This is later changed to whatever national language label is specified for this field in the national audience CC table. For more information on this, see:
 Printed_Field_Labels

Identical English and regional language fields are:
 ¥ue and ¥ur

Closely related fields are:
 ¥dn, ¥en, and ¥on
This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

**Ur**

<ur usage (regional)>
The regional language equivalent to the \ue field. This field should cover information such as common usage, or restrictions in usage. For more information, see:

\ue usage (English)

MDF adds the brackets "[ ]" around this field. If included in the output, it is treated as part of the national field.

Identical English and national language fields are:

\ue and \un

Closely related fields are:

\edr, \er, and \eor

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

**Uv**

<uv usage (vernacular)>
This field contains the vernacular description of common usage, or restrictions in usage, (such as taboos), or any other information that is needed to describe the lexeme fully.

This is intended for use in a monolingual dictionary, but can be used in diglot and triglot dictionaries as well. For more information, see:

\ue usage (English)

MDF initially adds the label "VerUsage: " to this field. This is not changed by either the English or national audience CC table. But you may add a rule to change this if you wish.

For more information on how to do this, see:

Printed_Field_Labels

Identical English, national and regional language fields are:

\ue, \un and \ur

Closely related fields are:

\ev, \ov, \dv, and \gv
va

\va variant forms
This is where variant forms of the lexical entry or subentry can be noted (be they from another dialect or minor alternation in the focus dialect, as in "do not" and "don't"). These variant forms can (but do not have to) refer to minor or minimal entries found elsewhere in the dictionary.

The \va field heads a bundle of comment fields (\ve, \vn, \wr). These comment fields can be used to specify the dialect name or area that uses the variant form given in the \va field. But because they are comment fields, you may enter any comment information that you want to appear in the dictionary with the variant form.

MDF adds the label "Variant:" to the beginning of the first \va field in any given section of a lexical entry. The comment fields are added to the variant form with parentheses, but no additional label. Multiple variant bundles are allowed.

The related comment fields are:

\¥ve, \¥vn, and \¥vr

For information concerning minor entries, see:

\¥mn main entry form

For a detailed discussion of the issues involved in using minor (or minimal) entries, see:

Using_Subentries_or_Lexical_Entries

<Optional>

\ve

\ve variant comment (English)
This is bundled with the \va field and is where English comments can be given for the variant form. For more information, see:

\¥va variant forms

MDF adds parentheses "( )" around this field.

The related comment fields are:

\¥vn and \¥vr

This is a "free-form" field. Punctuation and capitalization should be used as needed.
\vn
variant comment (national)
This is bundled with the \va field and is where national language comments can be given for the variant form. For more information, see:
\va variant forms

MDF adds parentheses "( )" around this field.
The related comment fields are:
\ve and \vr

This is a "free-form" field. Punctuation and capitalization should be used as needed.

\vr
variant comment (regional)
This is bundled with the \va field and is where regional language comments can be given for the variant form. For more information, see:
\va variant forms

If this field is included in the output, MDF adds parentheses "( )" around it and it is treated as part of the national field.
The related comment fields are:
\ve and \vn

This is a "free-form" field. Punctuation and capitalization should be used as needed.

\we
word-level gloss (English)
This gives the gloss to be used in word-level interlinear glossing.
Related fields are:
\wn and \wr

This field does not normally print.

\wn
word-level gloss (national)
This gives the gloss to be used in word-level interlinear glossing.
Related fields are:

\$\text{we}$ and \$\text{wr}$

\textit{This field does not normally print.}

\texttt{<Optional>}

\texttt{wr}

\texttt{\textbackslash wr word-level gloss (regional)}

This gives the gloss to be used in word-level interlinear glossing.

Related fields are:

\$\text{we}$ and \$\text{wn}$

\textit{This field does not normally print.}

\texttt{<Optional>}

\texttt{xe}

\texttt{\textbackslash xe translation of example (English)}

This provides the English translation of the example sentence given in the \texttt{xv} field.

For more information about example sentences, see:

\$\text{xv}$ example sentence (vernacular)

\$\text{rf}$ reference to notebooks, texts, etc.

Related fields are:

\$\text{xn}$ and \$\text{xr}$

\textit{This is a “free-form” field. Punctuation and capitalization should be used as needed.}

\texttt{<Basic>}

\texttt{xg}

\texttt{\textbackslash xg (discontinued field)}

This field has been discontinued. Originally, MDF reserved the marker \texttt{xg} for interlinear glossing, but its function was never fully developed. For this reason, it has been removed from the reserved list of field markers that MDF will recognized.

\textit{If this field is included in a lexical entry, MDF will simply treat it the same as an unknown field.}

For information on unknown fields, see:

\texttt{Unknown_Fields}
xn

xn translation of example (national)
This provides the national translation of the example sentence given in the \xv field.
For more information about example sentences, see:
\xv example sentence (vernacular)
\rf reference to notebooks, texts, etc.

Related fields are:
\xe and \xr

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Basic>

xr

xr translation of example (regional)
This provides the regional translation of the example sentence given in the \xv field.
MDF adds the brackets "[ ]" around this field. If included in the output, it is treated as part of the national field.
For more information about example sentences, see:
\xv example sentence (vernacular)
\rf reference to notebooks, texts, etc.

Related fields are:
\xe and \xn

This is a "free-form" field. Punctuation and capitalization should be used as needed.

<Optional>

xv

xv example sentence (vernacular)
A good rule of thumb is to keep your example sentences relatively short (one line or so).
Good examples should indicate the usage of the lexeme without being stilted.
Be aware that sentences taken straight from texts generally do not make good example sentences, because this removes them from their context. Such sentences are often encoded with particles operating on the larger discourse and are likely to be bound anaphorically with preceding participants.
Nouns rarely need example sentences, but good sentences are crucial to differentiating the various senses of verbs and can demonstrate and verify peculiar or rare domains or usages.
For more information about example sentences, see:
Related fields, used to translate or gloss the vernacular example sentence, are:

\r\n\r\nThis is a "free-form" field. Punctuation and capitalization should be used as needed.

<Basic>

1s 1p 1e 1i 1d 2s 2p 2d 3s 3p 3d 4s 4p 4d

Old verb paradigm markers

These fields were for various possible verb paradigm forms. For example

\r\n\r\nThese markers are no longer recommended. Use the Paradigm markers instead.

Other General Information

Uncategorized General Information

Marker Pane Width (from a txt file)

The Marker Pane has an assumed width, depending on whether the names or hierarchy are shown. However, you can readily change the width using the mouse.

Position the mouse cursor over the vertical line between the markers and the data until it becomes a double-headed arrow. Hold down the left mouse button and drag the line to the position you desire.

Changing how the markers are viewed (i.e., showing names or hierarchy) will reset the width to a default value.

Auto Save (Project menu)

Auto Save command (Project menu)

The Auto Save option causes Toolbox to save the database files and settings files automatically whenever you move to another entry in the same window and whenever you give the Interlinear command.

To enable Auto Save, click the item on the Project menu and a check symbol will appear next to it. To disable Auto Save when it is already enabled, click the item on the Project menu and the check mark will disappear.

Auto Save defaults to on.

Close (Project menu)

This command allows you to close the active project without exiting the Toolbox program. If you have made any changes to the loaded database(s), Toolbox will display a dialog box asking if you want to save the changes made to the specific database(s). Should you choose to close the project without saving, you will lose all changes not previously saved.

Choosing this command accesses the No project open dialog box.

Closing a project without saving it

If you have not saved any changes, then when you choose Project Close, it will ask if you want to close the project without saving. If you choose Yes, it will not overwrite any of the settings or data files.
This can be especially useful in cases where a project has been moved to a different computer and cannot find some of its data files. Closing the project without saving lets you try to put the files where the project can find them and then try to open the project again.

This feature may also be used to perform experiments on a data set and then close the project without saving so that the data is not modified. But to do that, you must turn off Auto Save.

**Close a Toolbox database**

**Close command (File menu)**

You can close the active database without exiting Toolbox. However, if you have made any changes to the database, Toolbox will display a dialog box asking if you want to save them unless you have Auto Save enabled. If Auto Save is enabled, files are automatically saved.

To close a database

Do one of the following:

- Double-click on the database Control menu box in the upper-left corner of the database window.
- Click once on the Close box in the upper-right corner of the database window.
- Click once on the database Control menu box, choose Close from the resulting menu.
- Use the shortcut keys, CTRL+F4
- Click on the File menu, choose Close. **Note:** Using the Close command (File menu) closes all duplicate windows of the active database. While the other options above close only the active database, leaving all duplicate database windows open.

If a database has changes you have not saved, Toolbox displays a dialog box asking if you want to save the changes before closing. **If you close a database without saving, you lose all changes not previously saved.**

**Close Last File dialog box**

This dialog box appears when you are closing the last database (file) open in the Toolbox program. The purpose of this message is to help prevent confusion for you, the user.

It is important to remember that the project file retains vital information about the configuration of the Toolbox workspace between each session — the size and position of the application window, the files that were loaded, how many database windows you had open, where they were placed on the screen, and how each particular window was sorted and filtered. These project settings are saved every time you exit the Toolbox program. If you want the project to open every time with certain files loaded, then you should **never** close those files and exit the program.

Most of the time, you will not really want to close all of the databases which are open in your project. However, if you are wanting to change the database files which are displayed in a particular project, you will want to close the current files.

**Note:** The Toolbox program is designed to **always** run with a project file. When no files are opened in a project, the Toolbox application window appears empty. This might be confusing to the user if he/she does not realize that there is **always** a project file loaded. The name of the project appears in the right corner of the status bar.

**Exiting or Quitting Toolbox**

**Exit command (File menu)**

When you exit Toolbox, all databases are saved. If you have made any changes to one or more of the databases, Toolbox asks if you want to save the database(s) before quitting unless you have Auto Save enabled. If Auto Save is enabled, files are automatically saved.
To exit or quit Toolbox
Do one of the following:
  • From the File menu, choose Exit.
  • Double-click on the application Control menu box or click once on the Close box in the upper-right corner of the application window.
  • Use the shortcut keys, ALT+F4.

**Note:** When exiting, do not close the files you have open. The files you leave open will be opened again the next time you run Toolbox, in the same configuration and at the same records as when you exit. This allows you to continue working without having to re-establish your workspace.

**Import dialog box**
**File Open dialog box (Open command, File menu)**

When the file you are opening is not in Toolbox format, this dialog box will automatically appear offering the following options:

**Database Type**
This combo box allows you to specify the database type. All databases must be assigned to a type. If you do not find a suitable type in the list, one may be added.

**Add a New Database Type**
This button accesses the Database Types dialog box.

**Make Consistent Changes**
The Use Conversion Table check box allows you to choose whether or not a conversion table (*.cct) will be used during the import process. The Browse button accesses the Choose the Conversion Table dialog box.

**Remove hanging indent spaces**
This check box allows you to choose whether or not you will eliminate leading spaces in the data. Some users may have used a hanging indent to separate markers from the data. In this case, enabling this check box will strip those spaces. However, if your data has charts or tables which use spaces as delimiters, then you will want to disable this feature.

**Backup original as *.ori**
This check box allows you to choose whether or not you will make a backup of your original file. If turned on, the backup file will have an *.ori extension. This option is recommended.

The text window in this dialog box shows a portion of the actual contents of the file you are importing.

**Note:** Any tab character will be converted to a space when imported into Toolbox.

**Network Safety**
Multiple users of Toolbox can safely share files on a network.

When Toolbox opens a file, it write protects the file. If another instance of Toolbox opens the same file, the second instance opens the file as read only, and does not make changes to the file. This means that if multiple users on a network open the same file, only the first user to open the file will be able to modify the file. (In Shoebox and in Toolbox 1.2 and earlier, if two users on a network opened the same file, both could make changes, but changes made by one of the users would be lost.) This feature also allows multiple Toolbox projects to be run on the same computer.
Toolbox write protects its project file to avoid the possibility of two instances of Toolbox opening the same project file. If when Toolbox is run, the project file is write protected and Toolbox is already running on the same machine, it refuses to open the project a second time.

However, if because of a power failure or other problem Toolbox has been terminated in an abnormal way, then the project file may remain write protected even though no instance of Toolbox has it open. If that is the case, then the first time you try to open the write protected project, nothing will happen. But Toolbox will have cleared write protect on the project file, so the second time you try to open it, it will open normally.

**New (File menu)**

This command is used to create a new database. Toolbox expects you to specify a name and assign an existing database type as the new database is created.

To create a new database:
1. From the File menu, choose New.
2. In the Enter the Name for the New Database dialog box, type the name of the database you want to create.
3. Choose the OK button.
4. At this point, Toolbox will ask you to assign a type for the new database. The Select Database Type dialog box opens with the following options:

   **Please select the database type for**
   A reminder of the name of the new database.

   **Database Type**
   Selects a database type.

   **Add New Type button**
   Accesses the Database Types dialog box. Click the Add button to add a new database type or click the Copy button to copy the highlighted database type to a new name.

**Note:** To open an existing database, use the Open command (File menu).

**New (Project menu)**

The best way to start a new project is to use the Toolbox new project kit. To do this, download the Toolbox New Project Package from the Toolbox website and unzip it. That will make a Toolbox Project folder with empty files ready for you to enter your data. It will create a shortcut on your desktop that you can use to open your project.

For more information, see the Toolbox Training package.

If you choose Project, New on the Toolbox menu, it will ask if you want to install the required files. If you say yes, it will tell you to install the New Project package.

If you say you do not want to install the required, then it will allow you to make a new empty .prj file. But this is rarely done because an empty .prj file is not very useful. The best way to make a new .prj file is to use Project, Save As. This makes a new .prj file that starts with the same information as your current one.

**Note:** If you are currently running Toolbox with an active project file, choosing Project, New will cause the active project file to be replaced with the new project file and any open databases and settings files will be closed. (Before closing a database in which changes were made, Toolbox will ask if you want to save first.) You will then need to load or create the appropriate files for the new project.

To add databases to a project, use the Open or New command of the File menu.

For more information about projects, see Projects, An Overview and The Toolbox Project.
No project open dialog box
Close command, Project menu

The Toolbox program is designed to run with a project file. Should you chose to close the active project, this dialog box will open with the following options:

Open an existing project
Accesses the Choose the Project file to open dialog box allowing you to specify or select the file name and location. If you do not see the project you want to open, select a new drive or folder (directory). This option is the same as choosing the Open command in the Project menu.

Create a new project
Accesses the Enter the Name for the New Project dialog box allowing you to specify the name and location of the project file. This option is the same as choosing the New command in the Project menu.

Exit the Toolbox program
Ends a Toolbox session. This option is the same as choosing the Exit command in the File menu.

Note: If you have made any changes to the database(s) in the open project, Toolbox will display a dialog box asking if you want to save the changes made to the specific database(s). Should you choose to close the project without saving, you will lose all changes not previously saved.

Open (File menu)
Open command (File menu)

This command opens a Toolbox database that has been saved on a disk.

To open an existing database
1. Do one of the following:
   • On the toolbar, click the Open button.
   • Use the shortcut keys, CTRL+O.
   • From the File menu, choose Open.
2. The Open dialog box appears allowing you to specify or select the name of the database you want to open.
   If you do not see the database you want to open, select a new drive or folder. If you type an existing file type in place of the wild card character in the File Name box, the file list will change to include only the files of that type.
3. Choose the OK button.

Note: If the file you are opening is not in Toolbox format, the Import dialog box will automatically appear.

When opening a file, markers in that file which do not already appear in the list of markers are automatically added and flagged with an asterisk in the field name.

Open (Project menu)
This command opens a Toolbox project file that has been saved on a disk.

To open an existing project
Do the following:
1. From the Project menu, choose Open
2. The Choose the Project file to open dialog box appears allowing you to specify or select the file name and location. If you do not see the project you want to open, select a new drive or folder (directory).
3. Choose the OK button.

Opening a file which has been moved
Dialog Box

This dialog box allows you to open Toolbox database file(s) associated with a project that have been moved to another folder (directory) or perhaps to another computer.

When opening a project, if Toolbox discovers the project file is in a different folder than it was the last time it was opened, Toolbox will attempt to open the data files. If the data files were in a folder either above or below the settings files, Toolbox will assume the are in the same relationship to the current location and attempt to locate them. If it is successful, the project will open without this dialog box being displayed. However, if the data files were in a folder not along the same path as the settings files, this dialog box is automatically opened.

Also, if you have moved the data file(s), even if the rest of the project was not moved, when Toolbox discovers the file is not where it was expected, this dialog box will be opened.

To open a file which has been moved, select the new drive or folder (directory) and then choose or type the name of the database.

If you choose Cancel, Toolbox will skip the file and no longer include it as a part of the project.

Note: If you type the extension of the file you want to open, along with the wildcard character, in the File Name box, the file list will change to include only files with that extension. For example, *.db or *.txt. This can make it easier to spot the file you are looking for.

Portability for Removeable Devices
Toolbox can automatically run a default project on removeable media. This, together with the ability to define an internal keyboard in Toolbox, allows for complete portability. This is intended to be a convenience for working in Internet cafes and for borrowing a friend's computer.

Criteria:
To be fully portable, these criteria must be met.
  • The Toolbox program file Toolbox.exe must be on the removeable drive.
  • The Toolbox program must be in the same folder as the settings file of the project to be run.
  • The Toolbox program must be run by clicking on the Toolbox program icon on the removeable drive. The project must be named Toolbox.prj or Toolbox Project.prj.

If these criteria are met, then when Toolbox is run, it will recognize the Toolbox.prj or Toolbox Project.prj in the same directory with itself and will open that project.

Normally when Toolbox runs, it leaves an indication of the project name in the registry of the host computer so the project can be run again by double-clicking the Toolbox program icon. However, if Toolbox runs a default project, it will not modify the registry on the host computer, nor will it leave any indication of having been run on the host computer.
Note that there can be multiple such projects on the same removable drive. But each project must have its own folder and each must have its own copy of Toolbox.exe in the settings folder.

Creating a Portable Project on Removable Media
To create a portable project, follow these steps:

• Create a folder for the project on the removable media.
• Copy the settings and data from the hard drive onto the removable media.
• On the removable drive, rename the .prj file to Toolbox.prj or Toolbox Project.prj.
• Navigate to c:\Program Files\Toolbox (or wherever you placed the program itself.) Copy the following files to the settings folder on the removable drive.

  • Toolbox.exe
  • Toolbox.hlp

(Do not copy the folder itself; only copy these files.)

Save (File menu)
Save command (File menu)
This command is used to save an existing database.
To save a database do one of the following:

• On the toolbar, click the Save All button.

  • Use the shortcut keys, CTRL+S.
  • From the File menu, choose Save.

Note: The Save command forces a save even if there were no changes made to the database. This causes the date and time on the file to change, even if no changes were made to the file. In contrast, the Save All command (or button) does not change the date and time on a database which has not changed.

The Auto Save option in the Project menu will cause Toolbox to automatically save whenever you move to another record or request interlinearization. For normal working, Auto Save is a very useful feature. Consultants and others doing somewhat experimental things with data may want to turn Auto Save off so they can exit without saving if appropriate.

Save (Project menu)
This command saves all files associated with the Toolbox project including the database(s) and settings files (*.typ, *.lng and *.prj). There is no dialog box associated with this command.

While this feature has the potential to save all files loaded into the current project, it does not force a save, but saves only when changes have been made during the current session or since you last saved.

Note: It is recommended that you save your project files periodically when making extensive setup changes.

Save All (File menu)
This command saves all databases currently loaded into Toolbox along with their settings files (*.typ, *.lng and *.prj). There is no dialog box associated with this command.

Note: While this feature has the potential to save all files loaded into the current project, it does not force a save, but saves only when changes are made to the database.
**Save As (Project menu)**
Saves the active project file to a new name.

To save project file to a new name do the following:

1. In the Project menu, choose Save As.
2. The Save This Project As dialog box opens allowing you to specify the file name and location. If no extension is entered, Toolbox will use *.prj.
3. Choose the OK button.

**Saving a database to a new name**
**Save As (File menu)**
Saves the active database under a new name or in a different location in Toolbox format.

To save a database to a new name do the following:

1. In the File menu, choose Save As
2. The Save As dialog box opens with the following options:
   - **Save in**
     Allows navigation to a different folder and/or drive. Click the drop arrow to see the directory tree.
   - **File name**
     Specifies the new name of the database.
   - **Save as type**
     Shows file type. A file may be saved with any extension—there are no limitations.
3. Choose the OK button.

This command saves the database to a new name and opens the new file in the same way that Word and other programs do. The database with the "old" name is closed but intact.

This feature saves the active database in Toolbox format. If you want to save a database to a different format, use the Export command in the File menu.

To quickly save a database with its existing name and location, click the Save button on the toolbar.

**Select Database Type dialog box**
**Enter the Name for the New Database dialog box, New command, File menu**

This dialog box opens when you are adding a new database (file) to the Toolbox project. It contains the following options:

**Please select the database type for**
A reminder of the name of the new database.
Database Type
Selects a database type.

Add New Type button
Accesses the Database Types dialog box. If none of the existing database types are appropriate for the new file, you can click the Add button to add a new database type or click the Copy button to copy the highlighted database type to a new name.

Note: If a database type that you expect to see is missing from the drop-down list in this dialog box, it may not be in your project folder (directory).

Status Bar: Displaying or Hiding
Status bar (View menu)
This command allows you to display or hide the status bar that is shown at the bottom of the application window. The status bar is displayed when a check mark appears beside its name. To hide the status bar, choose the command again.

Status bar

For Help, press F1  \%x adar  \%ps n  10/66  mdf.prj

The status bar is a horizontal bar at the bottom of the Toolbox window which displays the following information:

- A Toolbox program message, in this case, a reminder to press F1 for Help
- The record marker field and the contents or a portion of the contents, \%x adar
- The primary sort field and the contents or a portion of the contents, \%ps n
- The total occurrences of the primary sort field (66), along with the current position in the database (10), appear as 10/66. If the database is sorted on the record marker field, the total will also be the number of actual records in your database.
- The filename of the project currently loaded into Toolbox, mdf.prj.

To display or hide the status bar, choose Status bar from the View menu. To switch between a large status bar (to accommodate larger fonts) and the normal status bar, choose Large controls from the View menu.

Toolbar: Displaying or Hiding
Toolbar (View menu)
This command allows you to display or hide the toolbar as a part of the application window. The toolbar is displayed when a check mark appears beside its name. To hide the toolbar, choose the command again.

Toolbar
A bar with buttons that perform the following most common tasks in Toolbox:

- Opening a database
- Saving a database
To display or hide the toolbar, choose Toolbar from the View menu. To use the toolbar, click on the button which represents the action you wish to take.

**Toolbox application window**
The Toolbox main window has a menu and a toolbar at the top and a status bar at the bottom. Inside the main window are smaller windows that show the open databases.

**Data Related Topics**

**Jump, Find and Search**

*Locating records containing a particular field.*

There are several ways to locate the record(s) containing a particular field or particular data. Each way uses a different Toolbox feature:

**Find**
1. From the Edit menu, choose Find.
2. Select the particular marker without specifying any text.
3. Choose OK. (Toolbox automatically inserts the skip variable [...] as the text to be found.)
4. Use Find Next or Find Previous to move to other records within the database.

**Sort/Browse**
1. In the Database menu, choose Sorting.
2. Change the Sort Field to the marker you wish to locate.
3. From the View menu, choose Browse Fields.
4. Add the marker you wish to locate to the Browse Fields.
5. Do one of the following:
• On the toolbar, click the Browse button.
• Use the shortcut keys, ALT+R.
• From the View menu, choose Browse.

The Browse view will display all occurrences of this marker. Those records which do not contain the field will be labeled *no field*. Those which contain the field but in which it is empty will be next, labeled *empty*. (Note: The actual text for these special labels can be changed in the Browse Fields dialog box.)

**Filtering**

1. From the Database menu, choose Properties.
2. Choose the Filters Tab
3. Select Add.
4. From the Filter Properties dialog box, choose Marker Text.
5. In the Marker Text Filter Element dialog box, type or select the marker without specifying any text.
6. Choose OK.
7. Then, in the Filter Properties dialog box, type a name for the filter.
8. Choose OK. Toolbox displays the main window.
9. Activate the filter through the Active Filter combo box by clicking the drop-down arrow button, accessing the filter list and selecting the name of the filter you just created.

Only those records containing the marker will be displayed.

To restore to all records, again drop the list of available filters and choose [no filter].

These features are explained in detail in this document.

**Locating Data: Search, Find, Jump To**

There are important similarities and differences between the Search, Find, and Jump To commands. Jump Insert, Jump Path, and Return From Jump are also documented here.

**Search**

• Matches text string at the beginning of the primary sort field
• Restricted to the active database
• Gives option to match whole field
• Looks only through the (already exiting) index created with the Sorting command for possible matches to the text string

**Find**

• Matches text string anywhere within specified field(s) or language encoding
• Restricted to the active (filtered) database or current record (user option)
• Gives option to match whole word (You can choose to find an embedded string or not.)
• Looks through the entire record or database, more or less letter by letter, for the occurrence of the text string

**Jump To**

• Matches text string at the beginning of any field listed in the jump path; if there is no jump path, it searches the sorting index of all open databases for a match.
• Not restricted to the active database, or even the same database type
• Gives the option to match whole field
• Builds an index for each database-field combination in the jump path (if a jump path is specified).

**Note:** The Filtering command builds an index of records matching the filter criterion, thereby restricting the amount of data which Toolbox looks through when using the Search, Find, and Jump To commands.
**Match Characters**

Search, Find, and Jump all have an option called **Match Characters** which restrict or loosens their search. To avoid excessive repetition, it is documented here.

Match characters controls how Toolbox determines the match. For options below, consider an example with the following primary groupings in the sort order:

- A a à à
- B b
- D d

**By primary grouping only**

Allows any characters grouped together in the current sort order to be considered identical. Using the example above, this option would allow the search text `bad` to match `bad, bäd, bàd` and `Bad`.

**Disregarding case**

Matches the exact characters in the search text, but does not make an upper/lower case distinction. Using the example above, this option would allow the search text `bad` to match `bad` and `Bad`, but not `bád` nor `bàd`.

**Exactly by secondary ordering**

Matches the exact characters in the search text, distinguishing upper/lower case as well as other forms. Using the example above, this option would allow the search text `bad` to only match `bad`.

**Even those normally ignored**

Matches the exact characters in the search text, distinguishing ignore characters defined in the sort order. For example, consider the hyphen defined as an ignore character and the following records:

- labas outside
- lab-as to sell fish
- lab-og to wallow
- labog to discard

Using this option, a search for `labas`, would display only `labas`—the other Match Character options would also match `lab-as`; a search for `lab-` would match `lab-as` and `lab-og`—the other options would also match `labas` and `labog`.

---

**Search: An Overview**

This command allows you to search for a text string at the beginning of the primary sort field in the active database. Records which match the specified parameters are displayed. When there is no match, this feature allows you to insert a new record in the database or try again with new search parameters.

**To initiate a search**

Do the following:

1. From the Database menu, choose Search or use the shortcut keys, ALT+S.
2. The Search dialog box will appear with the following options:

   - **Search for**
     Specifies the search text. Note the primary sort field automatically appears as the field to search in.
   - **Match whole field**
     Recognizes a match only when the text string matches the complete contents of the field. When this option is not used, any text string that begins with the characters in the search text—including a phrase, a word, or part of a word—will be matched.
Match Characters determines how characters are compared (See details above.)

A search in Toolbox results in one of the following:

- The display of the record matching the search text (when only one record matches)
- The Multiple Matches dialog box (when more than one record matches the search text)
- The No Matches dialog box (when no records match the search text).

**Multiple Matches dialog box**

This dialog box is used by the Search, Jump To, Jump Insert, or Insert a Record commands when the specified text is matched by more than one record, Toolbox displays these matches in the Multiple Matches dialog box along with the following options:

**OK**
Changes your position in the database to the record highlighted in the match list. This can also be accomplished by double-clicking on the highlighted text.

**Cancel**
Cancels the search

**Insert**
Allows you to add a record to the database which does not match the text you have typed.

**Try Again**
Returns to the Search, Jump To, Jump Insert, or Insert a Record dialog box allowing you to enter new text and then repeat the action.

**Additional field to view**
Allows you to display another field in the dialog box window. For example, in a lexical database including the part of speech, *ps*, might help in differentiating similar occurrences of the same text. The example screenshot has included the lexeme since the database was sorted by \ge.

**Show path**
This check box allows you to change the list display to include the entire path—the drive and directories. For example, instead of LEX.DB, the display might be C:\Toolbox\Lexicon\LEX.DB. *This option is only applicable when using the Jump To command; otherwise nothing happens.*

**No Matches dialog box**

When the specified search or jump text is not matched, Toolbox displays the No Matches dialog box with the following options:

**Insert**
Adds a new record with the unmatched text entered into the record marker field.

- If you are **searching**, this new record is inserted in the current database. *Be sure you are searching the record marker field, as this will insert a new record even if you’re searching the gloss field.*
- If you are **jumping**, Toolbox can insert it into any destination database that is open and is not read-only as
long as the destination field is the record marker. If there is more than one such available destination, Toolbox displays the Select Database dialog box so that you can choose the correct place to insert the new record. This button will only be available if there is at least one database open with write permission whose record marker is specified as a destination in the primary jump path.

**Try Again**

Lets you repeat the search or jump. Returns to the Search or Jump To dialog box where you can:

- change the text
- alter the matching criteria
- select a different Jump Path.

**Cancel**

Cancels the search or jump.

---

**Find (Edit Menu)**

This command allows you to locate a text string within the specified field(s) of an active database. Unlike the Jump To and Search commands, Find is not limited to matching text at the beginning of a field of data. It can locate text anywhere within a field.

**To find a text string**

Do the following:

1. From the Edit menu, choose Find or use the shortcut keys, CTRL+F.
2. The Find dialog box opens with the following options:

**Text to Find**

Specifies the text string to be matched. A space within the text will match any white space.

User-defined variables and built-in variables can be used to match certain texts. Square brackets ([ ]) should be used when entering a variable. For example:

- [...] Skip variable
- [#] Word boundary variable
- [vowel] User-defined variable

Characters used as special symbols by the Toolbox program, can also be found in fields of data.

**Find In**

Specifies where to look for text. You may choose a specific field or all fields assigned to a specific language. You are also given a choice between looking only in the current record or in the whole (filtered) database.

**Match whole word**

When enabled, matches only whole words. For example, if the is the text string, it will only match the word the. When disabled, it would also match text strings such as, them, other, bathe, etc.

**Match characters**

Match characters are controls which specify the way Toolbox determines a match.

**To Find a Marker**

To find a marker, choose it as the field to find in, and leave the text to find blank.

**The Find button(s) and Keyboard Shortcuts**

The Find button on the Toolbar brings up the Find dialog box just as the Edit, Find command does. Also, the Find dialog box can be invoked by the keyboard shortcut CTRL+F. Similarly, Find Previous and Find Next have both buttons on the Toolbar and keyboard shortcuts.
Jump, an Overview

Jumping: Three forms
In Toolbox, jumping means moving to another record, possibly in another database, without losing your place in the record you are currently in. You can return to your original place either via the Return from Jump command or by switching focus to the window containing your original record. The object of the jump may be a word or any string. Toolbox will look for that string in the same or other databases, depending on the criteria specified.

Jumping with a "jump path" specified.
This path specifies explicit databases and fields which will be examined for the object of the jump. When a match is found a new window is opened, sorted by the field specified, and displaying the record containing the text string that was being jumped to.

This is useful for examining other information about a word or phrase. One can jump from text to a lexicon or concordance, or from a concordance reference or word list reference into a text.

If no jump path is specified,
then Toolbox will examine the sort indexes of all the currently open database views for the specified string. (Since a database may be sorted in different ways on different fields, it is each view, not each database, which is examined.) For each view in which the text string is found, that view will be changed to display that record.

This is useful when you have two views of a wordlist, one sorted normally and one sorted from the end of the word. Doing a right click on a word in the text lets you see all the words which begin and which end the same as the word you are examining. This helps find possible prefixes and suffixes. It is also useful when doing comparative analysis using multiple parallel texts in different languages and with the same referencing scheme. A right click on a concordance reference will cause all their displays to be shifted at once.

Jump Insert
is a special form of the jump command which assumes it should insert. Jump Insert requires that a jump path be specified. (If no jump path is specified, you just get a rather useless "No matches for jump" message.) Jump Insert has a different command from the usual jump.

This is particularly useful for jumping from interlinear text into the lexicon, inserting morphemes and words which did not analyze. Normal jump usually works fine for this, but if another word or morpheme of the same form already exists in the lexicon, then regular jump does not offer an opportunity to insert another entry. In this case, you can use jump insert (an easy way is to hold down the Ctrl key while doing the right click).

Contrasting the three forms of jumping

Jump To with jump path
- Expects to find a record
- If a match is found, displays the window immediately
- Requests the user to choose if more than one match is found
- Displays a single window even if multiple databases were on the jump path
- If a match is not found, offers the option of inserting a new record
Jump To without jump path
- Expects to find a record or records
- If a match is found, displays the window(s) immediately
- Displays all where multiple matches are found
- Displays as many windows as match the text string
- If a match is not found, "No matches for jump" message is displayed. Focus remains at the original window, but the text string is inserted onto the clipboard so it can be pasted readily in if the user chooses to insert a new record.

Jump Insert (jump path required)
- Expects to insert a new record
- If a match is found, a dialog box still offers the insert option
- Requests the user to choose if more than one match is found
- Displays a single window even if multiple databases were on the jump path
- If a match is not found, inserts a new record with the text string as the contents of the record marker.

Jump To (Edit Menu)
This command allows you to move to another record without losing your place in the record or text currently displayed. Toolbox accomplishes this by displaying a window on the destination database which is:
- Positioned at the record matching the jump text.
- Sorted by the marker matched in the jump path
- or sorted by a field of the same language encoding as the source word of the jump.

Text from the beginning of any field in any open database can be matched with this feature. If the specific marker and filename are included in the jump path, Toolbox will examine those fields. If a jump path is not specified, Toolbox will examine the sort field of all the open databases.

Jumping is most useful for finding another record related to the data that is visible in the record you are currently viewing. For example, when viewing an interlinear text, you may want to jump from a morpheme in the \m field to the lexeme in the \lx or \a field. When you jump, Toolbox will search the jump path destination(s) for the current data. The "current data" is one of the following:
- the selected text, if any. (The selection must be within a single field.)
- the data item that the mouse pointer is on or next to.
- in browse view, the contents of the field pointed at. (Note: If the field contains multiple words, the entire contents will be used regardless of the data properties settings.)

Multiple window Jump:
If there is no Jump Path defined, then Toolbox will examine the open database windows for any which are sorted by the same language encoding as the current data. Each window in which it finds the current data as the sort field will jump.

The fastest way to Jump is to place the mouse pointer within the desired data item or highlight the text to look for and click the right mouse button. This action causes Toolbox to perform the Jump To command automatically—bypassing the Jump To dialog box and using the options previously established.

If the data you want to look for is not visible in the current record or it needs to be modified in some way (e.g., removal of affixes that change the root), or if you need to change the setup of the jump paths or Jump To options, do one of the following:
- Use the shortcut keys, ALT+J.
- From the Edit menu, choose Jump To.
Toolbox will display the Jump To dialog box and allow you to make the needed changes.
If you have established a jump path, when you use the Jump To command, Toolbox will display one of the following:

- the record matching the jump text (when one record matches)
- the Multiple Matches dialog box (when more than one record matches the jump text)
- the No Matches dialog box (when no records match the jump text).

If Toolbox displays one of the above dialog boxes, you can click the Try Again button to return to the Jump To dialog box and modify the text to look for or the jumping options.

If you have not established a jump path, when you perform a right click, Toolbox will do one of the following:

- shift all open windows in which the jump text was found to display the jump text
- display the message "Not found"

There is no dialog box for revising the jump text and trying again.

**Using a Jump Target**

*Note:* Using the Jump Target command (View menu) allows you to "target" a database window as the destination for future jumps. The target window must be sorted by the marker in the jump path in order to be matched in the jump or be sorted by the same language encoding as the jump text if no jump path is set up.

With the Jump Target set, Toolbox moves to the record matching the jump text by positioning the database at that particular record.

Without the Jump Target set, Toolbox moves to the record matching the jump text by opening a new window on the destination database which is positioned at that particular record.

If the current database window is maximized, the destination window will also be maximized. At times the only obvious difference between the two is the number after the database filename on the title bar.

**Troubleshooting tips**

If you place the mouse pointer on a word and click the right mouse button and

- Toolbox beeps and does not perform the Jump To command, or
- the Jump To command looks for only part of the word you clicked;

check the language encodings specified for the field marker you are jumping from. Every character in the word must be defined in the Sort Order for the language for this feature to work correctly.

If a match is found for data that does not appear to match, check to make sure that the source and the destination fields have the correct language encodings and fonts defined. If they are different, Toolbox will compare the data based on the language encoding for the destination field(s). If the underlying character codes are the same, Toolbox will identify the match even though they may be displayed using different fonts in the source and destination databases.

If more matches than expected are found, make sure that all the characters in the word are included in the language encoding of the destination field(s). Also, make sure that your match criteria are set appropriately.

If fewer matches than expected are found, make sure that the jump path is set up correctly and your match criteria are set appropriately. Also, be sure all destination databases are open. (Toolbox will warn you the first time it encounters a closed destination database.)

**Jump Insert (Edit menu)**

This command allows you to add (insert) a new record to any open database in the current jump path without leaving the record(s) or text(s) currently displayed. Toolbox inserts the record into a duplicate window on the database.

Jump Insert is somewhat similar to the Jump To command—both allow you to go to or insert another record. However, Jump Insert is used when you don't expect the record to be in a database since insertion quicker and more convenient with Jump Insert than with Jump To.
Jump Insert also functions in somewhat the same way as the Insert Record command with two exceptions:

- Jump Insert allows the record to be added to any open database in the current jump path*, while Insert Record only allows the record to be added to the active database.
- Jump Insert opens a duplicate window on the database allowing you to retain your current position in the database. Insert Record adds the record to the active database and positions it within the database according to the primary sort field.

To insert a record with a jump

Do the following:

1. Place the mouse pointer within the desired word or highlight a text string and choose an option:
   - Use the shortcut keys, CTRL+J.
   - From the Edit menu, choose Jump Insert.
2. Either option above causes the Jump Insert dialog box to appear with the desired word or text string in the dialog box as the record to insert.

Note: In Jump Insert, Toolbox displays the Multiple Matches dialog box if at least one record matches the text. Text from any field in any open database can be matched, if the specific marker and filename are included in the jump path.

If no records are matched the Select Database dialog box opens allowing you to insert the record into any open database in the current jump path.*

* To be able to insert into an open database on the current jump path, the database must not be read-only, and the destination field specified in the jump path must be the record marker in the destination database.

Matches dialog box will be displayed with all records which begin with the data you are inserting.

Match characters

Match characters control how Toolbox determines the match.

Jump Path tab

Database Type Properties dialog box (Properties command, Database menu)

This tab allows you to specify jump paths (lists of destination database(s) and field(s) associated with source fields) which will be searched for a possible match of the jump text. Each entry in the path is considered in its order of placement within the list. The following options are available in the Jump Path tab:

Name list
Displays a list of all Jump Paths defined for the database type. Edit the list by using the buttons on the right.

Add button
Adds a new jump path by accessing the Jump Path Properties dialog box.

Copy button
Copies the target information (list of target database(s) and field(s)) from the selected jump path by accessing the Jump Path Properties dialog box.

Modify button
Modifies the selected jump path by accessing the Jump Path Properties dialog box.

Delete button
Deletes the selected jump path.
Notify me before closing a database referenced by a jump path
This check box allows you to choose whether or not Toolbox will display a dialog box before closing a database referenced by a jump path. This setting is remembered for the entire project, not just this database type.

Note: Choosing Cancel does not undo Add, Copy, Modify or Delete.

Return from Jump
(Edit menu)
The Return from Jump command provides a quick way (via the shortcut keys Ctrl-R) to return to the database you were working in from the previous jump.

If the database being returned to is an interlinear database, and if the record you are returning from was modified, then Toolbox invokes Interlinear upon return. Toolbox assumes you have Jumped from the interlinear text into the dictionary and that the modification you made is wanted for the interlinear text.

Return from Jump works even if you go to other records and other databases after the jump.

However, Return from Jump returns only to the window it came from. If the window you are in is a Jump Target and the jump only changes the view to display the new record, Return from Jump will not cause the former record to be displayed.

Return from Jump does not nest. That is, if you have jumped from database A to B and from B to C, the Return from Jump will only return you to database B.

Jump To command
(Edit menu)
Toolbox displays this dialog box when

- the Jump To command results in no matches or multiple matches and you click the Try Again button on the Multiple Matches dialog box or No Matches dialog box.
- you choose Edit, Jump To (as opposed to the mouse shortcut)

The Jump To dialog box allows you to specify several options that affect the way the Jump To command works:

Record to Jump To
Specifies the text to look for. Initially, this will usually be filled in with the current data item or selected data.

Language Encoding
Changes fonts and keyboard to facilitate viewing and typing the above text. (This does not affect the actual search in any way.)

Primary Jump Path
Displays the set of all jump paths defined for this type of database. Toolbox initially selects the primary jump path based on the field you were in when you clicked the right mouse button or chose Jump To from the Edit Menu. If this field is not included in the sources for a specific jump path, then the default jump path, if any, will be chosen as the primary jump path. If you are attempting a Jump To operation where your current cursor position should not be interpreted as the “source” of the jump, then you should select the primary jump path that correctly reflects the destination(s) you intend to search.

Match whole field
When selected, exactly matches the contents of the whole destination field. When cleared, matches a text string at the beginning of the destination field.

Match characters
Match characters are controls which specify the way Toolbox determines a match.

Jump Path
This button accesses the Jump Path Tab. Use this to specify database(s) and field(s) to be considered when matching the jump text.
Jumping Multiple Windows

It is possible to cause multiple windows to jump at the same time. This capability is enabled by having no jump path specified. The program then responds as it does to an external program jump and searches the sort index of each of the open windows. If the string is found, that window is moved to that record.

For example, you might have a text and a word list built on it. You could have two views of the word list—one sorted by the words, and the other sorting the words from the end. If you right-click on a word in the text, then both word lists will jump to that word. The one will have as context all the other words that begin the same way, and the other will have as context all the other words that end the same way. This is a useful way to examine text for prefixes and suffixes.

Multiple window jump is used for parallel jump. In parallel jump, whenever a window is moved to a different record, all other windows that are sorted by the same language jump to follow it. One use of this is to have a number of windows sorted by the same reference system. When one window is moved to a new referenced unit, all other windows move to the same reference.

Parallel jump is always active by default. If you don't want a particular window to follow the others, turn off its Jump Target setting.

Jump Path Warning dialog boxes

Database Location Changed? dialog box

This dialog box pops up to warn you if Toolbox is searching a jump path which refers to a destination database which is closed, but another database with the same name is open. Typically, this can happen if you have copied or moved a database from one location (directory) to another. It is similar to the Database Not Open dialog box, except that in this case Toolbox gives you the option to correct the destination of this and all other jump paths for this type of database so that they point to the open database instead.

The information displayed is:

Jump Path Name

This indicates the specific jump path that is being searched. Note there may be other jump paths that also contain a reference to the closed database. The decision you make here may affect other jump paths besides the one shown.

Closed Database

This indicates the path and filename of the database which is not open and therefore can't be searched.

Open Database with Same Name

This indicates the path and filename of the open database which Toolbox has found (Note: if two databases are open with the same name as the closed database, Toolbox arbitrarily chooses one).

The options are:

Fix It

Press this button to change this and any other jump paths to use the open database instead of the closed database (this change will be saved in your settings files).

Skip

Press this button to continue searching any remaining paths and databases and ignore this closed database in all future jumps that refer to it, for the remainder of this session or until you
open it. This allows for the somewhat advanced option of having databases on the jump path that are sometimes intentionally closed to reduce clutter or system demands.

Cancel
Press this button to cancel the Jump To operation. Choose this option if you think the path is set up incorrectly, but the fix requires manual intervention (i.e., it can't be fixed simply by changing the reference from the closed database to the open one listed). Once you have set up the jump paths correctly, you can repeat the Jump To operation.
Jump Insert dialog box
Jump Insert command (Edit menu)

When you choose the Jump Insert command in the Edit menu, this dialog box opens with the following options:

**Record to insert**
- Specifies the data to be inserted in the record marker field in a new record.

**Language Encoding**
- Allows you to specify the language for the record you will add to a database.

**Match whole field**
- When chosen, Toolbox only displays the Multiple Matches dialog box if the data you are inserting matches the whole field of a record already in the database. If not chosen, the Multiple Matches dialog box will be displayed with all records which begin with the data you are inserting.

**Match characters**
- Match characters control how determines the match.

**Jump Path**
- This button accesses the Jump Path Tab. Use this to specify database(s) and field(s) to be considered when matching a record.

**Note:** In Jump Insert, Toolbox displays the Multiple Matches dialog box if at least one record matches the text. Text from any field in any database can be matched, if the specific marker and filename are included in the jump path.

If no records are matched the Select Database dialog box opens allowing you to insert the record into any open database in the current jump path.

Database Not Open dialog box

**Jump Path Warning dialog boxes**
- This dialog box pops up to warn you if Toolbox is searching a jump path which refers to a destination database which is closed. Typically, this can happen if you closed the database in a prior session and forgot to reopen it. It can also happen when a database is inadvertently closed and the project is not set to warn you when you close a database referenced by a jump path (see setting up jump paths to learn how to turn on this warning). This dialog box is similar to the Database Location Changed? dialog box, except that in this case there is no option to fix the problem automatically.

The information displayed is:

**Jump Path Name**
- This indicates the specific jump path that is being searched. Note there may be other jump paths that also contain a reference to the closed database. The decision you make here may affect other jump paths besides the one shown.

**Closed Database**
- This indicates the path and filename of the database which is not open and therefore can't be searched.
Skip
Press this button to continue searching any remaining paths and databases and ignore this closed database in all future jumps that refer to it, for the remainder of this session or until you open it. This allows for the somewhat advanced option of having databases on the jump path that are sometimes intentionally closed to reduce clutter or system demands.

Cancel
Press this button to cancel the Jump To operation. Choose this option if you think the path is set up incorrectly, or if you would like to stop searching and open the closed database. Once you have set up the jump path correctly or opened the database, you can repeat the Jump To operation.

Jump Path Properties dialog box

Jump Path tab (Database Type Properties, Properties command, Database menu)
This dialog box is where jump paths are defined and modified. It is accessed through the Jump Path tab in the Database Type Properties dialog box and offers the following options:

Jump Path Name
Specifies the name of the Jump Path that you want to create or modify. The name should describe the purpose of the jump path, which may reflect where you are jumping from, where you are jumping to, or both. While there is no practical limit to the number of characters in the name, only 14 or 15 will be displayed at any one time in the dialog box.

Default Path
One default jump path is allowed per database type. If another jump path is not already marked as the default, this check box will be enabled. If you check this option, any specific source fields selected for this jump path will be forgotten.

Source
This section will be enabled unless you are creating or modifying the default jump path. It is similar to the Select Fields dialog box. Fields that you include as source fields will use this jump path as their primary jump path. That is, if you select Jump To from the Edit menu while the cursor is in one of these fields, this is the jump path that will be searched first (followed by the default jump path, if any, if no match is found). Once a field has been selected as the source of a jump path, it will not be in the list of available fields to be included in the source of another jump path. This ensures that Toolbox will consistently choose the correct primary jump path when jumping from data in any given field.

Destination
In this section, you specify the fields of specific databases that should be searched when Toolbox uses this jump path to find where to jump to. Each jump path destination consists of a specific database (in a particular location) and a field within that database. To define destinations, use the following controls:

Available Databases
Lists the database(s) currently open, even if minimized.

Field to Search
This combo box contains a list of all field markers used in the highlighted database above. Click on the arrow to display the available markers. Highlight the marker to be added to the path.
Databases in Path—Field to Search

This list shows the actual jump path target(s). Entries are moved in and out of this list with the following buttons:

- **First ->** Places database and field first in path
- **Last ->** Places database and field last in path
- **Insert ->** Places database and field next in path
- **< Remove** Removes database and field from path
- **< Clear** Removes all entries from path

**Show whole path**
When checked, changes the list display to include the entire path—the drive and directories. For example, instead of LEX.DB, the display might be C:\Toolbox\Lexicon\LEX.DB.

**Establish Data Link**
Select this option to tell Toolbox to ensure that cross-referenced data exists. If you establish a data link, Toolbox will check your data to verify that target data exist for the data in this jump path's source fields. Note that this checkbox is not enabled for the default jump path.

**Matching**
Choose Matching to display the Jump Path Matching Criteria dialog box, where you can modify the character-matching criteria for this jump path.
After you have edited a record, the appropriate range sets are checked automatically.

Toolbox can also check consistency of all records in a window. If it finds an inconsistency, it displays the Data Link Consistency Check dialog box.

**Tip:** Establishing a data link makes the most sense for a jump path where one or two sources link to a single destination, or where a single source links to a small number of destinations. If you think of your database as a hypertext document, establishing a data link is the way to ensure that clicking on a link to perform a jump will actually take you somewhere.

**Match Characters**
Match characters are controls which specify the way Toolbox determines a match. For the options below, consider an example with primary groupings in the sort order of:

- A a á à
- B b
- D d

**By primary grouping only**
- Allows any characters grouped together in the current sort order to be considered identical. Using the example above, this option would allow the search text **bad** to match **bad, bád, bàd** and **Bad**.

**Disregarding case**
- Matches the exact characters in the search text, but does not make an upper/lower case distinction. Using the example above, this option would allow the search text **bad** to match **bad** and **Bad**, not **bád** or **bàd**.
**Exactly by secondary ordering**

- Matches the exact characters in the search text, distinguishing upper/lower case as well. Using the example above, this option would allow the search text `bad` to only match `bad`.

**Even those normally ignored**

- Matches the exact characters in the search text, distinguishing ignore characters defined in the sort order. For example, consider the hyphen defined as an ignore character and the following records:
  - `labas`  outside
  - `lab-as` to sell fish
  - `lab-og` to wallow
  - `labog` to discard

Using this option, `labas`, would match only `labas`—the other options would also match `lab-as`; a search for `lab-` would match `lab-as` and `lab-og`—the other options would also match `labas` and `labog`.

**Select Database dialog box (Jump Insert command)**

This dialog box appears when no records are matched using the Jump Insert command. It also appears if no records are matched using the Jump To command and you choose the Insert option.

**Database to insert new record into**

This is a list of the open destination databases in the primary jump path. Read-only databases are excluded. Select the database you want to insert the new record into and choose **OK**.

**Show Path**

Changes the list display to include the entire path—the drive and folders (directories). For example, instead of MDFSampl.db, the display might be `C:\Toolbox\MDF\MDFSampl\MDFSampl.db`.

**Troubleshooting the Jump Path**

If the database you expected to insert into is not in the list, choose **Cancel**. If this dialog box doesn't appear and the new record is automatically inserted into the "wrong" database, delete the record. Now use the following checks to find the cause of the problem and fix it:

- Make sure the database is open. The first time each session that you use a jump path with a closed destination database, Toolbox will warn you with the **Database Not Open** dialog box. This is designed to alert you to this situation so that you will not be surprised when Toolbox fails to find the text you were jumping on or when the closed database does not appear in the Select Database dialog box.

- Make sure that the database is not read-only. If it was read-only at the time you opened it in Toolbox, you need to close it and make sure you have write permission before reopening it. To avoid losing some of the settings related to this database, it is probably best to close and reopen the entire project.

- Make sure that when you jump, you are using the correct primary jump path. Toolbox will attempt to select the appropriate jump path based on the field where the cursor is positioned when you jump. However, if the jump path source fields specify a jump path other than the one you intended, you need to select the correct primary jump path in the **Jump To** dialog box.

- Make sure that the intended destination database is actually included in the primary jump path you are using. If the database is in the default jump path but not in the primary jump path, Toolbox will not include it in the list of possible destinations. In this case, you can either explicitly select the default jump path as the primary one in the **Jump To** dialog box, or you can modify the primary jump path to include the intended destination.

- Toolbox can't automatically insert the unmatched text into a destination database whose destination marker is not the **record marker**. If this is what you need to do, you will have to manually insert the new record in the desired destination database and enter the appropriate data in the destination field (or find the appropriate existing record and add the needed data field).
For more information about setting up jump paths, see Jump Path tab.

Jump Path Matching Criteria
Jump Path Properties (Jump Path tab, Database Type Properties, Properties command, Database menu)

This dialog box is where the character-matching criteria for jump paths are defined and modified. It is accessed by choosing the Matching button on the Jump Path Properties dialog box and offers the following options:

Match whole field
When selected, exactly matches the contents of the whole field. When cleared, matches a text string at the beginning of the specified field(s).

Match characters
Match characters are controls which specify the way Toolbox determines a match.

Jump Path
This button accesses the Jump Path Tab. Use this to specify database(s) and field(s) to be considered when matching the jump text.

As an example of various options, consider the following record:

```
\lx lab-as
\ge to sell fish
```

If both the lx and ge fields were included in the primary jump path for the ge field, this record could be accessed by typing in lab-as or to sell fish. It would match to sell if you are not matching the whole field. However, if you tried to use fish as the jump text, it would not match, because in the Jump To command, Toolbox only matches a text string at the beginning of the specified field(s).

Note: The default matching criteria for a new jump path depend on whether it is established as a data link or not. By default, data links use the strictest matching, while other jump paths use more lenient matching.

Jump Insert (Edit menu)

This command allows you to add (insert) a new record to any open database in the current jump path without leaving the record(s) or text(s) currently displayed. Toolbox inserts the record into a duplicate window on the database.

Jump Insert is somewhat similar to the Jump To command—both allow you to go to or insert another record. However, Jump Insert is used when you don’t expect the record to be in a database since insertion quicker and more convenient with Jump Insert than with Jump To.

Jump Insert also functions in somewhat the same way as the Insert Record command with two exceptions:

- Jump Insert allows the record to be added to any open database in the current jump path, while Insert Record only allows the record to be added to the active database.
- Jump Insert opens a duplicate window on the database allowing you to retain your current position in the database. Insert Record adds the record to the active database and positions it within the database according to the primary sort field.
To insert a record with a jump
Do the following:

Place the mouse pointer within the desired word or highlight a text string and choose an option:

• Use the shortcut keys, CTRL+J.
• From the Edit menu, choose Jump Insert.

Either option above causes the Jump Insert dialog box to appear with the desired word or text string in the dialog box as the record to insert.

Note: In Jump Insert, Toolbox displays the Multiple Matches dialog box if at least one record matches the text. Text from any field in any open database can be matched, if the specific marker and filename are included in the jump path.

If no records are matched the Select Database dialog box opens allowing you to insert the record into any open database in the current jump path.

* To be able to insert into an open database on the current jump path, the database must not be read-only, and the destination field specified in the jump path must be the record marker in the destination database.

Matches dialog box will be displayed with all records which begin with the data you are inserting.

Match characters
Match characters control how Toolbox determines the match.

Jump Messages
Line Length, Line Wrapping
Topics Covered:
Auto Wrap (Database menu)
Set Wrap Margin (Database menu)
Reshape, Reshape All (Database Menu)
No Word Wrap (Database, Properties, Marker Properties, Data Properties)

Setting Line Length and Reshaping Line Breaks
Line length for the various fields in your database is not limited. The database window will scroll to the right as data is entered unless you set a wrap margin. Margins are set to the size of the current database window through the Set Wrap Margin command (Database Menu).

Once a margin is set, enabling the Auto Wrap command (Database Menu) turns on the feature which causes lines of data to automatically wrap at the margin settings when they are edited. If this feature is disabled (unchecked), line length will not be restricted within the window margins, and the database window will continue to scroll to the right as text is entered.

Sometimes certain fields of data should not be reshaped. For example, you may have fields where data is deliberately separated by line breaks. Each field has the option of “No word wrap”. This can be accessed by doing

Database
Properties
selecting the field
clicking on Modify (This takes you to the Marker Properties dialog box.)
choosing the Data Properties tab
checking the No Word Wrap box

Toolbox does not automatically reshape database fields when the margin settings are changed. You must specifically invoke one of the Reshape commands (Database Menu). Reshape applies only to that particular occurrence of a field. It does not apply to other occurrences of the same field elsewhere in the database. Reshape displays no dialog box. If you want to reshape all fields in the database, select the Reshape Entire File command (Database Menu).

Interlinear data can be safely Reshaped. Toolbox will break and wrap the entire bundle of lines belonging to a word as a unit.

To Reshape a single field
Do the following:

1. Place the insertion point in the field you wish to reshape
2. From the Database Menu, choose Reshape or use the shortcut keys, SHIFT+F5

To Reshape an Entire File
This command allows you to concatenate and re-break the contents of all of the fields in an entire file to fit within the current margin settings, except those with the No Wrap setting. As for the single-field Reshape, margins are set to the size of the current database window by using the Set Wrap Margin in the Database Menu.

To reshape all fields in the entire file, choose Reshape Entire File from the Database Menu. Toolbox then displays a dialog box and asks you to confirm or cancel the operation.
Browse (View menu)

The Browse command provides a view of multiple records in a database in rows and columns. It can be used to search for database records or display selected fields within the records.

**To enable or disable the Browse display**

Do one of the following:

- On the toolbar, click the Browse button.
- Use the shortcut keys, ALT+R.
- From the View menu, choose Browse.

**Note:** When a record does not contain a field that is being displayed in the browse view, Toolbox displays *no field* in that column. Similarly, if a record does contain a field but it is empty Toolbox displays *empty* in that column. The actual text of these special labels can be changed in the Browse Fields dialog box, which is displayed by doing View, Browse.

In the Browse view there are several ways to move through the records currently displayed:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next record</td>
<td>Next record button, ALT+N</td>
</tr>
<tr>
<td>Previous record</td>
<td>Previous record button, ALT+P</td>
</tr>
<tr>
<td>First record (Top)</td>
<td>First record button, ALT+T</td>
</tr>
<tr>
<td>Last record (Bottom)</td>
<td>Last record button, ALT+B</td>
</tr>
</tbody>
</table>

To switch to record view, do CTRL+R, double-click the record, or select the record and press ENTER.

To use the Jump To command to jump based on the contents of a field displayed in browse view, point to the field and click the right mouse button.

**Size of Browse Column**

The Browse view column width are initially equally divided among the specified columns. However, you can readily change the width of a column using the mouse.

Position the mouse cursor over the vertical line between columns until it becomes a double-headed arrow. Hold down the left mouse button and drag the line to the position you desire.

**Note:** Changing the Browse columns will reset the column widths.

**Note:** Also, reducing the window size will cause the width of the rightmost Browse columns to change as Toolbox tries to keep all the vertical lines in the window.

**Printing, Sorting, and Organizing the Browse View**

To print the Browse view

Choose Print from the File menu (or use CTRL+P).
To change the primary sort field while browsing
Click on the marker at the top of the column in browse view. The marker that was previously the primary sort field becomes a secondary sort field.

To sort a field from the end while browsing
Do Ctrl+click on the marker at the top of the column in browse view.

To shift the position of a browse column to the right while browsing
Do Ctrl+right click on the marker at the top of the column in browse view. (Column positions can also be adjusted in View, Browse Fields.)

To remove a browse column while browsing
If the browse column is already at the far right, doing Ctrl+right click on the marker at the top of the column will remove it from the browse view.

To add a marker to browse view from normal view
A quick way to add a marker to browse view is to do Ctrl+right click on the marker in normal view. The marker will be added to browse view, and browse view will be turned on so you can see the result.

To align a column from the right when not sorting by it
First, do Ctrl+click to make the column sort from the end. Then, click on the column you want to sort by. The other column will remain right-aligned.

**Browse Fields dialog box (View menu)**

This command is used to change the fields which are displayed in the Browse view. To change the display:

1. Choose Browse Fields in the View menu.

2. Use the dialog box to move record markers from the Available Fields box on the left to the Browse Fields box on the right. The Available Fields markers which appear are the ones inventoried in the Markers Tab in the Database Type Properties dialog box. To move the field markers between the boxes, select the marker and then choose the appropriate button, or double-click on the marker to place it next in the list.

   - *First ->* Places selected Available Fields marker first in Browse Fields list
   - *Last ->* Places selected Available Fields marker last in Browse Fields list
   - *Insert ->* Places selected Available Fields marker next in Browse Fields list
   - *< Remove* Removes selected marker from Browse Fields list
   - *< Clear* Clears all markers from Browse Fields list

3. When a record does not contain a field that is being displayed in the browse view, Toolbox displays *no field* in that column. Similarly, if a record does contain a field but it is empty Toolbox displays *empty* in that column.
To change what is displayed by Toolbox in the Browse view, edit the contents of the "Display for an empty Field:" and "Display if there is no Field:" edit boxes.

For example, you could specify that a dash be displayed if a field were empty, and specify that nothing at all be displayed where no field exists (do this by deleting the contents of that edit box).

Consistency Check
(Checks menu)

Toolbox can check the consistency of data in three ways:

1. If you edit a record, its data fields can be checked before any operation that could cause a different record to be displayed in that window (e.g. Next Record on the Database menu). To turn automatic checking on or off use the Check Consistency When Editing command. (When Toolbox does this kind of checking, any relevant consistency conditions are checked for all fields in the record.)

2. When you export a database, it is checked if you have selected that option in the export process properties. (All fields and all consistency conditions for all records to be exported are checked.)

3. When you select Consistency Check from the Checks menu, the records in the active database are checked. (If the window is filtered, any records that do not match the filter are not checked.)

If an inconsistency is found in the data, it is selected in the active window and a dialog box is displayed.

If you cancel checking from this dialog box and later in the session, with the same window active, select Checks, Continue Consistency Check Toolbox continues from the current insertion point. However, if you select Checks, New Consistency Check, Toolbox displays the dialog box (in case you want to change the options) and starts checking from the first record in the active (filtered) database.

As indicated above, for the types of checking described in 1 and 2, Toolbox does all relevant checks. However when you select Consistency Check or New Consistency Check from the Checks menu, Toolbox displays the Consistency Check dialog to allow you to indicate the fields to check and the types of checking to perform.

The Consistency Check dialog box

Fields to check

All Fields
The primary order of checking is by the Standard Format markers (in alphabetical order). All data fields for each marker are checked in the window's records from first to last (according to its filtering and sorting).

Single Field
To concentrate on the consistency of a particular field, select this option.

Selected Fields
To concentrate on the consistency of certain fields, select this option and use the Select Fields dialog box to choose the fields you wish to check.
Note that some markers may not require consistency checking. It is still okay to select All Fields or include them in the list of Selected Fields.

Check
To concentrate on particular consistency issues, select the individual options.

Data Properties
Toolbox will perform consistency checks based on Data Property settings to ensure that:
- single-word data does not contain more than one word;
- fields which require data are not empty.

When an inconsistency is found, the Data Properties Consistency Check dialog box is displayed.
Data Properties are established by doing Database, Properties, selecting the marker, clicking on Modify, and clicking on the Data Properties tab.

Range Sets
A range set defines a closed set of values that are valid for a particular data field (e.g. \ps Part of speech).
When an inconsistency is found, the Range Set Consistency Check dialog box is displayed.
A range set is established by doing Database, Properties, selecting the marker, clicking on Modify, and clicking on the Range Set tab.

Data Links
A jump path defines source fields (i.e., where you are jumping from) and jump path destinations (i.e., databases and fields you wish to search for a match to jump to). A data link is a special kind of jump path that is analogous to a hypertext link. If you establish and check data links, Toolbox can ensure that all "links" go somewhere. When an inconsistency is found, the Data Link Consistency Check dialog box is displayed.

To establish a data link Jump Path, set up the Jump Path and choose the “Establish Data Link” option at the very bottom of the dialog box. See Jump Path for details.

Check Consistency When Editing
Toolbox can help keep your data consistent by checking the consistency of an edited record. This happens any time you attempt to perform an action that could potentially cause the current record to change or that would close the window in which the edited record is displayed. By default, this type of checking is turned on (a check mark appears beside the command name). This feature can be turned on or off by clicking Checks, Check Consistency When Editing.

Reasons why you might want to turn off automatic consistency checking:
- You anticipate a lot of inconsistencies in your existing data because you have recently added data links or range sets or you have changed the data properties of markers. If you want to focus on data entry for awhile, you can turn off checking and resolve the inconsistencies later.
- The person doing data entry is not aware of the consistency issues or is unfamiliar with Toolbox's consistency checking feature.
- Your working style typically involves entering related data in several related records such that there may be temporary inconsistencies. If you work this way, it may be better to periodically run Check Consistency from the Checks menu, rather than having Toolbox pester you as you work. This may be an issue especially if you need to use the Jump To feature a lot as you enter data since jumping will normally trigger a consistency check for all records being edited.

Note that this setting affects the entire project (not just the active database), and it will be saved from one session to the next.

Continue Consistency Check
When the Consistency Check detects an error, an opportunity for correction is presented. However, sometimes it is necessary to cancel out of the Check in order to deal appropriately with the inconsistency.
When that is done, the Continue Consistency Check option on the Checks menu becomes active (it is usually greyed-out). Choosing that option from the Checks menu, or pressing ALT+Y, will cause the Check to continue from the point where it was cancelled. If preferred, a new Consistency Check, starting from the beginning of the file, can be initiated instead.

**Data Property Consistency Check**

Toolbox displays this dialog box when a data field fails its data properties consistency check.

**Inconsistency**

There are three kinds of inconsistencies for the Data Property Consistency Check:

1. This data field expects a single "word."
2. This field requires data.
3. Explicit line break in data item.

**Failed Data**

The contents of the data field that failed the consistency check. Note that in the case of missing data, there is no failed data to display.

**Replacement**

When the dialog box is first displayed, this edit box contains a copy of the failed data. If the value isn't valid, you can modify the value by typing the correct data in this edit box.

**Skip button**

Skip the inconsistency and let checking continue.

**Skip All button**

Skip the inconsistency this time and every subsequent time it is encountered during this check.

**Replace button**

Replace the inconsistency with the contents of the Replacement edit box. Let checking continue starting at the replacement.

**Replace All button**

Replace the inconsistency this time and every subsequent time it is encountered during this check. Note that if checking is interrupted later, the replacement will not have been made in unchecked records.

**Cancel**

If the correction requires more than a simple replacement, or if you notice something else in the active window that you want to edit immediately, choose Cancel. The effect depends on which of the three ways that you were checking consistency:

1. After you have edited a record: Cancels the operation that could have caused a different record to be displayed in the window (e.g. Next Record on the Database menu). The record that contains the inconsistency (i.e. the one that you had been editing) is still displayed in the active window.
2. When you export a database: Cancels the export process.
3. When you use the Check Consistency menu item: Cancels the consistency check. The record that contains the inconsistency is displayed in the active window. Later in the same session, if you use Continue Consistency Check (or its Alt+Y keyboard shortcut) for the window again, Toolbox continues from the current insertion point.

**Data Link Consistency Check**

Toolbox displays this dialog box when it detects a broken data link when performing a consistency check.

Data links are a special kind of jump path that can be established using the Jump Path Properties dialog box.

**Inconsistency**

The message displayed here will indicate the name of the broken data link.
Failed Data
The item in the source data field that doesn't reference a valid destination.

Replacement
When the dialog box is first displayed, this edit box contains a copy of the failed data.
1. If the value is valid, you can add it to the destination database.
2. If the value isn't valid, you can modify the value by typing in this edit box.

Skip button
Skip the inconsistency and let checking continue.

Skip All button
Skip the inconsistency this time and every subsequent time it is encountered during this check.

Replace button
Replace the inconsistency with the contents of the Replacement edit box. Let checking continue starting at the replacement. If it is still inconsistent, this dialog box will be shown again immediately.

Replace All button
Replace the inconsistency this time and every subsequent time it is encountered during this check. If it is still inconsistent, this dialog box will be shown again immediately. Note that if checking is interrupted later, the replacement will not have been made in unchecked records.

Insert button
Add the failed data item as a new record in the destination database and let checking continue. Note that this is only available if the destination field is the record marker in the destination database.
Otherwise, you must cancel the check, manually search for an existing record where the destination data belongs or insert a new record in the destination database, add the destination data field and missing data, return to the source database, and resume checking (see below).

Cancel
If the correction requires more than a simple replacement, or if you notice something else in the active window that you want to edit immediately, choose Cancel. The effect depends on which of the three ways that you were checking consistency:
1. After you have edited a record: Cancels the operation that could have caused a different record to be displayed in the window (e.g. Next Record on the Database menu). The record that contains the inconsistency (i.e. the one that you had been editing) is still displayed in the active window.
2. When you export a database: Cancels the export process.
3. When you use the Check Consistency menu item: Cancels the consistency check. The record that contains the inconsistency is displayed in the active window. Later in the same session, if you use this menu item (or its Alt+Y keyboard shortcut) for the window again, Toolbox continues from the current insertion point.

Range Set Consistency Check dialog box
Toolbox displays this dialog box when a data field that has a range set fails its consistency check.

Inconsistency
There are three kinds of inconsistencies for the Range Set Consistency Check:
1. Data item not found in range set.
2. Character "x" not found in range set. (This type of inconsistency will only appear for character-based range sets).
3. This field requires data.
**Failed Data**
The item in the data field that doesn't match the range set—it is also selected in the active window. Note that for inconsistent characters, the whole data item will be displayed here; the invalid character will appear in the Inconsistency message and will be selected in the Replacement edit box. If you do not understand why Toolbox has identified this data as an inconsistency, see Troubleshooting Range Sets.

**Replacement**
When the dialog box is first displayed, this edit box contains a copy of the failed data.

1. If the value is valid but isn't already in the range set, you can choose to add it.
2. If the value isn't valid:
   a. you can select a value from the Range Set list box (it is copied into this edit box) or
   b. you can modify the value by typing in this edit box.

**Range Set**
A list of the valid values for the data field. If you click a value, it is copied into the Replacement edit box. If you double-click, it is copied and Replace is chosen.

**Skip button**
Skip the inconsistency and let checking continue.

**Skip All button**
Skip the inconsistency this time and every subsequent time it is encountered during this check.

**Replace button**
Replace the inconsistency with the contents of the Replacement edit box. Let checking continue starting at the replacement. If it is still inconsistent, this dialog box will be shown again immediately.

**Replace All button**
Replace the inconsistency this time and every subsequent time it is encountered during this check. If it is still inconsistent, this dialog box will be shown again immediately. Note that if checking is interrupted later, the replacement will not have been made in unchecked records.

**Add to Range Set button**
Add the failed data item or character to the range set for this marker and let checking continue.

**Cancel**
If the correction requires more than a simple replacement, or if you notice something else in the active window that you want to edit immediately, choose Cancel. The effect depends on which of the three ways that you were checking consistency:

1. After you have edited a record: Cancels the operation that could have caused a different record to be displayed in the window (e.g. Next Record on the Database menu). The record that contains the inconsistency (i.e. the one that you had been editing) is still displayed in the active window.
2. When you export a database: Cancels the export process.
3. When you use the Check Consistency menu item: Cancels the consistency check. The record that contains the inconsistency is displayed in the active window. Later in the same session, if you use this menu item (or its Alt+Y keyboard shortcut) for the window again, Toolbox continues from the current insertion point.
Filters

Using filters allows you to view subsets of the records in your database. The subset will consist of all records which passed through the filter. When a filter is active, database commands (Next, Previous, Top, Bottom, Search, Browse, etc.) function only in this subset.

The following is an example of a filter that one might use in a lexical database:

<table>
<thead>
<tr>
<th>Name</th>
<th>Condition (Syntax)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pronouns</td>
<td>ps pron</td>
</tr>
</tbody>
</table>

This filter will temporarily narrow your focus in the database to those records which are pronouns. The name of the active filter is displayed in the Active Filter combo box on the far right side of the toolbar. When no filter is active, [no filter] is displayed:

Filters are defined as a property of a database type.

Making a Filter

Do one of the following:

1. From the Database menu, choose Properties.  
   The Database Type Properties dialog box opens.
2. Choose the Filters tab.
   The Filters tab offers the following option buttons:

<table>
<thead>
<tr>
<th>Add</th>
<th>Copy</th>
<th>Modify</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adds a new filter by accessing the Filter Properties dialog box.</td>
<td>Copies a filter by accessing the Filter Properties dialog box.</td>
<td>Modifies a filter by accessing the Filter Properties dialog box.</td>
<td>Deletes the highlighted filter if not currently active. This button is not available when the selected filter is currently active.</td>
</tr>
</tbody>
</table>

The first three buttons will take you to the Filter Properties dialog box.

The Filter Properties dialog box

Filter Name
Specifies the name of the filter that you want to create. The name should describe the action of the filter. While there is no practical limit to the number of characters in the name, only 14 or 15 will be displayed in the Active Filter combo box or at any one time in the dialog box. Spaces and backslashes are allowed.

Filter Elements
This list, on the left side of the dialog box, specifies the various elements you can use to make a simple or complex filter. Quite complex filters can be created by combining the primary filter elements using And, Or, Not, and the Opening and Closing parentheses.
These elements are described below after the primary filter elements: Marker Text, Marker Date, Language Text, With, and Non-Unique.

The filter elements are:

- Marker Text
- Marker Date
- Language Text
- And
- Or
- Not
- ( (Opening parenthesis)
- ) (Closing parenthesis)
- With
- Non-Unique

These elements are defined in detail below, after the end of the dialog box description.

**Element / Text list (Current condition of the filter)**

This large box on the right shows each portion of the filter as it is created or modified. To move elements in and out of this box, use the buttons in the middle between the Filter Elements list and the Element / Text list:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>Places the selected element first in the filter</td>
</tr>
<tr>
<td>Insert</td>
<td>Places the selected element next in the filter</td>
</tr>
<tr>
<td>Last</td>
<td>Places selected element last in the filter</td>
</tr>
<tr>
<td>Modify</td>
<td>Modifies the selected Marker Text, Marker Date or Language Text filter element.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the selected element from the filter</td>
</tr>
<tr>
<td>Delete All</td>
<td>Deletes all elements from the filter</td>
</tr>
</tbody>
</table>

**Match whole field checkbox**

When *enabled*, exactly matches the contents of the whole field. When *disabled*, matches filter condition anywhere within the specified field(s).

**Match characters (drop box)**

Match Characters control how Toolbox determines the match.

There is no practical limit to the number of filters that you can define.

If you are going to sort the database by a marker other than the record marker field and use a filter, do the sorting first. If you filter the database first, Toolbox will build the filter index according to the current sorting. Then, when you change the sorting it will have to rebuild the filter index as well as building the sorting index for the whole (unfiltered) database.

**Note**: When a filter is selected, right-clicking on the Active Filter combo box will automatically access the Filter Properties dialog box for easy modification to any of the properties.
The Filter Elements

Marker Text filter element

This filter element is used to specify the field marker and any text included in the filter condition. When choosing this element, the Marker Text Filter Element dialog box appears offering the following options:

**Marker**: Specifies the field to be matched in the filter condition.

**Text**: Specifies the text to be matched in the filter. If no text is entered, it will match all records containing the specified field. If text is entered, it matches text anywhere within the specified field. A space within the text will match one or more of any white space (i.e. space, tab or new line).

**Variable**: Lists the user-defined variables available for the languages used in this project

**Insert into the text**: Inserts the selected variable into the text box of the filter element.

Examples of possible uses for this filter element:

<table>
<thead>
<tr>
<th>Marker</th>
<th>Text</th>
<th>Matches records where</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ps</td>
<td>adj</td>
<td>part of speech is an adjective</td>
</tr>
<tr>
<td>\hm</td>
<td></td>
<td>all records contain the \hm field</td>
</tr>
<tr>
<td>\ge</td>
<td>tree</td>
<td>the English gloss field contains <em>tree</em></td>
</tr>
<tr>
<td>\ge</td>
<td>[#]man[#]</td>
<td>the English gloss field contains the word <em>man</em> (this expression uses the Word Boundary variable)</td>
</tr>
<tr>
<td>\lx</td>
<td>[#][U]</td>
<td>the <em>lx</em> (lexeme) field contains any word beginning with an uppercase letter</td>
</tr>
</tbody>
</table>

**Note**: The following characters – the backslash (\), the opening square bracket ([) and the closing square bracket (]) – are used as special symbols by the Toolbox program, but can also occur in fields of data. When these characters are included as text within the Marker Text element of the Filter definition, specify them by preceding them with a backslash (\):
Marker Date filter element
This filter element is used to specify the date marker, and the date and conditions which the filter will use to match records that were created or edited before, on or after a specified date. When choosing this element, the Marker Date Filter Element dialog box appears offering the following options:

Marker: Specifies the date stamp marker – a Toolbox feature that inserts the current date into a record when it is created or edited—usually \dt.
Before, Equal or After: Specifies the conditions of the date.
Date: Specifies the date written in the dd/Mmm/yyyy format, using a three-letter abbreviation for the month (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec).

Examples of possible uses for this filter element:

<table>
<thead>
<tr>
<th>Marker</th>
<th>Condition and Date</th>
<th>Matches records where</th>
</tr>
</thead>
<tbody>
<tr>
<td>\dt</td>
<td>&lt; 15/Mar/1995</td>
<td>date stamp is before March 15, 1995</td>
</tr>
<tr>
<td>\dt</td>
<td>= 1/Feb/1996</td>
<td>date stamp is February 1, 1996</td>
</tr>
<tr>
<td>\dt</td>
<td>&gt; 1/Jul/1996</td>
<td>date stamp is after July 1, 1996</td>
</tr>
</tbody>
</table>

A complex filter might also be built. For example:

\dt > 1/Feb/1996
And
\dt < 1/Jul/1996

This matches records which were created or edited between 1/Feb/1996 and 1/Jul/1996.

Language Text filter element
This filter element is used to specify the language and any text included in the filter. When choosing this element, the Language Text Filter Element dialog box appears offering the following options:

Language: Specifies the language(s) to be matched in the filter condition. If you want to look in all language fields, use the Any lang. option in the list.
Text: Specifies the text to be matched in the filter condition. Specifying a language automatically gives you the proper keyboard and fonts to use when entering text. If no text is entered, it will match all records containing the specified language. If text is entered, it matches text anywhere within the specified language fields. A space within the text will match one or more of any white space (i.e. space, tab or new line).
Variable: Lists the user-defined variables available for the languages used in this project
Insert into the text: Inserts the selected variable into the text box of the filter element.
Examples of possible uses for this filter element (from a Tagalog lexical database):

<table>
<thead>
<tr>
<th>Language</th>
<th>Text</th>
<th>Matches records where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td></td>
<td>there are fields using Default</td>
</tr>
<tr>
<td>Tagalog</td>
<td>carabao</td>
<td>any Tagalog field contains the word carabao</td>
</tr>
<tr>
<td>Any Lang.</td>
<td>carabao</td>
<td>any language field contains the word carabao (for example, an English encyclopedic field might contain this Tagalog word)</td>
</tr>
</tbody>
</table>

**Note:** To match records with empty fields use Any Lang. with no text and check the Match whole field option.

The following characters are used as special symbols by the Toolbox program, but can also occur in fields of data. When these characters are included as text within the Language Text element of the Filter definition, specify them as indicated below:

<table>
<thead>
<tr>
<th>To Filter on this</th>
<th>Type this</th>
</tr>
</thead>
<tbody>
<tr>
<td>[</td>
<td>{</td>
</tr>
<tr>
<td>]</td>
<td>}</td>
</tr>
<tr>
<td>\</td>
<td>\</td>
</tr>
</tbody>
</table>
With filter element for interlinear text filters

Filters using the With element find text strings that are aligned vertically in two different fields. Its principal use is in interlinear text databases, where you might want to find all records in which a specific word is analyzed as a noun.

Consider these two contrasting examples.

\mb de    the article de in the mb line
With      lines up with (is used as)
\ps prep  a preposition in the ps line

As a contrast, consider the following filter using And instead of With:

\mb de    the article de in the mb line
And       is in the same record as
\ps prep  a preposition in the ps line

While this second filter would display the desired records, it would also include records where de occurs glossed as an indefinite article, and where prep occurs as the gloss of some other morpheme. Using With ensures that a record will only be displayed if the two filter elements are vertically aligned.

Non-Unique filter element

This filter element matches records with duplicate contents of the field(s) by which the database is sorted. The filter's main purpose is to identify and properly distinguish between homonyms in a lexicon. Depending on the number of homonyms and the initial state of the data, there are three ways to use the Non-Unique filter element to accomplish this:

- With the lexicon sorted by lexeme only, use a filter with Non-Unique as the only element. This filter will match all homonyms. By displaying lexemes and homonym numbers in browse view, you can manually scan the list and find records that are not distinguished by homonym numbers. After you have added the needed homonym numbers, you can add that field as a secondary sorting field to make sure you didn't accidentally assign the same number to multiple records with the same lexeme.

- With the lexicon sorted by lexeme and homonym, use a filter with Non-Unique as the only element. This filter will match all homonyms that have missing or duplicate homonym numbers. Note that this will not catch homonym pairs where one lexeme has a homonym number and the other doesn't. Therefore, after you have added or modified the homonym numbers as needed, you should sort by lexeme only and display both lexemes and homonym numbers in browse view. This will allow you to scan the data and check for missing homonym numbers.

- With the lexicon sorted by lexeme only, use the following complex filter:
  Non-Unique And Not \hm
This filter will match all groups of homonyms which have no homonym numbers and might be useful if you are just beginning to add homonym numbers to your lexicon. Once you have added homonym numbers, you will probably want to use the preceding approach to make sure the homonym numbers are truly unique for a given lexeme.

Note that when the active filter has a non-unique filter element, Toolbox adds the Reapply Filter command to the View menu and the toolbar. This lets you control when the data in the database is refiltered so that records don't disappear from a filtered view before you have had a chance to deal with all the homonyms in a group.

The And element

The And operator is used to perform a logical conjunction on two expressions. It returns true if all its arguments are true, and returns false if one or more arguments is false.

An example of a possible use for this filter element:

\sc
And
\dt > 5/Dec/1995
The **And** operator matches records where **both** conditions are true. In this case, records are matched where there is an **sc** (scientific name) field created or edited after 5/Dec/1995.

**The Or element**
The **Or** operator is used to perform a logical disjunction on two expressions. It returns **true** if either argument is **true**.

An example of a possible use for this filter element:

```plaintext
\sy Or \th
```

In the example above, all records are matched which have the **sy** (synonym) field, even if they don’t have the **th** (thesaurus) field. It also matches all records which have the **th** field, even if they don’t include any **sy** field. It also includes all records contain both **sy** and **th** fields.

**And vs Or**
There is sometimes confusion, when using the logical operators **And** and **Or**. For example (in a lexical database)

```plaintext
\ps n Only matches records which
  And function as **both** a noun and a
  \ps v  verb

\ps n Matches **all** nouns and verbs
  Or in the database
  \ps v
```

The first filter selects only records which contain **both ps n** and **ps v**. That results in a limited selection of entries which are both nouns and verbs.

The second filter selects all entries which contains **either ps n or ps v**. That results in all nouns and all verbs being selected.
The Not element
The **Not** operator is used to negate an expression when placed immediately before the filter definition. It reverses the logic.

An example of a possible use for this filter element:

```
Not  This example matches records which
    do not
\dt  have a dt (date) field.
```

**Not** can be used in conjunction with **And** and **Or**.

```
\ps n
And
Not
\pl
```

The example above matches records which are nouns (i.e., have a **ps** containing an **n**) and have no **pl** (plural) field.

The Opening and Closing Parentheses elements
Filter elements can be nested or grouped together using parentheses to form complex filter expressions.

For example:
```
\ps n  Find all
   nouns
And   which also
(     (begin with
   \lx [\#]ka  ka
   Or     or
   \lx an[\#]  end with an
)
```

This filter uses the Word Boundary built-in variable: [\#].

This filter would match the records of some of the derived nouns in a Tagalog database—e.g., katulong (helper), kasama (companion), kababayan (countryman), aklatan (library).

Each part of the expression is evaluated in the following order—internal groupings (within parentheses) first, and then from top to bottom.

Parentheses are added to the filter definition with the buttons provided in the Filter Properties dialog box.
Filtering (Database menu)
The Filtering command allows you to define and modify filters. Filters are special criteria which can temporarily narrow your focus to a particular set of records in a database. For example, in a lexical database, you could use filters to display only the records which are nouns.

Filters are a property of a particular database type.

To filter records
To turn on a filter, select it in the active filter combo box, which is located on the far right end of the toolbar. This box is used to turn on a filter or turn filtering off.
When a filter is turned on, it applies only to the current view. Other views of the same database are not affected.

To modify filters
From the Database menu, choose Filtering. This goes to the Filters Tab in the Database Type Properties dialog box.
You can also modify the properties of the currently active filter by right clicking on its name in the active filter combo box. This goes to the Filter Properties dialog box for the currently active filter.

Filters tab
This tab allows you to define and manage the filters which are available for the database type. The following options are available in the Filters tab:

Name / Condition list
Displays a list of all filters defined for the database type. Edit the list by using the buttons on the right.

Add button
Adds a new filter by accessing the Filter Properties dialog box.

Copy button
Copies a filter by accessing the Filter Properties dialog box.

Modify button
Modifies a filter by accessing the Filter Properties dialog box.

Delete button
Deletes the highlighted filter if not currently active. This button is not available when the selected filter is currently active.

Note: Choosing Cancel does not undo Add, Copy, Modify or Delete.

The name of the active filter is displayed in the Active Filter combo box on the far right side of the toolbar. When no filter is active, [no filter] is displayed:

When a filter is selected, right-clicking on the Active Filter combo box bypasses the Filters tab and automatically accesses the Filter Properties dialog box for easy modification to any of the properties.

Filters: Complex Expressions
Quite complex filters can be created by combining the primary filter elements using And, Or, Not, and the Opening and Closing parentheses. For example:

| ( | This expression matches records |
| \sc | in the scientific name field |
| Or | or |
The above filter uses the built-in Word Boundary variable, [#].

The elements needed to create such filters are discussed below:

The filter expression above looks like this:
\(\text{\textbackslash s c Or \textbackslash e e) And (\textbackslash ps n And (\textbackslash l x [\#]ka Or \textbackslash l x [...]an[#])}\)

Each part of the expression is evaluated in the following order—internal groupings (within parentheses) first, and then from top to bottom (or left to right, depending on your view).

---

**Word List (Tools menu)**

This command allows you to create a word list from specified fields within the files of a text corpus. To accomplish this action, Toolbox scans the chosen text corpus and produces a database containing an entry for each word, along with counts and some references for each.

The word list display is "static" derived information and should not be edited. When the contents of the corpus change you'll need to create an updated list. The rebuild process is relatively smart and quick, however. The program detects which files have changed and only updates from those changed files.

**To create a word list**

Do the following:

1. From the Tools menu, choose Word List. (You can use the shortcut, Alt+L.)
2. The Create Word List dialog box opens with the following options:
   - Text Corpus
     - Specifies the name of the collection or body of Standard Format files to be used with this command.
   - Edit button
Accesses the **Text Corpora** dialog box allowing you to add, copy or modify a corpus. Modify the **Text Corpus Properties** to specify the word markers, reference markers, file names, and language encoding.

- **Corpus Folder**¹
  A reminder of the default folder where the output file will be saved.

- **Output File**
  Specifies the name of the output file which will contain the word list created by this command.

- **Browse button**
  Opens the Select Output File dialog box, allowing you to select an existing filename.

- **Include**
  When enabled these check boxes allow you to include word counts and references in the word list.

- **# Refs. to Display**
  Allows you to display all references with each word in the word list, or to set a limit.

3. The word list file is automatically loaded into Toolbox.

   After a jump from a reference in a wordlist or concordance into text, Toolbox does a find to highlight the relevant word in the text.

   If multiple windows jump, the find is performed on the last window that jumps. To make a file the last one to jump, close it and open it again.

---

**Concordance (Tools menu)**

This command allows you to look up a word (or a phrase or a portion of a word) within the files of a text corpus. To accomplish this action, Toolbox scans the chosen text corpus and produces a database containing an entry for each occurrence of the "word", along with a location reference, text before and text after. The first time you ask for a concordance, the process may be relatively slow because it may be building an index. The concordance uses the same index as a word list built on the same text corpus. So if you have already built a word list, the concordance lookup should be quick.

The concordance lookup display is "static" derived information and should not be edited. When the contents of the corpus change you'll need to create an updated concordance lookup. The rebuild process is relatively smart and quick, however. The program detects which files have changed and only updates from those changed files.

**To look up a word**

Do the following:

1. From the Tools menu, choose Concordance.
2. The Concordance Lookup dialog box opens with the following options:

- **Text Corpus**
  Specifies the name of the collection or body of Standard Format files to be used with this command.

- **Edit button**

---

¹ The first file in the Text Corpus list establishes the **corpus folder**—the default folder where corpus-related files such as a word list and concordance lookups are saved.
Accesses the Text Corpora dialog box allowing you to add, copy or modify a corpus. Modify the Text Corpus Properties to specify the word markers, reference markers, file names, and language encoding.

- **Search For**
  Specifies the word to look up (or the phrase or portion of a word) to search for.

- **Match**
  Allows you to restrict the match to whole words or beginnings or ends of words. Choosing "Middle" will match any part of a word. For example, to list all words that begin with a, you can search for a and match "Start".

- **Match Case**
  When enabled, this option matches the exact characters in the search text making an upper/lower case distinction. This option allows the search text bad to match bad and not Bad.

- **# Matches to Display**
  Allows you to display all matches or to set a limit.

- **Corpus Folder**
  A reminder of the default folder where the output file will be saved. The first file in the Text Corpus list establishes the corpus folder—the default folder where corpus-related files such as a word list and concordance lookups are saved.

- **Output File**
  Specifies the name of the output file which will contain the concordance created by this command.

- **Browse button**
  Opens the Select Output File dialog box, allowing you to select an existing filename.

3. The concordance file is automatically loaded into Toolbox.

Limitations: Unlike the Find command, the Concordance command cannot search for built-in variables or special symbols.

**Text Corpora (Project menu)**

This feature of Toolbox allows you to specify a collection or body of Standard Format files to be used with the Word List and/or Concordance commands. These files might be stories, analyzed texts, dictionaries, etc. The only limitation is that they must contain similar standard format markers.

It is important to know which markers are used and which fields contain the data from which you will create a word list or do a concordance lookup.

**To specify a text corpus**

Do the following:

1. From the Project menu, choose Text Corpora.
2. The Text Corpora dialog box will appear, showing a list of named text corpora. The following options are offered:

   - **Close button**
     Closes this dialog box.

   - **Add button**
     Adds a new text corpus.

   - **Copy button**
     Copies an existing text corpus and its properties to a new name.
Modify button
Modifies an existing text corpus and its properties.

Delete button
Removes the highlighted text corpus from the list making it unavailable.

Settings Folder
A reminder of where the settings files are saved.

The Add, Copy and Modify buttons access the Text Corpus Properties dialog box.

The first file in the Text Corpus list establishes the corpus folder—the default folder where corpus-related files such as a word list and concordance lookups are saved.

Text Corpus Properties dialog box
Text Corpus dialog box (Text Corpora, Projects menu)
This dialog box allows you to define the properties of the text corpus which will be processed by the Word List and/or Concordance commands.

The following options are available:

Name
Specifies the name of the corpus that you want to create. The name might describe the texts in a way that would serve as a reminder of its function.

Language Encoding
The language encoding chosen here determines exactly which characters are considered to form words. The Word List and Concordance features assume that punctuation characters have been removed from the language’s default sort order.

Edit button
Accesses the Language Encoding Properties dialog box allowing you to add, copy or modify a language encoding.

Files to Process Edit Files List button
Accesses the Select Files dialog box allowing you to specify the files which are included in the corpus.

Markers for Words to Process
Specifies the fields from which the word list and concordance will be formed. This option defaults to a standard set of markers. For example, for a text such as:

```
\id wildboar
\ref wildboar 001
\t hiitaiAiri kitairiki isaiki aNtamiki. ti haniAi apaani apaani.
```

You would use:
Markers for Words to Process \t

This corresponds to the standard markers inserted by Break and Number Text feature.

Reference Markers
Specifies the markers to use as locators that "point" to the word within the text(s). The reference markers specify where Toolbox collects location reference information for each word. This can be done in several ways. Specify only the reference markers you need. For example, if your texts have only a name reference in each file or section:

```
\id wildboar
\t hiitaiAiri kitairiki isaiki aNtamiki. ti haniAi apaani apaani.
```

Specify your reference markers like this:
Primary (textual ref) \id
Secondary (numeric ref)
Tertiary (numeric ref)
If your texts have names and sentence numbers like this:
\name wildboar
\ref 001
\t hiitaiAiri kitairiki isaiki aNtamiki.
\ref 002
\t ti haniiAi apaani apaani.
Specify your reference markers like this:
Primary (textual ref) \name
Secondary (numeric ref) \ref
Tertiary (numeric ref)
If your texts have names and sentence numbers together like this:
\ref Gahp.001
\t Nengmondi taiduwa numia.
\ref Gahip.002
\t Baing bai bala hainino, ba utau ngaxang bu ngala ngautu gahibigua.
Specify your reference markers like this:
Primary (textual ref) \ref
Secondary (numeric ref)
Tertiary (numeric ref)
Primary reference items may be alphabetic or numeric. The secondary and tertiary reference items must be numeric, as illustrated above.
Limits:
32750 different primary reference items
Numeric range 0-65500 for secondary references
Numeric range 0-255 for tertiary references In a primary reference field such as
\id wild boar
Toolbox uses only the first item or "word" in the field as the primary reference. If you want a multi-word phrase as a primary reference, remove the spaces between the words ("WildBoar") or join the words with underscores or other punctuation ("wild_boar").

Select Files dialog box (Text Corpora)
Text Corpus Properties dialog box, Text Corpus dialog box, (Text Corpora, Projects menu)
This dialog box is used to select the files to be included in a text corpus. The following options are available:
Folder
Provides a list of all folders in the current folder or drive. Double-click on the folder which contains the file you wish to select.
Up button
Moves up one level to the next folder (directory).
Drive
This drop-down list box located beneath the list of folders specifies the disk drive.
Available Files
Displays the files within the current folder. To move the files between this box and the Selected Files box, select the file and then choose the appropriate button, or double-click the filename to insert it next in the list.

- **First ->** Places selected file first in Selected Files list
- **Last ->** Places selected file last in Selected Files list
- **Insert ->** Places selected file next in Selected Files list
- **< Remove** Removes selected file from Selected Files list
- **< Clear** Clears all files from Selected Files list

Selected Files
Displays the list of files which make up the text corpus. Double-clicking on a file will remove it from the list. Note that the first file in the list establishes the corpus folder—the default folder where corpus-related files such as a word list and concordance lookups are saved.

Show Full Path
Changes the list display to include the entire path—the drive and directories. For example, instead of LEX.DB, the display might be C:\Toolbox\Lexicon\LEX.DB.

**Compare Files (Tools menu)**

Compare Files shows the differences between two files. Comparing a file with a backup version shows all the changes since the backup version was saved. Comparing two files that started the same, but were edited by different people shows the changes both people have made.

Compare Files requires that the two files to be compared both be opened in Toolbox and sorted the same. If you want to compare a file with a previous version, use File, Open to open the previous version.

To prepare for comparing files, arrange the windows of the two files side by side so that they do not overlap. You can use Windows, Tile Side by Side for a vertical arrangement) or Windows, Tile (for a horizontal arrangement), depending on the data you are comparing.

Put the focus on the window of the current version of the file. Go to the first record. Choose Tools, Compare Files (or press Ctrl+K).

You will see the first difference highlighted in the two windows. If the difference is a change, you will see the changed fields, words or letters highlighted in both windows. If the difference is an insertion or deletion, you will see the extra fields, words or letters highlighted in one window, and nothing highlighted in the other window.

To move to the next difference, press Ctrl+K again.

Each time you press Ctrl+K, a new difference will be highlighted. When there are no more differences, you will see a dialog box that says "No more differences found".

If whole fields are added or removed, things can get out of step. Just place the cursor at the beginning of a comparable field in each window and press Ctrl+K.

Toolbox compares markers, too. If you have changed a marker, even though the data is the same, Toolbox will show the whole fields as different.
In some cases, if the differences are large, or if a record is added or missing, or if you changed the lexeme or homonym numbers (and thus changed the sorting), you may find that the two records being compared are not the same record. If this happens, move the current version window to the next record and press Ctrl+K again. In most cases, this will cause the two windows to show the same record. If this doesn't work, you can move both windows forward until they show the same record.

The first time Compare Files is chosen, Toolbox looks for a candidate window for comparison. To be used for comparison, a window must be of the same database type as the current window, and must be sorted by the same fields as the current window.

These criteria should almost always find the correct window for comparison. If no window fits the criteria, then a message is given which lists the criteria. If more than one window fits the criteria, then the most recently opened one will be chosen.

In most cases, you will have recently opened a file for comparison, so Toolbox will find the correct window. If comparison begins on the wrong window, then close the incorrect window or sort it by some other field, or close and reopen the desired comparison file to make it the most recently opened window.

If Toolbox does not find a window that you think should be a candidate, check the sorting (Database, Sorting) to be sure the sort fields are all the same. One way they can be different is that clicking on the header of a browse view keeps the previous sort field as a secondary sort field. But opening a new file uses only the record marker as a sort field – or the \lx and \hm in the case of MDF dictionaries.

Hide Fields (View menu)

The Hide Fields command turns on and off the hiding of fields. The data is not discarded, it is just removed from view. This allows you to focus on particular fields without irrelevant “clutter”.

The list of fields to hide is specified in the Fields to Hide dialog box.

To enable or disable the hiding of fields

From the View menu, choose Hide Fields. When the Hide Fields option has a check mark, it means that fields are hidden. If there is no check mark, no fields are hidden.

You can also use the shortcut keys, CTRL+M.

Hidden fields will not be displayed in normal view. But they can be displayed in browse view.

The record marker field and the primary sort field cannot be hidden.

Interlinear fields may be hidden, but will unhide when an interlinearize or spell check command is performed.

Hidden field selection applies to all databases of the same database type.

Fields to Hide (View menu)

This command is used to change the fields which are hidden.

To change the display:

1. Choose Fields to Hide in the View menu. You will see a dialog box that is similar to the Browse Fields dialog box.
2. Use the dialog box to move field markers from the Available Fields box on the left to the Hide Fields box on the right. The Available Fields markers which appear are the ones inventoried in the Markers Tab in the Database Type Properties dialog box. To move the field markers between the boxes, select the marker and then choose the appropriate button, or double-click on the marker to move it to the other list.
Hidden fields will not be displayed in normal view. But they can be displayed in browse view.
The record marker field and the primary sort field cannot be hidden.
Interlinear fields may be hidden, but will unhide when an interlinearize or spell check command is performed.
Hidden field selection applies to all databases of the same database type.

<table>
<thead>
<tr>
<th>Insert ➔</th>
<th>Places selected Available Fields marker next in Hide Fields list</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;- Remove</td>
<td>Removes selected marker from Hide Fields list</td>
</tr>
<tr>
<td>&lt;- Clear</td>
<td>Clears all markers from Hide Fields list</td>
</tr>
</tbody>
</table>

**To hide and un-hide fields**

After specifying which fields to hide, the option "Hide Fields" on the View menu will be checked. To unhide the fields, simply click on Hide Fields on the View menu and the check mark will be removed.
To hide the same set again, choose Hide Fields and the option will be checked and the fields hidden.

**Parallel Movement**

If two (or more) windows are *sorted by the same language* and the second is a Parallel Movement Target, the second will move from record to record in step with the first. Parallel Movement can cause number of windows which are sorted the same to stay in step with one another. Parallel movement happens whether you are doing Search, Next Record, or any other way of changing to another record. A successful find triggers parallel move of other windows.

**Exceptions**

There are a couple of exceptions to parallel movement.

If a window is maximized, other windows will not move parallel to it.

Another exception is that a duplicate window on the same database, sorted the same, will not move in step with its twin. However, a duplicate window sorted differently will move. For example, if you make a duplicate window of a word list and sort the duplicate from the end, then the two windows will move together so that you can see both the words that start in a similar way and those that end in a similar way. This can be very useful for analyzing the morphology of a language.

To turn off this feature for a window, turn off the Parallel Movement Target option in the View menu for the window that you do not want to do parallel move when other windows move.

This is similar to the multiple window Jump. If you do a Jump and there is no Jump Path, then all windows which are sorted by the same language will jump.

**External Parallel Movement**

Some programs are set up to send and receive messages that cause them to move in parallel with other programs. In particular, Paratext and Translator's Workplace can receive references from Toolbox. If one or both of those programs are running at the same time as Toolbox, then when Toolbox moves to a new reference, those
programs will move their open windows to the same reference. In a similar way, Paratext can send references to Toolbox.

Toolbox only sends out references if the current window is sorted by a field whose language encoding is named “Book References”.

This feature works best if the reference system used has a dot between the book and chapter. If there is a space between the book and chapter, then Toolbox can send references, but it cannot respond to references from Paratext. Note also that the book abbreviations must match the three-letter abbreviations used by Paratext and Translator’s Workplace.

This feature can also be used to cause multiple instances of Toolbox to move in parallel.

**Match Characters**

Match characters are controls which specify the way Toolbox determines a match. For the options below, consider an example with primary groupings in the sort order of:

A a á à  
B b  
D d

**By primary grouping only**

- Allows any characters grouped together in the current sort order to be considered identical. Using the example above, this option would allow the search text bad to match bad, bád, bàd and Bad.

**Disregarding case**

- Matches the exact characters in the search text, but does not make an upper/lower case distinction. Using the example above, this option would allow the search text bad to match bad and Bad, not bád or bàd.

**Exactly by secondary ordering**

- Matches the exact characters in the search text, distinguishing upper/lower case as well. Using the example above, this option would allow the search text bad to only match bad.

**Even those normally ignored**

- Matches the exact characters in the search text, distinguishing ignore characters defined in the sort order. For example, consider the hyphen defined as an ignore character and the following records:

  labas outside  
  lab-as to sell fish  
  lab-og to wallow  
  labog to discard

Using this option, labas, would match only labas—the other options would also match lab-as; a search for lab- would match lab-as and lab-og—the other options would also match labas and labog.

**Sorting: An Overview**

Toolbox is able to sort data in a variety of ways. Toolbox can sort the records in a database by different fields. Toolbox also allows you to establish the proper order for sorting data for each language. It also offers the option of sorting the same field / language in different ways – from the end of the word, as well as by generalizations such as Consonant vs Vowel. The default phonetic sort order (also used as the default vernacular sort order) has the option of sorting “alphabetically” or by place of articulation or manner of articulation. Non-roman scripts and syllable-based sorting have also been accommodated.
Sorting by various fields and specifying the details of the sort order are two quite different topics. Typically the sort order for a language is set up once and then is not modified. Changing the fields or direction of sorting is done frequently, sometimes many times in the same day.

Specifying the details of the sort order is discussed under Advanced Topics in the Language Encodings section. Changing the fields or direction of sorting is discussed below.

**Sorting by Different Fields**

Toolbox initially uses the record marker field to order (or index) each record in your database. The purpose of Sorting is to allow you to organize data around another element in the record that is more important to the task at hand—for example, you might want to sort the records by the date field in order to see entries you haven’t edited recently. This is accomplished by changing the primary sort field in the database.

**Two ways to change the primary sort field**

4. Click on the marker at the top of the column in Browse view. The marker that was previously the primary sort field becomes a secondary sort field.

OR

5. From the Database menu, choose Sorting. The Sorting by Fields dialog box appears with the following options:

### Sorting Options

**Fields for Sorting**

Toolbox also allows you to sort on more than one field. When the primary sort field contains many elements that are identical—such as ps, part of speech—an additional sort field can help— for example by sorting each part of speech by the lexeme (so all the verbs will have the lexemes in alphabetical order, as will all the nouns, and all the adjectives, etc.)

Markers can be moved from the Available Fields box on the left to the Sort Fields box on the right. The markers which appear are all those inventoried in the Markers tab in the Database Type Properties dialog box. To move the field markers between the boxes, select the marker and then choose the appropriate button, or double-click on the marker to place it next in the list. The marker which is first in the list becomes the primary sort field.

- **First**
  - Places selected marker first in Sort Fields list
- **Last**
  - Place selected marker last in Sort Fields list
- **Insert**
  - Places selected marker next in Sort Fields list
- **Remove**
  - Removes selected marker from Sort Fields list
- **Clear**
  - Clears all markers from Sort Fields list

**Note:** When a new record is inserted it is displayed following any (and all) other records with which it sorts identically. For example, if you are sorting by part of speech only and add a new noun it will appear at the end of the set of nouns. If you want it to be shown in alphabetical order relative to other nouns, you must include the record marker field second in the sorting.

**Sort order for first field**

Toolbox uses the Sort Order defined for the language associated with the marker. If more than one sort order is defined for a language, this combo box will provide a list. Select the one you wish to use.
Sort first field from end
This check box allows you to choose to sort “backwards”. This allows you to examine suffixes, for example. Date fields should be sorted from the end in order to have a chronological order (grouping first by year, then month, then day).
Any marker can be sorted in this way.
To get a good overview of word patterns use the Browse view to display a list of the words in the database.
This works for both left-to-right and right-to-left scripts, “end of word” being defined as the last part spoken.
Note: This is not the appropriate option to use if you have a right-to-left script and you are trying to make it sort properly from the beginning of the word. If your script is not sorting properly, examine the Language Encoding Properties to be sure that the Right to Left option has been chosen under Advanced Options.

Sorting by a field which occurs multiple times in a record
In many records there may be multiple occurrences of a particular field. In order to see all the occurrences of this field in a Browse View, it is necessary to sort the database by that field marker.
As you move through the database, you will now encounter the same record multiple times, once for each occurrence of the sort field in the record. So if a record has

\lx gato
\ge cat
\ge tiger
\ge lion

and if you sort by the \ge field, you will see the record appear three times as you move through the database: once among the other “C” glosses, for “cat”; once among the other “L” glosses, for “lion”; and once among the other “T” glosses, for “tiger”. This does not mean that there are now three records, but only that there are three places you can find it if you sort by gloss. If you edit, any changes will appear in all three.

Note: if you delete one of the records, you delete the only record you have. These are not duplicates! These are different views of the same record. You can delete one of the sort fields – then you will only see the record twice. But if you delete the whole record, “all three” will disappear, since there really is only one record.

Note: When moving to a different record that the cursor is always initially placed in the primary sort field.

Sorting, Browse, and Hierarchy
In a lexical database where a single record might contain multiple occurrences of a field, each with data logically associated with the different occurrences (as in example below), sorting on \ps could leave you with confusing results.
Consider the record below from a lexical database without Marker Hierarchy defined:

\lx escape
\ps vi
\ge to get away
If you set the primary sort field for \textit{ge}, and \textbf{Browse} on \textit{ge}, \textit{lx} and \textit{ps}, the Browse View will display the following:

- avoidance of reality: escape vi
- providing an escape: escape vi
- to elude: escape vi
- to get away: escape vi

This is because, without the hierarchy, Toolbox looks for the first \textit{ps} in the record. However, if the hierarchy is created, as in the following:

\begin{verbatim}
\textit{lx} escape
\textit{ps} vi
\textit{ge} to get away
\textit{ps} vt
\textit{ge} to elude
\textit{ps} n
\textit{ge} avoidance of reality
\textit{ps} adj
\textit{ge} providing an escape
\end{verbatim}

Then the Browse of the \textit{ge} sort will display this:

- avoidance of reality: escape n
- providing an escape: escape adj
- to elude: escape vt
- to get away: escape vi

\textbf{Note}: If a dictionary is specified as an MDF dictionary, and is using the standard MDF markers, the hierarchy relationships for those markers are already defined.

**The Sort Field is displayed on the Status Bar**

You will notice that when you change the primary sort field to anything other than the record marker field, the primary sort field will appear on the \textit{status bar} (at the bottom) along with the data that is currently in focus. Thus, the single-record database above sorted by the lexeme will show just the lexeme:

\begin{verbatim}
\textit{lx} escape 1/1
\end{verbatim}

while sorting on the \textit{ge} the status bar will show the gloss field as well as the lexeme:

\begin{verbatim}
\textit{lx} escape \textit{gl} to elude 3/4
\end{verbatim}

The count of the occurrences of the primary sort field is also displayed on the status bar. In the one-record database example above, sorting on the \textit{lx} would display a count of 1, while sorting on the \textit{ge} would display a count of 4! It is essential to understand that the status bar displays the number of
occurrences of the primary sort field. If the database is sorted on the record marker field, then the number displayed will also be the number of actual records in your database.

**Sorting and duplicate windows on the same database**

Toolbox can show two or more views of the same database in different windows and sort their contents differently.

**To display duplicate windows:**

Do the following:

1. From the Window menu, choose Duplicate. A new window will open containing the active database. On the title bar, the database name will appear followed by a colon and the number of windows currently open on the database. For example, MDFSAMP:2.
2. From the Window menu, choose Tile Side by Side. This is a good layout for viewing multiple records.

It is important to remember that the duplicate command does not open a “copy” of the database, but instead opens another window on the database. As a result, any edits (changes) made in one window will be reflected in both windows.

**To sort a duplicate window by another marker**

Do the following:

1. Click somewhere in the duplicate window.
2. From the Database menu, choose Sorting
3. Choose Clear.
4. Select the marker(s) you wish to sort by from the Available Markers list and move it into the Sort Fields list. (Double clicking a marker in the Available Markers list moves it next in the Sort Fields list.)

The actions above change the first field by which the database is sorted to the selected marker. In other words, this field has now become the primary sort field. This change is reflected on the status bar, which shows the count of the occurrences of the primary sort field (not the number of records).
Contents

MERGE ............................................................................................................................. 153
To merge two databases: .................................................................................................. 153

Some notes: ....................................................................................................................... 153

How the merge is done .................................................................................................... 153

Merge if no match ........................................................................................................... 153
  Example (Merge, no match) .......................................................................................... 153

Merge if identical ............................................................................................................. 154
  Example (Merge, identical) .......................................................................................... 154

Merging in new fields .................................................................................................... 154
  Example (Merge, new fields) ........................................................................................ 154

Merging modified fields .................................................................................................. 155
  Example (Merge, modified fields) ................................................................................ 155

Merging if target fields are empty ................................................................................. 155
  Example (Merge, empty fields) ..................................................................................... 156

Merging if fields were deleted ....................................................................................... 156
  Example (Merge, deleted field) ..................................................................................... 156

Merging to multiples ...................................................................................................... 157
  Example (Merge, Merge-To Multiple) ......................................................................... 157

Merging same homonym numbers .................................................................................. 157
  Example (Merge same homonym) ................................................................................. 157

Merging with added homonym numbers ....................................................................... 158
  Example (Merge, added homonym number) .................................................................. 158

Merging with different homonym numbers .................................................................. 159
  Example (Merge, new homonym) ................................................................................ 159

Merging from multiples ................................................................................................... 159
  Example (Merge, Merge-From multiple) ...................................................................... 159

Preventing modification of certain fields while Merging .................................................. 160

Protecting existing data ................................................................................................. 160

Preventing modification of existing records .................................................................. 160

Preventing Adding new information to a lexicon .......................................................... 160

Discerning the source of data in the resulting file .......................................................... 160

Inserted records ............................................................................................................. 160

Inserted or modified fields ............................................................................................ 161

Using file compare to see the results of a merge .......................................................... 161

Using merge to insert a new field into every record ....................................................... 161
This command is used to combine an existing database with the current or active database. 

NOTE: In this section, the term “key field” was used to refer to the record marker field.

**To merge two databases:**

From the Database menu, choose Merge Database. The Select Database dialog box asks you to choose the Database from which to merge records and fields. You will be presented with a list of the open databases in the current project which have the same database type as the active database.

The Show Path option in the dialog box changes the list of files to include the entire path—the drive and folders (directories). For example, instead of LEX.DB, the display might be C:\Toolbox\Lexicon\LEX.DB.

**Some notes:**
- Only databases assigned to the same database type can be merged.
- There is no need for a “common ancestor”. Any two databases (of the same type) can be merged.
- You can even start with an “empty” database, and merge into it. Full duplicates from the Merge-From database will be discarded (keeping only one) and records with the same key fields will be merged together into a single record.

**How the merge is done**

**Merge if no match**

If a record in the Merge-From database finds no corresponding record in the Merge-To database, then the record from the Merge-From database will be inserted into the Merge-To database. This is the only circumstance in which a whole record is inserted.

**Example (Merge, no match)**

**Original Merge-To Database**

```
\lx record 10
\ge gloss 10

\lx record 20
\ge gloss 20
```

**-----------------------------**

**Merge-From Database**

```
\lx record 15 (new)
\ge gloss15
```

**-----------------------------**

**Final Merge-To Database**

```
\lx record 10
\ge gloss 10

\lx record 15 (new)
\ge gloss15

\lx record 20
\ge gloss 20
```
**Merge if identical**
If records are completely identical in contents, they will not be duplicated in the Merge-To database. In the comparison for identical contents, the date is ignored.

**Example (Merge, identical)**

**Original Merge-To Database**
```
\lx record 1
\ge gloss 1

\lx record 2
\ge gloss 2
```

**Merge-From Database**
```
\lx record 1
\ge gloss 1

\lx record 2
\ge gloss 2
```

**Final Merge-To Database**
```
\lx record 1
\ge gloss 1

\lx record 2
\ge gloss 2
```

**Merging in new fields**
With identical key fields, if some new fields have been added to the Merge-From database, the new fields will be added in approximately the same position into the Merge-To database.

**Example (Merge, new fields)**

**Original Merge-To Database**
```
\lx record 1
\ge gloss 1
\nt another field 1

\lx record 2
\ge gloss 2
```

**Merge-From Database**
```
\lx record 1
\ge gloss 1
\gn new field 1

\lx record 2
\ge gloss 2
\gn new field 2
```
Final Merge-To Database
\lx record 1
\ge gloss 1
\gn new field 1
\nt another field 1

\lx record 2
\ge gloss 2
\gn new field 2

Merging modified fields
With identical key fields, if some fields have been modified, the fields in the Merge-From database will be added into the Merge-To database, above their corresponding fields. You can manually delete the unwanted version.

Example (Merge, modified fields)

Original Merge-To Database
\lx record 1
\ge gloss 1

\lx record 2
\ge gloss 2

-------------------------------------------

Merge-From Database
\lx record 1
\ge modified gloss 1

\lx record 2
\ge modified gloss 2

Final Merge-To Database
\lx record 1
\ge modified gloss 1
\ge gloss 1

\lx record 2
\ge modified gloss 2
\ge gloss 2

Merging if target fields are empty
With identical key fields, if there are empty fields in the Merge-To database, and corresponding fields have contents in the Merge-From database, then the empty fields in the Merge-To database will receive the contents of the Merge-From fields. This is one way to ensure new data is placed correctly.
Example (Merge, empty fields)

Original Merge-To Database
```
\lx lexeme
\sn 1
\ge gloss 1
\gn national gloss 1
\xv
\sn 2
\ge gloss 2
\xv
\nt note
```

Merge-From Database
```
\lx lexeme
\sn 1
\ge gloss 1
\xv example 1
\sn 2
\ge gloss 2
\xv example 2
```

Final Merge-To Database
```
\lx lexeme
\sn 1
\ge gloss 1
\gn national gloss 1
\xv example 1
\sn 2
\ge gloss 2
\xv example 2
\nt note
```

Merging if fields were deleted

With identical key fields, if fields were deleted in the Merge-From database, there will be no indication that anything was changed in the Merge-To database. This assures that merge will never lose information that was in the Merge-To database.

Example (Merge, deleted field)

Original Merge-To Database
```
\lx record 1
\ge gloss 1a
\ge gloss 1b
```

Merge-From Database
```
\lx record 1
```
Merging to multiples
With identical key fields, if there are multiple records in the Merge-To database with the same key field (i.e., homonyms), Toolbox will merge the From data into the record with the most similar fields.

Example (Merge, Merge-To Multiple)

Original Merge-To Database
\lx record
\ge gloss 1

\lx record
\ge gloss 2

-------------------------------------------

Merge-From Database
\lx record
\ge gloss 2
\xv example 2

-------------------------------------------

Final Merge-To Database
\lx record
\ge gloss 1

\lx record
\ge gloss 2
\xv example 2

Merging same homonym numbers
If there is a record in the Merge-To database with the same key and the same homonym number, the two records will be merged.

Example (Merge same homonyms)

Original Merge-To Database
\lx record
\hm 1
\ge gloss 1

\lx record
\hm 2
\ge gloss 2

-------------------------------------------

Merge-From Database
\lx record
\hm 2
\ge gloss 2
Merging with added homonym numbers

If there is a record in the Merge-To database with the same key and no homonym number, the two records will be merged. This assumes that the record was edited to add a homonym number.

Example (Merge, added homonym number)

Original Merge-To Database
\lx record
\hm 1
\ge gloss 1

---

Merge-From Database
\lx record
\hm 1
\ge gloss 1

\lx record
\hm 2
\ge gloss 2
\xv example 2

---

Final Merge-To Database
\lx record
\hm 1
\ge gloss 1

\lx record
\hm 2
\ge gloss 2
\xv example 2
Merging with different homonym numbers
If all the records in the Merge-To database that have the same key have different homonym numbers, the record will be inserted whole, not merged into any other record.

Example (Merge, new homonym)
Original Merge-To Database
```
\lx record
\hm 1
\ge gloss 1
\lx record
\hm 2
\ge gloss 2
```
-----------------------------

Merge-From Database
```
\lx record
\hm 3
\ge gloss 3
\xv example 3
```
========================

Final Merge-To Database
```
\lx record
\hm 1
\ge gloss 1
\lx record
\hm 2
\ge gloss 2
\xv example 2
\lx record
\hm 3
\ge gloss 3
\xv example 3
```

Merging from multiples
With identical key fields, if there are multiple records in the Merge-From database with the same key field and no homonym numbers, but only one in the Merge-To database, the fields from the records of the Merge-From database will all be merged into the same record in the Merge-To database. If you want separate entries, either create them after the merge or temporarily modify the lexeme field of the Merge-From data before merging.

Example (Merge, Merge-From multiple)
Original Merge-To Database
```
\lx record
\ge gloss 1
```
Preventing modification of certain fields while Merging

Protecting existing data
If there are fields in the Merge-To database which are not to be modified but which may have been modified accidentally in the Merge-From database, then Hide the fields. (Hide Fields will cause the fields in both database to be hidden, since they are the same database type.) Toolbox will not merge hidden fields. Hidden fields will not be brought in or modified.

Preventing modification of existing records
If person B is officially just making new entries, but might have modified existing entries, you can prevent modification of existing data by hiding all fields except the lexeme. If this is done, then new records will be added, but existing records will not be modified.

Preventing Adding new information to a lexicon
If person B adds new fields, such as dialect variants or scientific names, you can hide all fields except the new fields to be added. This will assure that no other modifications are made during the merge.

Discerning the source of data in the resulting file

Inserted records
One way to be able to identify newly inserted records is to put a source field in those records before they are merged.

The best way to add a source field is for each person working on the dictionary to use a personalized template that includes a source (so) field with their initials. If this is done, then after the merge, you can filter for a combination of that source and a date since the last time their database was merged.

Another way to identify newly inserted records is to have a special master database tag in every record of the Merge-To database. This could be a special value in a source field or a note field. Since the records
in the Merge-From database won't have that tag, it is easy to filter for them. After they have been reviewed and accepted, the master database tag field can be added to them by pasting in the field.

**Inserted or modified fields**
The date field of inserted records is left as it was in the original Merge-From database, but the date field of any record that has a new field added is changed to the current date. So filtering on the current date shows the modified records (if nothing else had today's date).

If the Merge-From file is still loaded, then parallel movement will let you see readily what came from the Merge-From.

If a field is added, you can see it as new in the Merge-To database and you can review it for correctness. If it is not in the correct place, you can move it.

If a field is changed, the record will contain both the original Merge-From field, and the modified Merge-To field, with the Merge-From field first. You can readily delete the unwanted version of the field.

**Using file compare to see the results of a merge**
Another way to see the results of a merge is to use file compare. Before you do the merge, make a copy of the original Merge-To file in a different folder or with a different name. After the merge is done, open the copy of the original Merge-To file, and place its window beside the new Merge-To file. (You can use Window, Tile Side-by-Side. The file in focus will be placed on the left.) Go to the first record of the new Merge-To file and choose Tools, Compare files. You will see the first newly added record or field highlighted in the Merge-To file. If a field has been inserted or modified, you will see the original record in the window of the original Merge-To file. After reviewing the newly inserted field, and making any desired adjustments, you can use Ctrl+K to move to the next difference.

If multiple records have been added in the same place, compare may not be able to find the next matching record. In that case, move forward in the new Merge-To database until you come to a record that is also in the original database. Then continue the compare, by pressing Ctrl+K.

**Using merge to insert a new field into every record**
A special case of Merge can be used to insert one or more new fields into every record in a database. To do this, make a database that contains only one record, and leave the key field empty. If such a database is the entire Merge-From file, then the single record with empty key is merged into every record of the Merge-To file. The result is that if the single record contains one or more fields besides the empty key, those fields will be added to every record in the Merge-To file that does not already contain them. The placement of added fields follows the principles above. For example, to cause an added field to be placed after an existing field, include the existing field in the Merge-From record.

**More Merge Examples**

**Example (Merge new homonym)**

**Original Merge-To Database**
```
\lx record
\hm 1
\ge gloss 1
\lx record
\hm 2
\ge gloss 2
```

**Merge-From Database**
```
\lx record
```
Final Merge-To Database
\lx record
\hm 1
\ge gloss 1
\lx record
\hm 2
\ge gloss 2
\xv example 2
\lx record
\hm 3
\ge gloss 3
\xv example 3

Example (Merge, add homonym number)

Original Merge-To Database
\lx record
\ge gloss 1

Merge-From Database
\lx record
\hm 1
\ge gloss 1
\lx record
\hm 2
\ge gloss 2
\xv example 2

Final Merge-To Database
\lx record
\hm 1
\ge gloss 1
\lx record
\hm 2
\ge gloss 2
\xv example 2

Example (Merge, deleted field)

Original Merge-To Database
\lx record 1
\ge gloss 1a
Example (Merge, empty fields)

Original Merge-To Database
\lx lexeme
\sn 1
\ge gloss 1
\gn national gloss 1
\xv
\sn 2
\ge gloss 2
\xv
\nt note

Merge-From Database
\lx lexeme
\sn 1
\ge gloss 1
\nxv example 1
\sn 2
\ge gloss 2
\xv example 2

Final Merge-To Database
\lx lexeme
\sn 1
\ge gloss 1
\gn national gloss 1
\xv example 1
\sn 2
\ge gloss 2
\xv example 2
\nt note

Example (Merge, identical)

Original Merge-To Database
\lx record 1
\text\ge\ gloss\ 1
\text\lx\ record\ 2
\text\ge\ gloss\ 2
-------------------------------------------

**Merge-From Database**
\text\lx\ record\ 1
\text\ge\ gloss\ 1
\text\lx\ record\ 2
\text\ge\ gloss\ 2
======================================

**Final Merge-To Database**
\text\lx\ record\ 1
\text\ge\ gloss\ 1
\text\lx\ record\ 2
\text\ge\ gloss\ 2

---

**Example (Merge, Merge-From multiple)**

**Original Merge-To Database**
\text\lx\ record
\text\ge\ gloss\ 1
-------------------------------------------

**Merge-From Database**
\text\lx\ record
\text\ge\ gloss\ 1
\text\lx\ record
\text\ge\ gloss\ 2
\text\lx\ record
\text\ge\ gloss\ 3
======================================

**Final Merge-To Database**
\text\lx\ record
\text\ge\ gloss\ 3
\text\ge\ gloss\ 2
\text\ge\ gloss\ 1

---

**Example (Merge, Merge-To homonyms)**

**Original Merge-To Database**
\text\lx\ record
\text\hm\ 1
\text\ge\ gloss\ 1
\text\lx\ record

---
Merge - From Database
\lx record
\hm 2
\ge gloss 2
\xv example 2

Final Merge - To Database
\lx record
\hm 2
\ge gloss 2
\xv example 2

Example (Merge, Merge-To Multiple)

Original Merge-To Database
\lx record
\ge gloss 1
\lx record
\ge gloss 2

Merge - From Database
\lx record
\ge gloss 2
\xv example 2

Final Merge - To Database
\lx record
\ge gloss 1
\lx record
\ge gloss 2
\xv example 2

Example (Merge, modified fields)

Original Merge-To Database
\lx record 1
\ge gloss 1
\lx record 2
\ge gloss 2
Merge-From Database
\|x record 1
\ge modified gloss 1
\|x record 2
\ge modified gloss 2

Final Merge-To Database
\|x record 1
\ge modified gloss 1
\ge gloss 1
\|x record 2
\ge modified gloss 2
\ge gloss 2

Example (Merge, new fields)

Original Merge-To Database
\|x record 1
\ge gloss 1
\nt another field 1
\|x record 2
\ge gloss 2

Merge-From Database
\|x record 1
\ge gloss 1
\gn new field 1
\|x record 2
\ge gloss 2
\gn new field 2

Final Merge-To Database
\|x record 1
\ge gloss 1
\gn new field 1
\nt another field 1
\|x record 2
\ge gloss 2
\gn new field 2

Example (Merge, no match)
Exporting

Export: An Overview (File menu)

You can export a Toolbox database to other programs for different kinds of linguistic analysis or to Microsoft Word for printing. The database type file contains the export processes that have been established for all databases of that type (for example, anthropology notes, interlinear text, lexicon). You can define multiple named export processes for specific purposes.

For example: When you print a draft of a lexical database to check for consistency and completeness, you may print it on full pages; when you print a copy for publication, you may exclude some data fields and print it in booklet form using a custom paper size. By making two export processes, you let Toolbox remember the formatting differences for both approaches.

Besides offering the option to include or exclude fields, all export processes except MDF offer the possibility of specifying a custom CC table. Export processes can be chained together to process the data through multiple CC tables. (CC stands for Consistent Changes. A "table" is a set of changes that manipulate the data in various ways.)

CAUTION: Consider carefully whether it is appropriate to edit an exported file directly (for example, to correct a misspelling). If you consider the database "primary" and the exported file strictly "secondary," you should instead edit the database and export it again. This is one of the reasons why Toolbox has export processes.

TIP: Use the processes with the general names "Standard Format" and "Rich Text Format" for one-time tasks. Anything you will do more than once you probably want to name and save for future use.
If you only want to print a limited amount of data quickly, just as it appears on the Toolbox screen, you could use the Print command on the File menu instead. This is particularly useful for printing the Browse view. However, the Print command has a problem with blank lines in the data and one causes the rest of the page to be overstruck. This is a Windows error, not a Toolbox error.

To export, place your cursor in the file you wish to export. This will select the proper database type. (That is, you can't export your lexicon if your cursor is in your interlinear text file.) Then do File, Export. The Export shortcut key is CTRL+X.

The Export dialog box offers the following options:

**Export process**
Select an export process from the list and choose OK to export or Modify to change the options for that process.

To the right of the list are buttons that let you manage the list of processes:
- **Add**
  Accesses the Export Process Type dialog box so that you can select the type of a new export process:
  - Standard Format (SF)
  - Rich Text Format (RTF)
  - Extensible Markup Language (XML)
  - Multi-Dictionary Formatter (MDF)
- **Copy**
- **Modify**
- **Delete**

**Records to include**
All but the MDF export process offer the following options shown below the list of export processes:

**Entire file**
Export every record in the database, sorted in the basic record marker sort order. Ignores the current window's sorting and filtering properties.

**Current window's records**
Applies the current window's sorting and filtering properties. If the primary sorting field occurs multiple times in a record, that record is exported multiple times. If the primary sorting marker isn't the record marker, the Export Current Window dialog box lets you choose whether the relevant sorting field should be moved to the top of the exported record. **Current record only**

Note: MDF provides the option to export filtered data as part of its list of features. Click on Modify to locate this feature in MDF.

**Show properties dialog box on Export**
When checked, the dialog box for the selected export process will appear when you choose OK. If you modify any properties, they will affect this exported file and will also be remembered for future exports.

**Automatically open document in word processor**
When checked, Toolbox will cause the application associated with RTF files (usually Microsoft Word) to open files exported using RTF and MDF processes (on Windows systems only). If the Confirm Conversions option is enabled in Word, you will see a Convert File dialog box when the exported RTF file is opened.
TIP: If an exported RTF file is already open in Word, remember to close it before exporting again (for example, if you find a misspelling in the document and correct it in the Toolbox database). Otherwise Toolbox will not be able to close the file (the file which is open in Word) and the export will abort.

Database Type
All databases of the same type share the same set of export processes.

**Export Process Type dialog box**

When you choose Add in the Export dialog box, this dialog box appears so that you can select the type of process:

![Export Process Type dialog box]

- **Standard Format (SF)**
  A Standard Format file is like a Toolbox database, except that it lacks the "header" field. Use an SF Export Process to make systematic changes to a Toolbox database or to reorganize the data for a different analysis task.

- **Rich Text Format (RTF)**
  A Rich Text Format file is a document that can be printed using Microsoft Word (or other word processors that can read RTF files). Use an RTF Export Process to print your data in a natural-looking format.

- **Extensible Markup Language (XML)**
  The Extensible Markup Language is a data format for structured document interchange on the World Wide Web. XML is similar to Standard Format: fields with markers are converted to elements with tags. Use an XML Export Process to convert data from Toolbox to display it using a web browser program like Microsoft Internet Explorer or Firefox or Chrome.

- **Multi-Dictionary Formatter (MDF)**
  This option is enabled if the Multi-Dictionary Formatter option is checked in the Options tab of the Database Type Properties dialog box (Properties command, Database menu).
  MDF assumes a particular set of markers. It then process the data, selecting certain fields according to the options you choose, and reorders the data to its standard.
  Use the Multi-Dictionary Formatter to export a lexical database as a dictionary or reversed index (finderlist) document that you can print using Microsoft Word.

Note: an alternative to MDF is available which has broken down the complex CC tables of MDF into smaller pieces. This allows more flexibility for printing dictionaries which do not conform to the MDF standard. Contact Toolbox@sil.org for more information.
Note: Once you have added an export process, you cannot change its type, but you can modify its properties.

**SF Export Process Properties dialog box**

Defines the properties of a named process for exporting a database in Standard Format. The exported file is like a Toolbox database, except that it lacks the "header" field. If you open the SF file in Toolbox, you will have to select an appropriate database type in the Import dialog box.

**Process Name**
A descriptive name that you select to help you remember the purpose for each export process you make. **TIP:** Use the process named Standard Format for one-time tasks.

**Fields to export**
- **All fields**
- **Select Fields button**
  When All fields is un-checked, the Select Fields dialog box determines which fields in the database's records are included in and excluded from the exported file.

**Consistent Changes Table**
When a file name is given, the CC table's rules are used to make systematic changes to the exported file. When this is left blank, no changes are made.

- **Browse button**
  Accesses the Browse for a Consistent Changes Table dialog box.

**TIP:** Keep CC tables in the project's settings folder. You can use Consistent Changes to change the order of data fields; reorganize the data for a different analysis task.

Note: Toolbox has Consistent Changes built in. It doesn't use the stand-alone CC program for DOS, Windows, or Macintosh that you may have on your computer system.

**Check consistency before exporting**
If you select this option, Toolbox checks all consistency conditions for all data fields as you have specified in the Check Consistency option under the Checks menu.

**Output file**
The name of the output file can be specified here. It contains the name of the most recent file output by this process. If this box is blank, then Toolbox asks for the name of the output file at the time of export.

**Overwrite file without asking**
If you select this option, any existing version of the file will be overwritten without asking.

**Automatically open exported file**
If you select this option, the output file will load automatically into Toolbox. It will be assigned the same database type as the file being exported unless that is explicitly changed by a CC table.

**Next process in chain (optional)**
If you enter the name of an export process here, the output file of this export process will be reloaded and then automatically exported through the "next process" export process. *For this to work you must also choose to automatically open the exported file.*

Note that it is possible to modify the chaining sequence by putting at the top of the data file the line "\_chain X", where X is the name of the next process in the chain. When an export process automatically opens a file, if the file contains "\_chain", then the named process is used as the next process in the chain. The normal way this is done is to have the export CC table of an SF export process add the chain command to the file according to some option or other condition. This allows a CC table to modify the sequence of export processes.

Also note that using this feature actually changes what is specified as the "next process" in the export process.

**Add alphabetic dividers**
If you select this option, the export process will insert alphabetic dividers of the kind used in dictionaries.

**Order of processing steps:**
1. Filtering records (if either the Current window's records or the Current record only option is selected).
2. Selecting fields.
3. Making CC table changes to the data being exported.

If you are Adding, Copying, or Modifying a process, choosing OK returns to the main Export dialog box.

When you export a database using an SF process, this dialog box appears if the Show Properties dialog box on Export option is enabled in the Export dialog box. If you modify any properties, they will affect this exported file and will also be remembered for future exports. Choosing OK accesses the Exported File dialog box. However, if the Current window's records option is selected in the Export dialog box and the window's primary sorting marker isn't the record marker, the Export Current Window dialog box will appear next.
RTF Export Process

Properties dialog box

This dialog box defines the properties of a process for exporting a database as a document in Rich Text Format. The exported file can be printed using Microsoft Word (or other word processors that can read RTF files).

To better understand RTF export, notice the following correspondences:
- a Toolbox database is converted to a formatted document;
- Standard Format markers in the database are converted to styles in the document; the database type defines the markers and a document template defines the styles.

This dialog box offers the following options:

Process Name
A descriptive name that you select to help you remember the purpose for each export process that you make. TIP: Use the process named Rich Text Format for one-time tasks.

Fields to export
- All fields
- Select Fields button
When All fields isn't checked, the Select Fields dialog box determines which fields in the database's records are included in and excluded from the exported file.

Consistent Changes Table
When a file name is given, the CC table's rules are used to make systematic changes to the exported file. When this is left blank, no changes are made.
**Browse button**
Accesses the Browse for a Consistent Changes Table dialog box. **TIP:**
Keep CC tables in the Toolbox project's settings folder.

You can use Consistent Changes to
- add formatting (for example, labels to identify data fields; see also Special symbols used in exporting data)
- change the order of data fields
- select styles according to context-sensitive rules (by changing or inserting markers)

Note: Toolbox has Consistent Changes built in. It doesn't use the stand-alone CC program for DOS, Windows, or Macintosh that you may have on your computer system.

**Style formatting**
Markers are converted to styles (the marker's field name is the style name). The Style to Export option in the Marker Properties dialog box determines whether a marker becomes a Character or Paragraph style.

Note: If a record's first field has a character style, a new Normal paragraph is automatically started.

**Document template**
This option lets you use all the style formatting attributes that Microsoft Word provides, not just the few that you can define in Toolbox (e.g., Toolbox doesn't have paragraph formatting).

**TIP:** Keep document templates in the Toolbox project's settings so you can back it up with the rest of the project.

When a file name is given, the exported document will have that document template attached so that styles will be updated automatically when it is opened in Microsoft Word. (When this option is selected but the file name is left blank, the exported document gets "plain vanilla" formatting, 10 point regular Times New Roman. After you open it in Word, you can attach a template and update the styles.)

**TIP:** To create a document template, export your database the first time using Marker font properties (the other Style formatting option), open the exported document in Microsoft Word, delete all its text, and save it as a document **template**.

**Marker font properties**
Text in the exported document gets character formatting the same as the corresponding fields have on the screen in Toolbox (i.e. styles in the document are defined by Marker and Language Font properties in Toolbox).

Note: Colors you may have chosen for data fields in Toolbox aren't exported: all the text in the exported document is black.

**Page Setup button**
Accesses the Export Page Setup dialog box (which has a Document tab and a Section tab). Templates in Microsoft Word don't update a document's page and section formatting, only its paragraph and character styles. Therefore, you must select options like paper size and number of columns in the Export Page Setup dialog box to avoid having to select them over and over again each time you open an exported RTF file in Word.

**Interlinear**
There are some special formatting features for databases that contain interlinear aligned data fields. In the following sample, the `\t`, `\m`, `\g`, and `\p` fields are aligned, but the `\f` field is not.

```
\t Anong oras?
\m anos -ang oras
\g what is the hour
```
What time is it?

Align fields using tabs
You have more formatting choices in the word processing program if Toolbox uses tabs to align the interlinear data, therefore this is the default method. The alternative—alignment using spaces—requires that the Text, Morpheme, Gloss, and Part of speech styles have the same fixed width font at the same point size. If the text is written using a non-Roman script and the glosses are written using a Roman script, alignment using spaces just doesn't work.

Spacing
Toolbox doesn't compute the ideal spacing for aligned interlinear fields when it exports them in RTF. It sets the tabs stops using a spacing factor (i.e. nominal character width) that you can adjust to get pretty good results for the fonts that you use. The default value is 120; values between 60 and 240 are allowed. If words in any interlinear line seem to touch or overlap each other, increase the spacing; if words have excess space, decrease it.

It is generally better to have a little extra space than not enough. The examples in your paper will not look bad if they have a bit more space than the minimum. But if some words in an example touch or overlap, that will look bad.

Experiment with this number until you get a value that gives good spacing most of the time. Then if in a specific example a particular annotation is noticeably too tight or loose, you can adjust the spacing of that example by putting the cursor in the example and dragging one or more tab stops slightly up or down.

Note: In RTF Export, spacing is measured in "twips" (1440 twips = 1 inch, 567 twips = 1 centimeter). Therefore, you can set the spacing to an average of between 6 and 24 characters per inch.

TIP: Set the wrap margin so that Toolbox limits the width of interlinear lines to what will fit the margins that you have defined for the exported document (using Page Setup, discussed previously).

Check consistency before exporting
If you select this option, Toolbox checks all consistency conditions for all data fields that you have specified under Checks, Check Consistency.

Output file
The name of the output file can be specified here. It contains the name of the most recent file output by this process. If this box is blank, then Toolbox asks for the name of the output file at the time of export.

Overwrite file without asking
If you select this option, any existing version of the file will be overwritten without asking.

Add alphabetic dividers
If you select this option, the export process will insert alphabetic dividers of the kind used in dictionaries. Note, these are inserted as Standard Format entries, using the marker \zzzAlphDiv. This allows you to manipulate the alphabetic information or the data immediately before or after the alphabetic break using the CC table. As with all markers, it is converted to an RTF style at the end of the exporting process.

Order of processing steps:
1. Filtering records (if either the Current window's records or the Current record only option is selected under Records to include in the Export dialog box).
2. Selecting fields.
3. Making CC changes to the exported records and fields.
4. Converting data markers to styles, defining the styles, and defining page and section formatting.

Note: RTF and SF export processes share steps 1-3. An RTF process writes a temporary SF file, which is converted into an RTF file in step 4.

If you are Adding, Copying, or Modifying a process, choosing OK returns to the Export dialog box.

When you export a database using an RTF process, this dialog box appears if the Show Properties dialog box on Export option is checked in the Export dialog box. If you modify any properties, they will affect this exported file and will also be remembered for future exports. Choosing OK accesses the Exported File dialog box. However, if the Current window's records option is selected in the Export dialog box and the window's primary sorting marker isn't the record marker, the Export Current Window dialog box will appear next.

If the Automatically open document in word processor option is enabled in the Export dialog box, Toolbox will cause the application associated with RTF files (usually Microsoft Word) to open the exported file (on Windows systems only).

**TIP:** If an exported RTF file is already open in Word, remember to close it before exporting again (for example, if you find a misspelling in the document and correct it in the Toolbox database). Toolbox cannot write a file of a name that Word has open, so leaving the file open in Word will cause Toolbox to abort the export.
Page Setup, Document tab (RTF and MDF export processes)

Templates in Microsoft Word don't update a document's page and section formatting, only its paragraph and character styles. To avoid having to select document-level formatting over and over again each time you open a file exported by an MDF or RTF process, you can set the following options for documents exported by this process:

**Paper**

- **Size**
  - Letter, legal, A4, or a custom size given in either English or metric units.
  - Note: The paper size determines the unit of measure used for defining other formatting attributes. If the paper size is A4 or custom metric, then you must use centimeters to define the margins, etc.
- **Width and Height**
  - These are enabled when custom paper size is selected.

**Margins**

- **Top**
- **Bottom**
- **Left**
  - If you select the Mirror margins option, this will be the Inside margin instead.
- **Right**
  - If you select the Mirror margins option, this will be the Outside margin instead.
- **Gutter**
  - Gutter refers to the area along the inside edge of a printed page, used to bind the document as a book. Any gutter dimension is added to the left (or inside) margin to determine the placement of the text on the page.
- **Mirror margins**
  - Check this box if odd and even pages of the document are to be printed front-to-back on the paper (like a book). This option reflects the margins so that the right margin for the right (odd) page becomes the left margin for the left (even) page and vice versa.

Note: when switching from English to metric or vice versa, Toolbox does not convert the values specified for custom page size or margins. So a Top Margin of one inch will become one centimeter if you change to metric and don't adjust the margin values.
Page Setup, Section tab (RTF and MDF export processes)

Templates in Microsoft Word (version 6 and later) don't update a document's page and section formatting, only its paragraph and character styles. To avoid having to select section-level formatting over and over again each time you open a file exported by an MDF or RTF process, you can set the following options for documents exported by this process:

**Header and footer**

- **Header to edge**
  The distance from the top edge of the header text to the top edge of the paper.

- **Footer to edge**
  The distance from the bottom edge of the footer text to the bottom edge of the paper.

- **Different first page**
  Select this option when the first page of the document requires different formatting from other pages (e.g. no header, page number in the center of the footer).

- **Different odd and even**
  Select this option when the document will be bound like a book: headers and footers will be set up appropriately for right (odd) and left (even) pages.

  Note: You can select any combination of the first page and odd/even options: none, both, one or the other.

**Columns**

- **Number**
  The number of columns on the exported page. You may choose one or two columns for MDF export; between one and four columns for general-purpose RTF export.

- **Spacing**
  The width of the white space separating columns of text. Experimentation is the best way to determine what looks best.

- **Line between columns**
  Print a vertical line between columns.

Note: The unit of measure is determined by the paper size defined in the Document tab.

Special symbols used in exporting data

Export dialog box (Export command, File menu)
When exporting data in Rich Text Format using an RTF export process or the Multi-Dictionary Formatter, there are special symbols that will be converted to formatting characters. The notation uses the vertical bar and curly braces. The following are currently available:

|~| Non-breaking space
|\-| Optional hyphen
|\_| Non-breaking hyphen
|\tab| Tab
|\emdash| Emdash
|\endash| Endash

**TIP:** Avoid using special symbols (or any other kind of direct formatting) in your database. It's better to use a Consistent Changes table to insert them when exporting the RTF file.

Example: MDF’s change tables convert ordinary hyphens in vernacular language fields to non-breaking hyphens in the exported document (to prevent hyphens used in affixes and bound roots from getting separated from the morpheme by a line break).
Export Current Window dialog box

Export dialog box (Export command, File menu)

If the Current window's records option is selected in the Export dialog box and the window's primary sorting marker isn't the database's record marker, you can choose to export the records as-is or "reversed" (i.e. the primary sorting field becomes the record marker in the exported file).

Moving the relevant sorting field to the top of the record
You'll probably want to select this option if the primary sorting field can occur multiple times in a record.

Example: If you sort a lexical database window by gloss and export it using this option, you'll get a reversed index (finderlist). In the exported file the gloss will be the record marker. You'll probably also use Select Fields to include fields like lexeme and part of speech, but to exclude most other lexical data fields.

Note: If you open such an exported Standard Format file in Toolbox, you'll need to make an appropriate database type for it, since the record marker is different from the original database. If you export as RTF, you may want to attach a different document template that defines appropriate formatting.

CAUTION: Consider carefully whether you actually need to export the database. Many data exploration and analysis tasks can be done within Toolbox using Filtering, Sorting, and Browsing.
Select Fields dialog box (Exporting)

This dialog box is used to select which fields are exported from Toolbox. Use the Add and Remove buttons to move markers between the Excluded and Included lists.

TIP: Use the Shift and Ctrl keys with mouse clicks or arrow keys to select multiple markers to Add or Remove all at once.

Automatically include new markers
When new markers are added to this database type, should this export process include them or exclude them?

Multi-Dictionary Formatter:
By default, the `\dt`, `\ec`, `\es`, `\ls`, `\re`, `\rn`, `\sd`, `\so`, `\st`, `\th`, `\we`, `\wn`, and `\wr` fields are excluded. Use this dialog box to include any of these fields, or to exclude groups of fields not controlled by MDF's standard Options. Excluded fields are omitted unconditionally, independent of MDF's other field selection options.

Example: To print example sentences without their references, check the Include example sentences option but move the `\rf` marker to the Excluded list.

CAUTION: Don't exclude the `\lx`, `\hm`, `\lc`, `\se`, `\ps`, `\pn`, or `\sn` fields!

Note: You can't use this dialog box to cause fields to be exported in a different order. However, you could write a Consistent Changes table to reorder fields and select it in either the SF Export Process Properties or RTF Export Process Properties dialog box.

Exported File dialog box

Similar to a File Save As dialog box. It offers the following options:

Save in
Shows the folder (directory) and drive where the exported file will be written. Allows navigation to a different folder and/or drive. Click the drop arrow to see the directory tree.

File Name
The name of the exported file. Each export process remembers the last file you exported using it.

Save as type
If you are exporting using an SF process, you can choose the file extension freely; when using an RTF or MDF process, use.\rtf as the extension.
Multi-Dictionary Formatter (MDF): An Overview

Printing a dictionary frustrates many linguists and anthropologists. Converting lexical databases to formatted documents has been so difficult that—for some—it never gets done before editing is completed. The Multi-Dictionary Formatter (MDF) enables ordinary computer users to print a dictionary. Removing the difficulty from printing saves time; adding the ability to inspect fully formatted lexical entries throughout the dictionary-making process improves quality.

Toolbox includes MDF as a kind of export process in the Export command (on the File menu). If you use its system of data field markers, MDF can automatically format your lexical data as a dictionary or reversed index (finderlist). Toolbox can check the data for consistency before exporting it. The data is exported as a document in Rich Text Format (RTF), which you can print using Microsoft Word. You can also convert the document to a Portable Document Format (PDF) file using Adobe Acrobat. MDF works on both Windows and Macintosh computers. Since MDF uses document templates and character styles, it requires version 6 or later of Microsoft Word. (Note: Adobe and Acrobat are registered trademarks of Adobe Systems Incorporated.)

MDF is flexible enough to organize lexical data from all over the world for a variety of purposes and print it in several formats. It defines about 100 data field markers. The average dictionary compiler will use about 20 to 30 of them on a regular basis. The markers follow a mnemonic system for ease of learning and recall. Please note that Toolbox does not require you to use MDF. If you already have a well-organized set of markers for lexical data, you can continue to use them (by defining your own database type).

Using Unicode with MDF

In order for Unicode to work correctly with MDF, it is necessary that you turn on the Unicode switch in all language encodings.

In particular, you must turn on the Unicode switch in the Vernacular, National, and Regional language encodings even if you do not use them. This is because the MDF export process changes fields to those language encodings during its processing. If they are not set to Unicode, the output will not be correct.

Multi-Dictionary Formatter (MDF) dialog box

Export dialog box (Export command, File menu)

Defines the properties of a named process for exporting a lexical database as a dictionary or reversed index (finderlist) document. Since MDF uses document templates and character styles, it requires version 6 or later of Microsoft Word. It works on both Windows and Macintosh.

For more information, read Multi-Dictionary Formatter: An Overview

Note: The version of MDF built into Toolbox is compatible with existing lexical databases, however it doesn't use any of DOS MDF's change tables, answer files, stylesheets, etc.

This dialog box offers the following options:

Process Name
A descriptive name that you select to help you remember the purpose for each export process.
Audience
Determines the second language in a dictionary and the indexing language in a finderlist. When you select Audience and Format, MDF includes fields that contain data in the appropriate languages. Most of the vernacular language fields are included. If you choose Options, you can include or exclude groups of fields more selectively.

   **English National**
   For example: French, Indonesian, Spanish.

   **Setup**
   Accesses the Multi-Dictionary Formatter Audience Setup dialog box, in which you can select change tables that define labels and document templates that define formatting, especially for the national language audience.

Format
You can use a single lexical database for different purposes by printing it in the appropriate format:

   **Dictionary**
   Consists of lexemes in the vernacular language and translation equivalents with reference to...

   **Diglot**
   ...a second language, for example, Buru - English (Vernacular - English).

   **Triglot**
   ...a second and third language, for example, Buru - Bahasa Indonesia - Ingris (Vernacular - National - English).

   Note: Triglot format isn't usually recommended for publication, but it is useful during the drafting and pre-publication stages to check for consistency and completeness. **Gloss index (finderlist)**
   Index to vernacular lexemes, subentries, and senses of meaning by translation equivalents in another language. For example, Bahasa Indonesia - Buru (National - Vernacular). In version 4.1 (and later), the index language is independent of the audience language: you can select English, National, or Regional language glosses, for either an English or a National audience.

MDF doesn't focus on printing monolingual dictionaries, although it does support the necessary data fields.

Title in the footer of the document
You can choose a title for each of the ten combinations of Audience and Format. The title appears in the center of the footer at the bottom of each printed page in the exported document. If you don't want a title, delete the contents of this box.

Filter records
If you select a filter, the exported dictionary or finderlist includes only the database records that match it. MDF doesn't use the current window's filtering properties.

   **TIP:** You can use filters to publish subsets of the lexicon as separate volumes focusing on topics such as plants, animals, or kin terms.

Options button
Accesses the Multi-Dictionary Formatter Options dialog box. It contains options for data selection and formatting that don't tend to be changed very often.

Exported file type
When lexical data is exported as a document (in Rich Text Format), it can be printed using a word processor: Microsoft Word (version 6 or later).

Check consistency before exporting
If you select this option, Toolbox checks all consistency conditions for all data fields.

Order of processing steps:
1. Filtering records
2. Selecting fields
   a. Fields in the Select Fields Excluded list are excluded unconditionally, independent of any other options (dictionaries only).
   b. Fields are conditionally included or excluded according to the Audience and Format options, also the Include... check boxes under Options. This step is implemented by the MDFDict1.cct and MDFList1.cct change tables.
3. Sorting entries
   Note: If entries aren't ordered correctly, check the Sort Orders in the Language Properties for the following markers: \lc, \lx, \hm, \re, \rm, \ps, \pn.
4. Formatting
   This step is implemented by the MDFDict2.cct and MDFList2.cct change tables.
5. Changing data and range set labels
   This step is implemented by the MDF_xyz.cct change tables selected under Audience Setup.
6. Converting to RTF
   Document: markers are converted to styles according to the special MDF_RTF.typ database types and the appropriate document template is attached.

If you are Adding, Copying, or Modifying a process, choosing OK returns to the Export dialog box.

When you export a database using an MDF process, this dialog box appears if the Show Properties dialog box on Export option is enabled in the Export dialog box. If you modify any properties, they will affect this exported file and will also be remembered for future exports. Choosing OK accesses the Exported File dialog box. If you have the status bar showing in Toolbox, you will see on the progress indicator that MDF is selecting, sorting, formatting, and exporting your lexical data. The MDF Reminder dialog box tells you about the "Finish MDF Export" command on the Tools menu in Microsoft Word, that you use to complete the MDF RTF formatting process. (Note: This may be called "Finish exporting from Shoebox" in older versions of the template.)

Multi-Dictionary Formatter (MDF) Audience Setup

This dialog box offers the following options for both the English and the National language audience:

Change table for labels
MDF uses data labels to distinguish many of the fields in a dictionary entry. You can select Consistent Change tables to modify MDF's default English labels or to translate labels for the national audience. Three fields contain user-defined data that may need to be translated or modified for printing: \lf Lexical function, \pdl Paradigm label, and \ps Part of speech. These data items are called range set labels,
because they have a limited set of values that you are encouraged to define in Toolbox as Range Sets. You can translate the range set labels that occur in your lexical database file by editing the CC table that you select.

Change tables for French, German, Indonesian (the default), and Spanish come with MDF.

**Browse button**
Accesses the Change Table for English/National Audience Labels dialog box.

**RESTRICTION:** MDF requires that its CC tables be in the Toolbox project’s settings folder.

**Document template**
MDF converts database markers to document styles. Style formatting is defined in a document template file. If you have to select an appropriate font or modify the format to match publishing conventions in your area, you do it by modifying the template’s styles in Microsoft Word.

Use the national language template to account for format differences for the national audience. By default the only difference is that the italic attribute of the English and national styles is exchanged. That makes the second language regular, and the third language (of a triglot dictionary) italic, no matter which audience is selected.

**Browse button**
Accesses the Document Template for English/National Audience dialog box.

**TIP:** Keep document templates either in the Toolbox project’s settings folder or in Microsoft Word’s folder for templates.

**Multi-Dictionary Formatter (MDF) Options**

This dialog box contains options that don’t tend to be changed very often.

**TIP:** If you select different options for draft and final printouts (note the suggestions below), you can make a second MDF process using the Copy button in the Export dialog box.

**Dictionary**

**Select Fields to be Excluded button**
When you select Audience and Format, MDF includes fields that contain data in the appropriate languages. Most of the vernacular language fields are included. By default, the `\dt, \ec, \es, \ls, \re, \rn, \rr, \sd, \so, \st, \th, \we, \wn, and \wr` fields are excluded. To include any of these fields, or to exclude groups of fields not controlled by the standard options, use the Select Fields dialog box.

Excluded fields in the Select Fields dialog box are omitted unconditionally, independent of the other field selection options. Included fields (see the options below) are printed conditionally, if they haven’t been excluded by any other options.

Example: To print example sentences without their references, check the Include example sentences option, but move the `\rf` marker to the Excluded list in the Select Fields dialog box.

**CAUTION:** Don’t exclude the `\lx, \hm, \lc, \se, \ps, \pn, or \sn` fields!

**Sort by `\lc` citation form**
Because the headword comes from the `\lc` field if there is one, the default is to sort by it. If you want entries to be sorted by the `\lx` Lexeme field even when there is an `\lc` Lexical citation field, uncheck this option.
CAUTION: If \( \text{x} \) and \( \text{c} \) have different language encodings (e.g. one is a non-Roman script and the other a Romanized transliteration), it doesn’t make sense to sort by \( \text{c} \) unless every record has a citation form.

Include \( \text{x} \) lexeme with \( \text{c} \)
If you check this option, MDF would format

\[
\text{x} \text{-hilu} \\
\text{c} \text{ nahilu}
\]
as

nahilu (from: -hilu)

Note: The Sort by \( \text{c} \) and Include \( \text{x} \) options work together.

Include example sentences
Controls the \( \text{rf}, \text{xv}, \text{xe}, \text{xn}, \text{xr} \) fields as a group.

Include notes
Controls the \( \text{nt}, \text{na}, \text{np}, \text{ng}, \text{nd}, \text{ns}, \text{nq} \) fields as a group.

Include regional with national
When the Audience and Format options cause national language fields to be included, this option includes regional fields too, immediately following the corresponding national fields.

Include non-MDF fields
You can freely use non-MDF markers in your database, but MDF prints them only if you check this option. This allows you to include special bookkeeping fields for your own use in Toolbox without having to worry about how they will print. If you want your non-MDF fields included in a draft printout, they will be grouped together at the end of a given hierarchical section (\( \text{x}, \text{se}, \text{ps}, \text{sn} \)) in square brackets marked with question marks [??].

Gloss index (finderlist) Include part of speech
By default, MDF includes the part of speech with each gloss in the index. This makes it clear whether glosses like grill refer to a noun or verb, and avoids the problems associated with glosses like to grill.

Marker hierarchy
The hierarchical order of the \( \text{x}, \text{se}, \text{ps}, \) and \( \text{sn} \) markers is defined in the database type associated with this lexical database.

Document formatting
Headwords in the header
Puts the first and last headwords (lexeme in a dictionary, gloss in a finderlist) that occur on the page in the header, like in a typical dictionary. Uncheck to option to get a blank header.

Date in the footer
Puts the current date in the page footer. This helps you clearly identify draft printouts made at different times. Uncheck this option for final printouts.

Total number of entries
By default, MDF will automatically count every entry in the formatted document and put the total number on the last page. You’ll probably want to uncheck this option for final printouts.
**Page Setup** button
Accesses the Export Page Setup dialog box (which has a Document tab and a Section tab). Templates in Microsoft Word (version 6 and later) don't update a document's page and section formatting, only its paragraph and character styles. Therefore, you must select options like paper size and number of columns in the Export Page Setup dialog box to avoid having to select them over and over again each time you open an exported RTF file in Word.

**Web page copyright**
Since MDF can be used to prepare language and culture data for electronic publication on the World Wide Web, it includes copyright information in the header and at the bottom of each exported web page. If you don't need or want copyright information, delete the contents of this box. If you need to insert the copyright symbol, press the Alt key and type the digits 0 1 6 9.

**MDF Reminder dialog box**
To complete the formatting process, you need to choose the “Finish MDF Export” command in Microsoft Word. This is a menu customization in the document template that runs a macro.

Note: You may find it called “Finish exporting from Shoebox” in older versions of the template.

The macro
1. reverses right-to-left data fields (if you use the MDF_eRtl.dot or MDF_nRtl.dot templates)
2. inserts graphic pictures
3. inserts page breaks before section headings that would be left alone at the bottom of a page
4. saves the exported file as a Word Document (a *.doc file)

When you see a dialog box that says “Finished exporting”, the document is ready to print.

**TIP:** One way to take care of specialized formatting issues beyond MDF's built-in capabilities is to modify this macro.

**Direct Printing**

**Overview**

**Print (File menu)**
Controls how a database or record is printed. Before using this command, a printer must be installed and selected. To install a printer, use either the Windows Setup or the Windows Control Panel. For information about installing printers, see your Windows documentation. To select a printer, see the Print Setup dialog box.

**To print**

1. Do one of the following:
   - Use the shortcut keys, CTRL+P.
   - From the File menu, choose Print.
2. The Print dialog box opens with the following options:
   **Printer**
   Displays the name of the active printer and the printer connection.

**Print Range**
Specify the pages you want to print.
All
Prints the entire database.

Selection
Prints only the selected text.

Pages
Prints the pages you specify. Type page numbers separated by commas, or a range of pages with a hyphen between the page numbers. For example, to print pages 2, 4, 5, 6, and 8, type 2,4-6,8

Print quality
Choose a resolution depending on the printer chosen. Resolution is measured in dots per inch (dpi).

Copies
Prints multiple copies. Type the number of copies desired.

Collate copies
Organizes pages when you print multiple copies. Toolbox prints a complete copy of the first print assignment before it begins to print the first page of the second.

Setup button
Opens the Print Setup dialog box.

Note: To print the Browse view, do the following:
Access the Browse view.
From the File menu, choose Page Setup.
In the What to Print option, choose This Record Only.
Choose the OK button.
From the File menu, choose Print

Note: There is a bug which can cause printed lines to overlap each other at the top of a page. This bug happens only when the top line of a page is a continued line inside a field, not the first line of a field. To work around the bug, add extra lines as needed to change where the page break happens. or use the shortcut keys, CTRL+P.

---

**Page Setup (File menu)**

This command is used to specify the page layout of the printed database—the page margins, the header and/or footer, which records to include and possible page breaks between record.

When a new database type is created, Toolbox assigns default margin settings and a default header and footer. These settings are effective for all databases of this type.

**To setup the page**

1. From the File menu, choose Page Setup.
2. The Page Setup dialog box opens with the following options:

**Header**

Specifies the header to be printed at the top of every page. The header may contain any text, plus printing codes for printing the time, date, file name, page number and other data.

**Footer**

Specifies the footer to be printed at the bottom of every page. The footer may contain any text, plus printing codes for the time, date, file name, page number and other data.

**Margins**
Specifies the size of the margins for the printed pages. Type the measurement for the margin you want to adjust in the Left, Right, Top or Bottom box.

**What to print** Choose between:

- **Whole view**
  Prints all the records in the current view (including the Browse view) as filtered and sorted.

- **This record only**
  Prints only the current record.

**Page break between records**
When enabled, each record is printed on a new page. When disabled, records are printed continuously with a space between each record.

**Note:** If you have sorted by a field with multiple occurrences in a single record, that record will be reprinted for each occurrence of that field. In other words, if there were five occurrences of the primary sort field within a single record, that record would be printed five times.

**Printing codes for the header and footer**
There are printing codes which can be used within the header and the footer for the time, date, file name, page number and other data. This can help to organize printouts.

<table>
<thead>
<tr>
<th>Printing code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;F</td>
<td>File name (without path)</td>
</tr>
<tr>
<td>&amp;P</td>
<td>Page number</td>
</tr>
<tr>
<td>&amp;n</td>
<td>Record number</td>
</tr>
<tr>
<td>&amp;N</td>
<td>Number of records in printed view</td>
</tr>
<tr>
<td>&amp;R</td>
<td>Record marker field</td>
</tr>
<tr>
<td>&amp;r</td>
<td>Contents of record marker field</td>
</tr>
<tr>
<td>&amp;K</td>
<td>Primary sort field</td>
</tr>
<tr>
<td>&amp;k</td>
<td>Contents of primary sort field</td>
</tr>
<tr>
<td>&amp;D</td>
<td>Current date (yyyy-mm-dd)</td>
</tr>
<tr>
<td>&amp;T</td>
<td>Current time in 24-hour format (hh:mm)</td>
</tr>
<tr>
<td>&amp;L</td>
<td>Filter, if active</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>Literal ampersand character</td>
</tr>
</tbody>
</table>

**Header example**

<table>
<thead>
<tr>
<th>&amp;F, Date: &amp;D</th>
<th>Actual Header Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record &amp;n of &amp;N</td>
<td>MDFSAMPL.DB, Date: 1996-07-10</td>
</tr>
<tr>
<td>&amp;n of &amp;N, \k &amp;k</td>
<td>Record 18 of 57</td>
</tr>
<tr>
<td></td>
<td>18 of 57, \lx abu</td>
</tr>
</tbody>
</table>

**Note:** Every data code must be preceded by an & (ampersand). If the literal ampersand character is included in your header or footer, use && Printing codes are case-sensitive.

**Print Preview (File menu)**
Opens a window in which Toolbox displays a page as it will look when printed.

**To preview a printout**
Do the following:
From the File menu, choose Print Preview.
A window opens displaying a view of the page(s) to be printed. Buttons across the top of the window offer the following options:

**Print**
Opens the Print dialog box without exiting the Print Preview.

**Next Page**
Displays the next page.

**Prev Page**
Displays the previous page.

**One Page/Two Page**
Toggles between showing one or two page(s).

**Zoom In**
Enlarges the display.

**Zoom Out**
Reduces the display.

**Close**
Closes the Print Preview window.

**Print Setup (File menu)**
This command provides a list of installed printers and access to various options for the selected printer.

**To setup the printer**

From the File menu, choose Print Setup.

The Print Setup dialog box opens with the following options:

**Default Printer**
Displays the name of the default printer and shows the printer connection.

**Printers**
Choose the default printer or another printer from the list of those installed by using either Windows Setup or the Windows Control Panel. For information about installing printers, see your Windows documentation.

**Orientation**
Choose one of two page orientations: portrait (vertical) and landscape (horizontal).

**Paper**
Select the paper size and the source.

**Options button**
Lists specific properties for the selected printer

**Network button**
Connects your computer to a network printer. The Network button is available only if your computer is connected to a network.
Interlinear topics

Interlinearizing is a way to analyze text by breaking the words into morphemes and by glossing each morpheme in the line under itself. This is useful for studying the morphology and syntax of a language. Toolbox provides the means to produce interlinear text.

If you have not done interlinear text using Toolbox, it is strongly recommended that you first go through the Toolbox Self-Training Exercises. Then the following topics can be used for reference.

**Interlinearizing: An Overview**

Interlinearizing is the process of defining words in texts, and thus analyzing that text by creating interlines. Each line of text has, for example, a corresponding line where its morphemes are split off, a line for the gloss and perhaps also a line for the part of speech. Typically there will also be a free translation of the unit beneath the interlines. This information is aligned by word and/or morpheme. For example:

```
\t Anong oras?
\m ano -ang oras
\g what.is the hour
\p pron art n
\f What time is it?
```

Toolbox provides the means to gloss text in this way and to add new words to a lexical database at the same time.

If you have not done interlinear text using Toolbox, it is strongly recommended that you first go through the Toolbox Self-Training Exercises.

It is good to first decide how to organize the data. The best way to organize interlinear text is to make each text a record in a larger text file, such as is done in the Toolbox Self-Training interlinear examples, using the file `id` marker as the record marker.

Often the text is typed directly into Toolbox, either the whole text at once or sentence by sentence as the interlinearizing proceeds. If the text comes from an outside source, such as Word, it must first be prepared by inserting a set of Standard Format markers which are used in the interlinear process.

The first time text is interlinearized, one reference unit (usually a sentence) is done. The extra interlinear lines are added, the text is reshaped to the current wrap margin, and all words in the sentence are interlinearized. A failure mark (usually three stars) is put under each morpheme that cannot be found in the lexicon. The normal sequence is to then add such morphemes to the lexicon and re-interlinearize the failed words. When a word is re-interlinearized, only that word and the things under it are changed.
Text: Organizing and Preparing

Organizing the data
Before entering text, you must first decide how to organize the data. The recommended approach is to make the whole text a single record. This allows Toolbox to readily number the text at the beginning, or to renumber as needed. It also allows you to see the context in which a particular sentence occurs.

A possible second approach for long texts is breaking the text into logical segments, something like chapters. Toolbox does not need this as it can handle quite large amounts of data as a single record, but for ready access this might be a reasonable approach.

Sometimes people have made each sentence a record, using the reference as the record marker. Because this approach makes each sentence a separate record, you cannot see the sentences before and after for context. It also makes it impossible to use the Numbering and Renumbering features. This one-sentence-per-record approach is not recommended.

The text line should not be used as the record marker. Attempting to do this will cause the text to be sorted into nearly random order, which is not what you want.

Bringing text into Toolbox
If you have typed up a text in some word processing program and now wish to bring it into Toolbox, the easiest approach is to create a new (empty) record in the text database. Then cut and paste the entire text into the text line. From that point you can readily have Toolbox break and number the text.

If you do not yet have a text database, you can use the one provided in the start-up kit.

Numbering: An Overview
In preparation for interlinearization, texts are typically divided up into named and numbered segments. Usually each segment contains one sentence. This allows convenient referencing in a grammatical write-up.

If the text contains sentences which are too long for convenient analysis, the Break and Number Text feature can be used to break the text down further by specifying additional punctuation (such as semicolons and/or commas). This feature is also useful for processing texts entered directly into Toolbox, to break the text into segments for the first time.

If the text-break process results in some text segments which are too short or which are incorrectly broken (e.g., following abbreviations), these segments should be manually joined in Toolbox. The numbering of these segments can then be fixed automatically using the Renumber Text feature.

You can also rebreak sentences by hand. That may be easier. It gives a lot more control than Break and Number, or Renumber.

Setting up a Database for Numbering
In order to number or renumber a text, the database type needs to know what fields are being used for the reference line and for the text. This information is recorded on the Numbering Tab of the Database Type Properties. It is also possible there to disable numbering for databases for which numbering is not a meaningful operation.

Note: If numbering or renumbering is attempted for a database type that has not been set up for numbering, Toolbox will display the Database Type Properties dialog so that numbering can be set up.

Break and Number Text Dialog Box
The Break and Number Text dialog box is accessed by Tools, Break/Number Text. This command allows you to break a new text into numbered segments or to further break a numbered text by specifying additional text-break punctuation.
For each of the markers to be included when breaking text, Toolbox will scan them for text-break punctuation. Each text segment thus delimited will be placed in its own text field according to the Text Marker defined in the numbering setup and be preceded by a numbered reference line. The numbering of existing reference lines will also be updated so that it is sequential. See below for some examples to understand better how to use numbering.

The Break and Number Text dialog box opens with the following options:

**Apply to:**
- **Current Record**
  Renumbering will update the reference line numbers for the text segments of the record currently displayed in editing mode or currently selected in a browse view.
- **Whole Database**
  Renumbering will start with the first record in the (unfiltered) database, and update the reference line numbers for all records in the database.

**Options:**
- **Text-break Punctuation**
  Specifies the text-break punctuation characters that signify the end of a text segment. Typically, this will be sentence-end punctuation or clause-end punctuation. Toolbox will break your text at the first whitespace following any of the punctuation characters entered here.
- **Fields to be included when breaking text**
  This box displays the fields that Toolbox will search for text-break punctuation. Text in any of the fields listed here will be broken into text segments and placed into fields marked with the text marker, as defined in numbering setup. Most of the time you will want to include the text marker in this list, so Toolbox always adds it back into the list when this dialog box opens. However, it can be removed (see the examples below for a situation when this might make sense).

**Select Fields**
Pressing this button will open the Select Fields dialog box. Here you can determine which fields will be included when breaking text.

By default, only the Text Marker (defined in numbering setup) will appear in the list of included markers. You will need to add additional markers that contain text that you want to prepare for interlinearization. For example, if you import or enter text in a paragraph ('p) field, you should include this field for breaking text.

Use the Add and Remove buttons to move markers between the Excluded and Included lists. **TIP:** Use the Shift and Ctrl keys with mouse clicks or arrow keys to select multiple markers to Add or Remove all at once.

Only field markers with the same language encoding as the Text Marker will appear as possible markers to be included. This prevents you from using the Break and Number feature to accidentally change the language encoding of your text. This dialog box also excludes other special field markers, namely the Reference Marker, the Record Marker, and the Datestamp Marker.

**TIP:** If the marker you want to include does not appear in either list, make sure it has the correct language encoding. In language projects where multiple language encodings use the same or similar scripts, it is easy to assign the wrong language encoding to a marker and have the mistake go undetected.

**Name of Text:**
- **Use Contents of Field**
  When this option is selected, the field chosen will supply the "name" of the text that will appear on the reference line. By default, this is the record marker. Note that if the contents of the field chosen here has text followed by a number, Toolbox will strip the
numeric portion; otherwise, the entire contents will be used. Generally speaking, the field chosen should have a short, descriptive name.

If you choose this option and this database is included in a text corpus, you will probably want to set the primary and secondary reference markers for the text corpus to use the reference marker shown in this dialog box.

**Use this Name**
When this option is selected, the name entered will appear on the reference line. This option is useful if no existing field contains an appropriate name for the text. It can also be used to generate reference lines with numeric references only, by leaving the name blank. If breaking and numbering text for the whole database, use this option only if you intend for all records to have the same name (or no name) and for sequence numbers to run sequentially throughout all records of the database.

**Starting Number**
Enter the number at which Toolbox should begin numbering. If you are renumbering the whole database, this will only apply to the first record. You will almost always want to accept the default; this option is mainly provided to allow for sequence numbers that span files or records in unusual ways.

**Numbering Examples**
1. Assume you have a database type with record marker `\id`, and that numbering setup has been performed so that `\ref` is the reference marker and `\tx` is the text marker. Assume the data looks like this:

   ```
   \id    Gahip
   \ref   Gahip.001
   \tx    Nengmondi tanduwa numia gumbuwa zuma nagdi, mofita por zulumi felwopi bala wala.
   \ref   Gahip.002
   \tx    Baing bai bala hainino, ba utau ngaxang bu ngala ngautu gahibigua.
   ```

   You decide that the text segments would be easier to work with if they were broken down by clauses instead of sentences. So you choose Break and Number Text for the Current Record. You add a comma (,) to the list of text-break punctuation. You leave the Name of Text section as is, so that the name will be taken from the `\id` field. And you accept 001 as the starting number. When you press OK, the result looks like this:

   ```
   \id    Gahip
   \ref   Gahip.001
   \tx    Nengmondi tanduwa numia gumbuwa zuma nagdi,
   \ref   Gahip.002
   \tx    mofita por zulumi felwopi bala wala.
   \ref   Gahip.003
   \tx    Baing bai bala hainino,
   \ref   Gahip.004
   \tx    ba utau ngaxang bu ngala ngautu gahibigua.
   ```

2. Using the same database type as in the first example, you insert a new record and enter the following:

   ```
   \id    Frog Meets Fish
   \tx    Todn lyfch nyr velgow. Lyfch tap plap joygow. Lyfch tap plapegonsew. Dwn todn jlyj lyfch blo wiglo bzervegonsew. Don jlyn dwj todj tap bzervegonso. Dwn todn doj jlyj blo bzervegonsew. Dwn todn sezgonsew, "Detn klimen blo doshusten eh?" Don jlyn sezgonso, "Dukyt! Dukytch joyme!"
   ```

   This is a new story that has never been broken and numbered before. You choose Break and Number Text for the Current Record. You accept the default list of text-break punctuation (.?!).
You change the \t field to \tx. You change the Name of Text section to "Use this Name" and specify the name "Frog". And you accept 001 as the starting number. When you press OK, the result looks like this:

\id Frog Meets Fish
\ref Frog.001
\tx Todn lyfch nyr velgow.
\ref Frog.002
\tx Lyfch tap plap joygow.
\ref Frog.003
\tx Lyfch tap plapegonsew.
\ref Frog.004
\tx Dwn todn jlyj lyfch blo wiglo bzervegonsew.
\ref Frog.005
\tx Don jlyn dwj todj tap bzervegonso.
\ref Frog.006
\tx Dwn todn doj jlyj blo bzervegonsew.
\ref Frog.007
\tx Dwn todn sezgonsew, "Detn klimen blo doshusten eh?"
\ref Frog.008
\tx Don jlyn sezgonso, "Dukyt! Dukytch joyme!"

3. Using the same database type as in the previous examples, you now import a large text. In fact, it's so large that you and your co-worker have it split into two files so you can work on the second half while he works on the first. So you import the second half, and you have something like this:

\id Chapter.6
\ref Chapter.6.001
\tx Luego, Marcos entro su canoa.
\ref Chapter.6.002
\tx Navegaba a un pueblito cercano.
\ref Chapter.6.003
\tx Al llegar, saludo a la gente.
...
\ref Chapter.6.120
\tx Despues de platicar un ratito, los hombres salieron a pescar.

\id Chapter.7
\ref Chapter.7.001
\tx Pescaron bastante esa noche.
\ref Chapter.7.002
\tx Un hombre pesco seis peces grandes.
...
\ref Chapter.7.087
\tx Marcos decidio pasar la noche alla.

The text breaks seem okay, but you really want to have a more meaningful text name and you want sequential numbering across records, rather than restarting at 000 for each new chapter. Furthermore, since you will probably eventually combine this with the first half of the story, you co-worker informs you that you should begin numbering at 1560, since his last numbered segment is 1559. So you use Break and Number Text on the whole database. You could remove all the punctuation characters and/or remove the \t field from the list of fields to be included when breaking text to avoid further text breaks, but since you didn't make any manual changes to recombine segments, it isn't necessary. Note, however, that removing the \t field from the list of fields to be considered for breaking text will speed up processing. For the text name, you select "Use this Name" and enter "Canoe" and you enter the starting number as 1560. When you press OK, Toolbox will update the names and numbers on the reference lines so that your data will look like this:

\id Chapter 6
\ref Canoe.1560
Luego, Marcos entro su canoa. Canoe.1561
Navegaba a un pueblito cercano. Canoe.1562
Al llegar, saludo a la gente.

... Canoe.1679
Despues de platicar un ratito, los hombres salieron a pescar.

... Canoe.1680
Pescaron bastante esa noche. Canoe.1681
Un hombre pesco seis peces grandes.

... Canoe.1766
Marcos decidio pasar la noche alla.

**Numbering tab (Database Type Properties)**

This tab allows you to manage the setup for numbering and renumbering text. Options include:

**Disable the Text Numbering feature**

For many types of databases (e.g., lexicons), numbering is not a useful or meaningful operation. By checking this box, the Break and Number Text and Renumber Text menu items will be disabled for this type of database. This can help to prevent accidental numbering of a database that should not be numbered.

**Reference Marker**

Select the field marker to be used for the reference line when numbering. This is usually `\ref`. If the reference marker or text marker to be used does not appear in the list of choices, switch to the Markers tab and add it first. Then return to the numbering page to select the desired markers. Note that the date stamp marker and the record marker are not valid choices for the reference marker and text marker.

**Text Marker**

Select the field marker to be used for the text segments when numbering. If the texts in this type of database are being numbered to prepare them for interlinearization, this marker should be the same text marker used as the first line of the interlinear setup.

**Renumber Text**

This command allows you to update numbering of the reference line of text segments.

The numbering of existing reference lines will be updated so that it is sequential, beginning with the starting number specified. If reference lines contain a textual name preceding the numbers, the name will be preserved. If a reference line is encountered without a name, the name from the previous reference line in that same record will be used, if any. If renumbering is being performed on an entire database and the same name is used across multiple records, the reference number sequence will span all records using the same name. Otherwise, numbering will begin with 001 at the start of each new record. See below for some examples to understand better how to use renumbering.
To renumber text

1. From the Tools menu, choose Renumber Text.
2. The Renumber Text dialog box opens with the following options:

   **Apply to:**
   
   **Current Record**
   Renumbering will update the reference line numbers for the text segments of the record currently displayed in editing mode or currently selected in a browse view.

   **Whole Database**
   Renumbering will start with the first record in the (unfiltered) database, and update the reference line numbers for all records in the database.

   **Starting Number**
   Enter the number at which Toolbox should begin numbering. If you are renumbering the whole database, this will only apply to the first record. You will almost always want to accept the default; this option is mainly provided to allow for sequence numbers that span files or records in unusual ways.

See the examples under Break and Number text for illustrations of renumbering.

### Interlinear Setup

If you use the Start-Up kit you don’t have to do any of the following interlinear setup.

This graphic illustration shows the two approaches to interlinear setup available in Toolbox. Click on the various areas of the graphic below for more information.

![Interlinear Setup Diagram]

The Interlinear tab (Properties command, Database menu) contains a Quick Setup button that is the recommended way to generate a starting interlinear setup. This quick setup, once defined, can be customized by choosing the Modify button. If the Parse process is selected, this option accesses the Parse dialog box. If the Generate or Lookup processes are selected, this option accesses the Lookup dialog box.

While not recommended, it is possible to define the Interlinear setup through the Add button in the Interlinear tab.
Interlinear tab (Database Type Properties)

This tab allows you to manage the setup for interlinearizing text. Options include:

**From To Process**
This list box shows the current interlinear setup. It shows a summary list of active interlinear processes. Each process produces a line of interlinear text. For each process, the list shows the type of process, the marker it comes from, and the marker it goes to. For example:

```
 t  m  parse
 m  g  lookup
 m  p  lookup
```

The first process goes from the original text line to the first interlinear line. This line containing morpheme breaks is a Parse process which breaks words into morphemes.

The second process is a Lookup process to look up the morphemes in the lexicon and output an interlinear gloss for each one.

The third process is also a Lookup process to look up the morphemes in the lexicon and output an interlinear part of speech.

**Add button**
Adds a new process after the currently selected one. Choosing this option accesses the Select Process dialog box.

**Modify button**
Access properties of the currently selected process:

- Parse
- Lookup
- Rearrange
- Generate
- Given

**Delete button**
Deletes the currently selected process.

**Quick Setup button**
To start a new interlinear setup, delete all old processes, and choose Quick Setup. This collects a minimum of information and sets up a typical setup, which can then be customized. The quick setup button is not available unless the process list is empty.

The Quick Setup Interlinear Text Markers dialog box is accessed through this option.

**Interlinear Spacing**
This is additional space available to space morphemes further apart. If it is zero, then the closest morphemes will touch.

**Note:** Choosing Cancel does not undo Add, Copy, Modify or Delete.

**Quick Setup Interlinear Text Markers**
This dialog box contains a list of the markers that will occur in your interlinear text files. Quick Setup generates a four line interlinear layout of text, morpheme breaks, gloss, and part of speech. This can be customized after it is generated.

You can enter any marker you choose in these boxes. If the markers do not exist in your interlinear text database type, they will be added to the type. The following options are offered in this dialog box:
Text Marker
Specifies the top marker for interlinear. This is the original text that is interlinearized. If the database is set up for renumbering text segments, this marker should match the Text Marker specified in numbering setup.

Morpheme Breaks
Specifies the marker for a line showing words parsed into morphemes, and showing the underlying form of each morpheme.

Gloss
Specifies the marker for a gloss. A gloss is a terse statement of the main meaning of the morpheme, usually one word.

Part of Speech
Specifies the marker for the part of speech or grammatical category of each morpheme.

Choosing OK, accesses the Quick Setup Lexicon Markers dialog box.

Quick Setup Lexicon Markers
(Interlinear)
This dialog box allows you to list the lexicon(s) and the markers in your lexicon used for interlinearization. The following options are offered:

Available Databases
Lists the database(s) currently open, even if minimized.

Databases to Search
Lists the database(s) containing the lexical data. For example, LEX.DB. Databases are moved in and out of this list with the Insert and Remove buttons.

Show whole path
Changes the list display to include the entire path—the drive and directories. For example, instead of LEX.DB, the display might be C:\Toolbox\Lexicon\LEX.DB.

Gloss Marker
Specifies the marker to look up for glosses.

Part of Speech
Specifies the marker to look up for part of speech.

Alternate form
Specifies the marker that parse will use for alternate forms of morphemes or words.

Underlying form
Specifies the marker that parse will use for the underlying form.

Select Process dialog box (Interlinear)
This dialog box specifies the type of process to add to the process list in the Interlinear tab. The following options are offered:
Parse
A Parse process breaks words into morphemes by separating affixes from roots and separating compound roots. It uses the affixes and roots in the lexicon to do this. There is normally only one parse process, and it is usually the first process. (It is possible to do a second parse, for example to break derivational affixes off stems, but this is not usually done because it requires all of the stems to be in the lexicon as well as all the roots.)

Lookup
A Lookup process looks up morphemes in a lexicon and outputs the content of another field. Lookup is used for gloss and part of speech. It can also be used for adapting to a related language.

Rearrange
A Rearrange process is a process in related language adaptation that makes adjustments (rearrangements) in the word and morpheme order between the source and target languages.

Generate
A Generate process generates surface forms from underlying forms. It is normally the last process in an adaptation set-up. It removes hyphens and joins morphemes together. It also applies a phonological rule file that can be used to account for morphophonemic changes.

Given
A Given process is used in related language adaptation to specify markers for texts which have already been interlinearized. It can also be used to enter interlinear text manually.

Note: Double-clicking the process name is the same as selecting it and choosing OK.

Parse dialog box (Interlinear)
The morphological parse process uses the roots and affixes in the lexicons to divide words into morphemes.

The Parse dialog box offers the following options:

From Marker
Specifies the marker of the text line to be parsed.

To Marker
Specifies the marker for the output of the parse—shows words broken into morphemes.

Lexicons
This button accesses the Lexicons for Interlinear dialog box allowing you to setup the lexicon files and lexicon markers to be used for parsing.

If parse fails
Offers the following choices for the output when the parse fails:

• Output failure mark—outputs three asterisks for the root.
• Output original word—outputs the entire original word.
• Output root guess—outputs its best guess at the root.

The fail mark option can be chosen along with the original word or root guess. In this case, the output will be a single asterisk at the beginning of the original word or root guess.
For normal interlinearizing, the recommended setting is to output the failure mark and possibly a root guess. The other settings are more useful for related language adaptation.

For adaptation, one may want to output the original word so that it can be processed through regular sound changes in a later process.

**Infix before root**

When enabled, this option causes an infix to be placed before the root from which it was extracted. If this option is off, then an infix will be placed after the root from which it was extracted.

Note that only one infix will be found in a root. But it is possible to find multiple infixes by entering the combination as a alternate. For example, if \texttt{-b} and \texttt{-c} are infixes, then you can include an \texttt{\`a -bc-} field followed by a \texttt{\`u -b- -c-} field. This will allow the two infixes to be extracted from a single root.

**Allow multiple roots**

When enabled, this option causes the parser to search for multiple roots in a word. If this option is off, then only one root is allowed in each word.

Even if multiple roots are allowed, a whole root will always override a parse made up of multiple roots. For example, in English if \texttt{blackbird} is in the lexicon, it will override the compound root parse of \texttt{black bird}.

**Search sequence**

Offers the following choices for search sequence:

- Prefer prefixes—tries prefixes before suffixes.
- Prefer suffixes—tries suffixes before prefixes.
- Balance prefixes and suffixes—tries a suffix, then a prefix, then a suffix, and so on.

This option is most useful in combination with "Longest affix override". But the most way of parsing is to turn off "Longest affix override" and use word formulas.

**Longest affix override**

When enabled, this option causes longer affixes to override shorter ones. This is a possible technique to reduce ambiguities. But it is recommended that this option not be used, and that word formulas be used instead.

**Longer override on show all**

If this option is off, then the command "Interlinearize (show all parses)" will not use longest root override, and will show more possible parses. This may cause the correct parse to be shown.

For example, the word \texttt{number} would not normally be parsed as \texttt{numb -er}. But if this option is off, then "Interlinearize (show all parses)" will show it as one of the possible parses in the list of ambiguities.

**Keep capitalization and keep punctuation**

When enabled this option indicates that capitalization and punctuation should be applied to the output line. These are for adaptation so that the adapted output will have the same capitalization and punctuation as the input.

**SH2 style parse**

When enabled, this option allows you to use a separate parse database and conjoined affix database like SH2 did. When you turn it on, additional buttons appear for those two databases. SH2 style parse also limits parsing to find only one suffix string, one prefix string, one infix string, and one root string. (Each of these can be made up of multiple morphemes.)
These buttons (Conjoined Affixes and Parse Database) both access the Lexicons for Interlinear dialog box.

**Morpheme break characters**
Provides a list of characters to be considered morpheme break characters. These are important because they are used to identify the affixes in the lexicons. These are used by lookup processes as well, but are specified in the parse process box.

**Forced gloss start and end**
Specifies the character that starts a forced gloss and the character that ends it. These should be characters that are not used anywhere in your glosses or parts of speech.

**Morphophonemic boundary**
Specifies the character that shows the morphophonemic boundary in an underlying form. It divides the underlying form of a morpheme from the changed material that is attached to the part left after the morpheme is removed.

**Enable word formulas**
When Toolbox produces both valid and invalid parses for a word because of limitations in its basic morphological parsing features, it can use word formulas to select the valid parses automatically. The formulas are kept with the rest of the Parse properties in the database type file for interlinear texts. If you disable word formulas, they aren't deleted—the parser just doesn't use them.

Choose **Formulas** to access the Word Formulas dialog box. From there, you can modify individual formulas by accessing the Word Formula Properties dialog box. The Ambiguity Selection dialog box also has a Formulas button.
Word Formulas dialog box

Symbol and Patterns
The list displays all the word formulas (in alphabetical order by their symbols). The primary formula is indicated by a P to its left. Symbols that occur in patterns are displayed in bold font style (the primary formula's symbol is also bold).

Language Encoding for Patterns
Because the patterns contain lexical data, you need to be able to select the appropriate language encoding (its options determine the font for displaying patterns and the keyboard layout for editing them).

Show expanded list
By default, the first pattern for each formula is shown. In an expanded list, the first three patterns are shown. If a formula has more patterns than can be shown, there's an ellipsis . . . at the right of the Patterns column in the list.

Buttons
• To define a new formula, choose Add.
• Occasionally, you might want to "split" a formula: select it and choose Copy. Delete any of the original patterns that the new formula doesn't need. Also modify the original formula's patterns as needed.
• To access the Word Formula Properties dialog box, select a formula in the list and choose Modify. You can also double-click it.
• When Toolbox matches the parses for a word, it starts with the primary formula (indicated by a P in the list). To make Toolbox start matching with a different formula, select it and choose Primary. (By default, the primary formula's symbol is Word. If you just want to change its symbol, modify it.)
• If you reorganize the formulas so that you no longer need one of them, you can select it and choose Delete. Toolbox doesn't let you delete the primary formula.
• If you decide to quit using word formulas, you can choose Clear before you disable them in the Parse dialog box. Even when you clear the list, Toolbox leaves a default primary formula (in case you ever did want to start again).

Lookup dialog box (Interlinear)
The Lookup process looks up morphemes in the lexicon and outputs associated information, such as gloss, part of speech, or equivalent in another language.

This dialog box offers the following options:

From Marker
Specifies the marker of the text line to be looked up.

To Marker
Specifies the marker for the output of the lookup.

Lexicons
This button accesses the Lexicons for Interlinear dialog box allowing you to setup the lexicon files and lexicon markers to be used for the lookup.
If lookup fails

Offers choices for the output when the lookup fails:

- Insert into lexicon offers to insert the word into the lexicon. For this to work, there must be a jump path from the From marker to the appropriate lexicon.

If insert into lexicon is on, and output word is also on, then if the word is in the lexicon, but the required output field is missing or empty in the lexicon, Toolbox will jump to the lexical entry and stop there to wait for the missing field to be filled in.

- Output failure mark—outputs three asterisks.
- Output word—outputs the entire original word.
- Apply CC table to output—applies a Consistent Change table to the output of the lookup.

The fail mark option can be chosen along with the original word or root guess. In this case, the output will be a single asterisk at the beginning of the original word or root guess.

For normal interlinearizing, the recommended setting is to output the failure mark and possibly the word. The other settings are more useful for related language adaptation.

For adaptation, one can output the original word so that it can be processed through regular sound changes done by the CC table.

Keep capitalization and keep punctuation

When enabled this option indicates that capitalization and punctuation should be applied to the output line. These are for adaptation so that the adapted output will have the same capitalization and punctuation as the input.

Gloss separator

Specifications a character that separates a single gloss field into multiple glosses (or any kind of output field into multiple outputs).

Stop at separator

When enabled, this option will not use any glosses after the separator character. This allows the glosses after the separator to be available for information or reversal, but not to be used for lookup. If the separator is at the front of the field, then none of the glosses in the field are used. This effectively disables the gloss field. It is possible to have multiple gloss fields, some disabled and others active. This feature can be used to control ambiguity by limiting the number of glosses seen by the lookup process.

Adaptation process

This check box allows you to specify whether or not this process is used for related language adaptation.
Rearrange dialog box (Interlinear)

In the transfer step of related language adaptation, a Rearrange process changes the order of morphemes in a word and words in a phrase.

The Rearrange dialog box offers the following options:

**From Marker**
Specifies the marker of the line to be rearranged. Rules can refer to this line.

**To Marker**
Specifies the marker for the rearranged output.

**Part of Speech Marker**
Specifies the marker containing the part of speech. Rules can refer to both the line to be rearranged (From Marker) and the part of speech line.

**Punctuation Marker**
Specifies the optional marker to use for copying punctuation from the source to the target. Normally this is not used. It is only useful for isolating languages or for CARLA Lite in which the rearrange process is the final process in adaptation. In all other types of adaptations, the Generate process copies punctuation from source to target.

**Rule File**
A reminder of the current rule file specifying the rearrangement rules.

**Browse button**
Accesses the Choose Rule File dialog box allowing you to select a rule file.

Generate dialog box (Interlinear)

A Generate process generates surface forms from underlying forms. It is normally the last process in an adaptation set-up. It removes hyphens and joins morphemes together. It also applies a phonological rule file that can be used to account for morphophonemic changes.

The rule file is applied before the hyphens are removed, so by referring to the hyphens, changes can be limited to apply only at morpheme boundaries.

The Generate dialog box offers the following options:

**From Marker**
Specifies the marker of the gloss from which the generate will look up the appropriate morpheme.

**To Marker**
Specifies the marker for the output of the generate process.

**Punctuation Marker**
Specifies the optional marker to use for copying punctuation from the source to the target. Normally this is not used. It is only useful for isolating languages or for CARLA Lite in which the rearrange process is the final process in adaptation. In all other types of adaptations, the Generate process copies punctuation from source to target.
Rule File
A reminder of the current rule file specifying the rearrangement rules.

Browse button
Accesses the Choose Rule File dialog box allowing you to select a rule file.

**Given dialog box (Interlinear)**
The Given process is used in related language adaptation to specify interlinear markers for texts which have already been interlinearized.

The Given dialog box offers the following option:

**Marker**
Specifies the marker of the interlinear line.
Lexicons for Interlinear dialog box

This dialog box allows you to specify the lexicon(s) and the markers in your lexicon used for interlinearization.

The following options are offered:

Available Databases
Lists the database(s) currently open, even if minimized.

Databases to Search
Lists the database(s) containing the lexical data to be used by a parse or lookup process. Multiple lexicons can be used as long as they all have the same database type. If you try to insert a file of a different database type from those already in the list, you will get a message saying that it cannot be done because all databases in the list must be of the same type. Databases are moved in and out of this list with the Insert and Remove buttons.

Show whole path
Changes the list display to include the entire path—the drive and directories. For example, instead of LEX.DB, the display might be C:\Toolbox\Lexicon\LEX.DB.

Available Markers
Lists the markers which are available for use in the database type.

Markers to Find
Specifies the markers that should be looked up in the lexicons. There may be more than one of these. In parsing, the first is treated specially in that it becomes the default underlying form for alternate forms that do not have an explicit underlying form.

Marker to Output
Specifies the marker to use for output. For parsing, this is the underlying form, usually \u. If no underlying form is found, then the first in the list of markers to find, usually the lexeme, is output. For lookup, the marker to output is the gloss, part of speech, related language form or other field to be output.

Punctuation issues

Punctuation characters do not have to be defined per se in a Toolbox language encoding. You can either list all characters that are not letters (word forming characters) in the Punctuation list of the language encoding, or you can leave the Punctuation list blank. If you leave the Punctuation list completely blank, then all characters that are not included somewhere in the default Sort Order are considered punctuation.

The exception is that for finding and filtering, you can use the punctuation variable.

Thus, punctuation-related concerns are handled in the following ways:

1. Sometimes Toolbox needs to process individual words in a field or text corpus—for example, when interlinearizing text or creating a word list. In this case, Toolbox uses all characters not listed in the Punctuation list, of the text's language encoding to select all the valid words. Characters listed in the Punctuation field will be ignored and assumed to be punctuation or white space.

If you leave the Punctuation completely blank, then all characters that are not included somewhere in the default Sort Order are considered punctuation.
If you see **punctuation characters being processed** by Toolbox as part of a word, or if you see the word list or interlinear process **breaking words where they shouldn't**, you need to modify the applicable Language Encoding and remove those punctuation characters from, or add the missing characters to, the Sort Order and/or Case definition. (See the Language Encoding Properties dialog box for details.)

2. In Filters and in the Find command, variables such as ![punctuation] (punctuation) can be used to match appropriate characters. You can define the exact list of punctuation characters for this purpose in the Variables tab within the Language Encoding Properties dialog box. Variables are not used in (interlinear) parsing except in reduplication.

**Interlinearize (Tools menu)**

Interlinearize means to generate interlinear text. Interlinear text can be set up for a wide variety of layouts and line contents.

Before interlinearizing, the text file must be prepared. For more information see Organizing and Preparing Interlinear Text. **To interlinearize text**

Do one of the following:

- On the toolbar, click the Interlinearize button.
- Use the shortcut keys, ALT+I.
- From the Tools menu, choose Interlinearize.

There is no dialog box associated with this command.

**Note:** This command is disabled if Interlinear processes have not been defined. For more information on the setup procedure, see the overview topic on Interlinearizing.

Other topics related to this:
- Parsing: An Overview
- Reduplication
- Underlying form
- Punctuation issues
- Showing all parses
- Verifying

**Interlinearize (show all parses) (Tools menu)**

This version of the Interlinearize command shows all parses in the Ambiguity Selection dialog box. Use it to see if parsing problems are caused by word formulas. In the list of parses, the invalid ones—that don't match the formulas—have an asterisk * to the left.

**Example**  [Example lexical database records]

If Toolbox shows all parses for **bears**, you can see that there are two valid parses and two invalid ones. In this case, morphological parsing and word formulas are both working okay.

```
  bear{Ursidae}{n} -s(pl){nsuf}
* bear{Ursidae}{n} -s(3s){vsuf}  *
  bear{hold_up}{v} -s(pl){nsuf}
  bear{hold_up}{v} -s(3s){vsuf}
```
<table>
<thead>
<tr>
<th></th>
<th>bears</th>
<th>bears</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>bear</td>
<td>-s</td>
<td>*bear</td>
<td>-s</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>Ursidae</td>
<td>-pl</td>
<td>*Ursidae</td>
<td>-3s</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>n</td>
<td>-nsuf</td>
<td>*n</td>
<td>-</td>
<td>vsuf</td>
</tr>
<tr>
<td>t</td>
<td>bears</td>
<td>bears</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>*bear</td>
<td>-s</td>
<td>bear</td>
<td>-s</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>*hold_up</td>
<td>-pl</td>
<td>hold_up</td>
<td>-3s</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>*v</td>
<td>-nsuf</td>
<td>v</td>
<td>-</td>
<td>vsuf</td>
</tr>
</tbody>
</table>
**Parsing: An Overview**

This topic describes how to structure your lexical database for parsing. The Toolbox program comes with a morphological parser which can break off any number of affixes and handle morphophonemic changes to affixes and roots. If you desire, all the necessary information for parsing can be kept in the lexical database.

**Parsing: Roots and Affixes**

The Toolbox morphological parser requires that you specify whether morphemes are affixes or roots by using the morpheme break character (normally the hyphen) in the lexical database as follows:

- **prefix** must be followed by a hyphen (e.g. x-)
- **infix** both preceded and followed by hyphens (e.g. -x-)
- **suffix** preceded by a hyphen (e.g. -x)
- **root** neither preceded nor followed by hyphens (e.g. x)

**TIP:** The convention that identifies lexemes as bound roots using a hyphen conflicts with parsing in Toolbox. You can solve this problem by using two data fields. If you use MDF markers, enter the root without hyphens—for parsing—in the \lx Lexeme field and the bound root form with a hyphen—for printing—in the \lc Lexical Citation field.

### Example 1

\t unsuccessful
\m un-  success  -ful
\g OPPOS- achievement  -ADJZR
\p neg  n  -nadjzr

\lx un-
\px neg
\ge OPPOS

\lx success
\px n
\ge achievement

\lx -ful
\px nadjzr
\ge ADJZR

### Example 2

Words in the text that include hyphens between their morphemes are also parsed correctly.

\t non-reader
\m non-  read  -er
\g not-  look_at_book  -AGENT
Infixes (Interlinear)
An infix is an affix that occurs within the root of a word.

In the lexical database, you must identify infixes by preceding and following them by the morpheme break character (usually a hyphen).

In the interlinear morpheme line, infixes appear before or after the root in which they occur, depending on the setting of the "Infix before root" option in the parse process.

An infix will be found only in a root, not in an affix. Only one infix unit will be found. To handle multiple infixes, list each combination as an alternate with its separate pieces as underlying.

Tagalog Example
\lx dara
\ps v
\ge carry
\lx -in-
\ps vinfl
\ge completed_action

\t dinara
\m dara -in-
\p v -vinfl-
\g carry -completed_action-

Compound Roots
Words with compound roots will be parsed correctly if both roots are in the lexicon. Consider the following entries in an English lexicon as an example:
\lx black
\ps adj
\ge dark
\lx bird
These entries would cause the word blackbirds to be interlinearized correctly:

```
\t blackbirds
\m black - bird -s
\g dark - flying_creature -PL
\p adj - n -ninfl
```

**Note:** In Shoebox, if the parse of a string failed, Interlinear would only show the prefixes it had attempted to pick off. Toolbox will display as far as it reached into the string, including roots. This is particularly an improvement for languages which orthographically do not have spacing between words in a phrase.

## Reduplication in Interlinear Text

Simple reduplicative processes can be represented in Toolbox. The following example specifies a reduplication of from one to three consonants followed by a vowel at the beginning of the reduplicated root, stem or word. For Toolbox to recognize an entry for reduplication, the field must contain the letters *dup* somewhere. (It may be best for these to occur at the beginning of the field so that all entries for reduplication sort together.) The allomorph fields are used for specifying the pattern to match using variables defined in the Language Encoding Properties dialog box.

```
\lx dupCV-
\a [cons][vowel]-
\a [cons][cons][vowel]-
\a [cons][cons][cons][vowel]-
\ps intens
\ge very
```

English does not make regular use of reduplication, but if the process above was English, then the following would be correct parses:

```
\t bibig
\m dupCV- big
\g very- large
\p intens- adj
\t strostrong
\m dupCV- strong
\g very- powerful
\p intens- adj
\t blablack
\m dupCV- black
```
Reduplication suffixation can also be specified, as can reduplication with fixed letters. For example, the following specifies a reduplication of the final consonant cluster with an immediate i:

```
lx -dupiC
a -i[cons]
a -i[cons][cons]
a -i[cons][cons][cons]
ps dimin
ge a_bit
```

These entries would produce parses such as:

```
t bigig
m big -dupiC
g large -a_bit
p adj -dimin
t stronging
m strong -dupiC
g powerful -a_bit
p adj -dimin
t blackick
m black -dupiC
g dark -a_bit
p adj -dimin
```

Reduplication of a whole unit can be specified as follows:

```
lx dup
a [...] ps redup
ge informal
```

By adding hyphens, you can also specify prefix full reduplication or suffix full reduplication.

The following is an English example of full reduplication, which also demonstrates its interaction with suffixation and the fact that reduplicative forms with hyphens are parsed correctly:

```
t goody-goody
m dup good -y
g informal nice -FAMIL
p redup adj familiar
```

**Reduplication (More)**

Reduplication is a linguistic term referring to duplication of all or part of a word. For example, Tagalog has a prefix indicating Future that consists of a copy of the first consonant and vowel of the root: bi-bilih.
The lexical entry for the reduplication morpheme could be:
\lx CVdup
\a [cons][vowel]-
\g FUT
This lexical entry would parse bibilih into CVdup- bilih.

In the above entry, [cons] refers to a variable named cons, and [vowel] refers to a variable named vowel. These can be named anything you choose, but the names you choose must be defined in the properties of the Language encoding.

In the above entry, the \a field shows the form of the reduplication, and the \lx field shows the underlying form, which will be displayed on the morpheme breaks line of the interlinear text.

In the above entry, [cons][vowel]- is called the reduplication template. It gives the shape of the reduplication.

NOTE that the letters dup must occur as part or all of the underlying form of the reduplication morpheme. Capitalization is not relevant, so dup, Dup, DUP could be used as part of the underlying form.

Some orthographies put hyphens between the reduplicated portions. This is allowed. For example, if bibilih were written bi-bilih, it would still parse.

If the hyphens are before the fields, then the reduplication is a suffix. For example:
\lx -CVdup
\a [-cons][vowel]
\g FUT
This entry would parse bilili into bi CVdup (not a real word).

A specific letter is used in the template, it must be found in the reduplication. For example, a template of [cons]a- matches ba-bilih.

A template of [...]- means that the entire root must be reduplicated. For example, this template matches bibi or bilihbilih, but not bibilih. This is called general reduplication.

If general reduplication is specified to be a prefix, then the first of the two identical morphemes is considered the reduplicated part. For example, bilihbilih would parse Dup- bilih. If general reduplication is specified as a suffix, then the second of the two morphemes is considered the reduplicated part. For example bilihbilih would parse bilih -Dup.

If general reduplication is a root (no hyphen on either side), then it matches the second of two identical words. For example, bilih bilih would parse bilih Dup. This works even if the first word has been broken into morphemes. For example, dogs dogs could parse as dog -PL Dup.

Infixed may occur inside reduplication. For example, if -um- is an inflex, bumblrhh can be parsed as Dupbilih -um- (or -um- Dup- bilih if inflexes are being moved to the front).

Reduplication can be found after other affixes have been removed. Other affixes can be found after reduplication has been removed.

**Alternate Forms of Morphemes**

If a morpheme has more than one surface form, it can be specified with an alternate form field (\a) as in the following example from an English lexicon:

\lx a
\a an
\ps art
The parsed form is taken from the `lx` field:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>an</td>
<td>enormous</td>
<td>phone</td>
</tr>
<tr>
<td>m</td>
<td>a</td>
<td>enormous</td>
<td>telephone</td>
</tr>
<tr>
<td>g</td>
<td>INDEF</td>
<td>very_big</td>
<td>telephone</td>
</tr>
<tr>
<td>p</td>
<td>art</td>
<td>adj</td>
<td>n</td>
</tr>
</tbody>
</table>

Consider these lexical entries for English morphemes with alternate forms:

- `lx` `in-`
- `a` `im-`
- `a` `il-`
- `a` `ir-`
- `ps` `neg`
- `ge` `OPPOS`
- `lx` `-s`
- `a` `-es`
- `ps` `ninfl`
- `ge` `PL`
- `lx` `not`
- `a` `n't`
- `ps` `neg`
- `ge` `NEG`

Some examples of interlinear text using these entries:
\t faces
\m face -s
\g head -PL
\p n -ninfl

\t haven't
\m have not
\g own NEG
\p v neg

Notice that foxes and faces are parsed correctly due to the presence of fox and face in the lexicon (but not foxe and fac). Notice also that haven't has been analyzed here as a compound.
Underlying Forms (Interlinear)

If you require that a variant form has its own lexical entry, you can use an underlying form field (\u).
Here's an alternative way of treating the word phone than the one described in the previous topic:

\lx phone
\u telephone

\lx telephone
\ps n
\ge transceiver

You can enter the morpheme breakdown of a lexical entry in an underlying form field. Consider this example of one way to handle the suppletive verb form went:

\lx went
\u go -ed
\lx go
\ps v
\ge proceed
\lx -ed
\ps vinfl
\ge PAST

It is recommended that you leave spaces between the morphemes in an underlying form field, as in go -ed above, to distinguish roots, prefixes and suffixes. In this case, the word went parses as follows:

\lt he went
\nm he go -ed
\g 3SM proceed -PAST
\p pron v -vinfl

Rather than have a separate lexical entry for each irregular form, these forms may be included in the main entry by using the alternate form field and an underlying form field. This is another way to handle went and other irregular verb forms:

\lx go
\a went
\u go -ed
\ps v
\ge proceed
\lx find
\a found
\u find -ed
\ps v
\ge locate
An underlying form field, \u, is associated with whichever \lx or \a field precedes it. (If no \u field follows the \lx or \a field, the contents of the \lx field are used as the underlying form.)

Notice that the word hit is ambiguous and so two underlying forms are given in order to force Toolbox to display the Ambiguity Selection dialog box.

The following are examples of the parse for the verbs above (with the past tense chosen for hit):

```plaintext
\t went
\m go -ed
\g proceed -PAST
\p v -vinf1
\t found
\m find -ed
\g locate -PAST
\p v -vinf1
\t hit
\m hit -ed
\g strike -PAST
\p v -vinf1
```

Sometimes it may be necessary to use an underlying form field for affix sequences. The affix -ability is an example in English:

```plaintext
\lx -able
\ps vadjzr
\ge ABIL
\lx -ity
\lx -ability
\u -able -ity
```

The preceding entries would cause the word readability to parse as follows:

```plaintext
\t readiness
\m read -able -ity
\g look_at_book -ABIL -NOMZR
\p v -vadjzr -anomzr
```
Finally, here are some examples of underlying forms of compounds:

\lx have
\a I've
\u I have
\ps v
\ge own
\lx brunch
\u breakfast lunch

Entries from above parse like this:

\t I've
\m I have
\g 1S own
\p pron v

\t brunch
\m breakfast lunch
\g morning_meal noon_meal
\p n n

Note: It doesn’t matter in this version of Toolbox whether a form that needs parsing information is included in a main lexical entry as an alternate form or has its own lexical entry (perhaps in a separate database).

**Underlying forms (More)**

Underlying form is a linguistic term that refers to a standard form of a morpheme. This term is in contrast to the surface form which refers to the way a morpheme appears in context. For example, the English surface form *went* can be considered to have an underlying form of *go-*ed.

Roots are often given underlying forms that reflect their morphological characteristics. For example, Spanish verbs are given underlying forms like *hablar*, *comer* and *vivir* that show by the -ar, -er and -ir endings to which class they belong.

The underlying form of a morpheme or combination of morphemes can be specified in the lexicon. For example, *went* can be listed in the lexicon as having an underlying form of *go-*ed. Such a listing can be either a separate entry, or it can be included under another entry—such as the entry for *go*.

For example, assume the lexicon has a record marker of \lx and the underlying form is marked with \u. One possible entry is the following:

\lx went
\u go-ed

Another possibility is to put *went* under *go*. This option would use another field marker for the alternate form. Assuming that the marker is \a, the entry could be as follows:

\lx go
\a went
\u go-ed

Underlying form is also used to show morphophonemic changes. For example, the following entry shows the change from *y* to *ie* before the plural *s* in English. (This is actually an orthographic change
rather than morphophonemic, but it illustrates the principle. In parsing both morphophonemic and orthographic changes must be handled.)

\lx -s
\a -ies
\u y+s

The \u line could also be written as:
\u y+ -s

The entry above expresses a generative rule that could be written:
y > ie / _ -s

Or it could be written:
y+ s > ies

Note that as illustrated above, if you prefer to write the underlying form of an affix with a hyphen, there must be a space between the hyphen and the plus (+).

The following principles describe how underlying forms are associated with surface forms. In these rules, \lx is the marker for the lexeme, \a is the marker for alternate form, and \u is the marker for underlying form.

1. All \u fields found between \lx and the first \a are applied to the \lx. There may be other fields between \lx and \u.

2. All \u fields found between an \a and the next \a are applied to the previous \a.

3. If an \lx or \a has no \u before the next \a, \lx is used as its underlying form.

4. An \lx or \a can have multiple \u fields. But a \u applies to only one \a or \lx, the nearest preceding.

The numbers in the following examples show how the fields are associated.

\lx 1 (lexeme)
\p N (other fields can be between)
\u lx1 (underlying form of lx1)
\u lx1 (second underlying form of lx1)
\a 2 (alternate form of lx1)
\u a2 (underlying form of a2)
\lx 2 (no u below, so acts as its own underlying form)
\a 3 (alternate form of lx2, uses lx2 as underlying form)
\a 4 (alternate form of lx2)
\u a4 (underlying form of a4)

There may be other fields besides \lx and \a in the search list. The first field in the search list is used as the underlying form for all fields not followed by an explicit underlying form. This means that all fields after the first are handled in the same way as the \a field above. If the underlying form of an affix does not have an explicit morpheme break character (such as hyphen) it is given the first morpheme break character in the list. This may give unexpected results. For example, assume the list of morpheme break characters is hyphen and equal sign '='.

Then the entry below outputs -ish instead of =ish, as follows:
\a =ish
\u e+ish
To get the equal sign, put it explicitly in the underlying form, as follows:
\( a =ish \)
\( u e+ish \)
Ambiguity: Forced Values

Where a morpheme has more than one meaning, it is sometimes helpful in the underlying form to specify which gloss is required so that Toolbox will not display the Ambiguity Selection dialog box.

Most cases for which forced values were used in the past can be resolved by using word formulas. However, the notation is shown here for use if it is needed. Consider the following entry:

```
\l x abcde
\a abcdefg
\u abcde -fg{hi}
```

This says the string "abcdef" can be parsed into two segments, "abcde" and "-fg", and the suffix "-fg" should be glossed as "hi".

The process is the same even if the ambiguity is on the root.
Ambiguity Selection dialog box

If the Toolbox parser produces multiple parses or finds multiple lookup data items, it displays this dialog box so that you can select which parse or item to insert in the interlinear text. In the main window, the ambiguous element is highlighted. If the text is covered by this dialog box, click its title bar and drag it out of the way. Toolbox will remember where you moved it.

On a small screen the dialog box may be so wide that the right edge with the scroll bar is off the screen. If that is the case, you can use Page Up, Page Down, Home and End to scroll the ambiguities to get the right one into view.

Ambiguities

☐ To insert an item from the list in the text, select it and choose OK (or double-click it). ☐ To stop the interlinearizing process, choose Cancel.

Data items shown surrounded by the forced value delimiters (usually braces or parentheses) will be inserted in interlinear fields under the \m "morpheme breaks" field.

Font

The entire ambiguity is shown in a single font, even if includes glosses and parts of speech as forced values. The font used is the language font of a field from the dictionary. If glosses are shown, then the font is the language font of the gloss field in the dictionary. If glosses are not shown, it is the font of the lexeme in the dictionary.

Formulas

When Toolbox produces both valid and invalid parses for a word because of limitations in its basic parsing features, it can use word formulas to select the valid parses automatically.

Choose Formulas to access the Word Formulas dialog box. From there, you can modify individual formulas by accessing the Word Formula Properties dialog box. Toolbox will use the formulas to match the valid parses for the current word and update the list.

Enable word formulas in the Parse dialog box. It also has a Formulas button.

See also: Selecting valid parses and Troubleshooting

Parsing: Preventing Invalid Parses

If "Longest affix override" is set, the Toolbox morphological parser gives higher priority to parses that contain longer affixes than to other potential parses that contain shorter ones. In many cases, this policy reduces false ambiguity and reduces the number of times that you must select from a list of multiple parses in the Ambiguity Selection dialog box. In other cases, it causes problems because the correct parse contains a shorter affix.

When the "longest" or "greedy" match principle prevents Toolbox from producing the correct parse, you can add "direct parses" to the lexical database using the \a and \u fields.

Example 1

Using the following three records from a lexical database file, Toolbox can produce an invalid parse because there are two surface forms for the inflectional affix -s that have different lengths.

```
\lx hop
\ps v; n
```
In situations like this, you can add a "direct parse" for an inflected form of the word. It overrides the invalid analysis, because it's longer than the individual morphemes.

Example 2

Toolbox can also produce an invalid parse of a stem by matching the wrong prefix because it is longer than the right one.

Even though all the right morphemes are in the lexical database, you must provide a "direct parse" of the stem. Because the stem demist is longer than the prefix demi-, Toolbox automatically selects the parse that contains it.
Example 3

When a word or stem is truly ambiguous but the morpheme breaks occur in different places in its different parses, Toolbox may omit parses containing shorter affixes. Also, Toolbox does not normally offer parses that split up a whole word or root. To make Toolbox recognize the ambiguity and produce all the valid parses, you must provide "direct parses" of equal length.

When there are multiple parses, Toolbox lists them in the Ambiguity Selection dialog box so that you can select the correct one (i.e. the right analysis of the word in its context).
Morphophonemics

Simple morphophonemic alterations (changes) can be expressed by using alternate forms and underlying forms. The following orthographic rules:

\[ y \rightarrow i \] / \_ +ed or \[ y + ed \rightarrow ied \]

\[ y \rightarrow ie \] / \_ +s or \[ y + s \rightarrow ies \]

would be expressed as follows:

\[ \text{\_lx -ed} \]
\[ \text{\_a -d} \]
\[ \text{\_a -ied} \]
\[ \text{\_u y+ed} \]
\[ \text{\_ps vinfl} \]
\[ \text{\_ge PAST} \]

\[ \text{\_lx -s} \]
\[ \text{\_a -es} \]
\[ \text{\_a -ies} \]
\[ \text{\_u y+s} \]
\[ \text{\_ps vinfl} \]
\[ \text{\_ge 3S} \]

The way this works is that the alternate form contains the surface form including the whole of the suffix. The underlying form field contains the underlying form of the part of the root (or preceding suffix) that is modified followed by + and the underlying form of the suffix.

This is how forms of the verb try and tie are parsed:

\[ \text{\_t tried} \]
\[ \text{\_m try -ed} \]
\[ \text{\_g attempt } -\text{PAST} \]
\[ \text{\_p v} -\text{vinfl} \]

\[ \text{\_t tied} \]
\[ \text{\_m tie -ed} \]
\[ \text{\_g bind } -\text{PAST} \]
\[ \text{\_p v} -\text{vinfl} \]

\[ \text{\_t tries} \]
\[ \text{\_m try -s} \]
\[ \text{\_g attempt } -\text{3S} \]
\[ \text{\_p v} -\text{vinfl} \]
ties
m tie -s
g bind -3S
p v -vinfl

The following is an example of doubling the consonants before the -ed suffix:

lx -ed
a -d
a -pped
u p+ed
ps vinfl
ge PAST

The entry above would cause the word hopped to parse:

<table>
<thead>
<tr>
<th>t</th>
<th>hopped</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>hop -ed</td>
</tr>
<tr>
<td>g</td>
<td>jump -PAST</td>
</tr>
<tr>
<td>p</td>
<td>v -vinfl</td>
</tr>
</tbody>
</table>

The approach is equivalent for prefixes. There are very few processes involved in English prefixes, but if we suppose the following rule:

\[ sp \rightarrow p / \text{dis} + \_ \_ \text{or } \text{dis} + sp \rightarrow \text{disp} \]

as exemplified by the word dispirited was productive, then we would specify it like this:

<table>
<thead>
<tr>
<th>lx</th>
<th>dis</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>disp-</td>
</tr>
<tr>
<td>u</td>
<td>dis+sp</td>
</tr>
<tr>
<td>ps</td>
<td>neg</td>
</tr>
<tr>
<td>ge</td>
<td>OPPOS</td>
</tr>
</tbody>
</table>

This would cause dispirited to be parsed as follows:

<table>
<thead>
<tr>
<th>t</th>
<th>dispirited</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>dis- spirit -ed</td>
</tr>
<tr>
<td>g</td>
<td>OPPOS- vitality -possessing</td>
</tr>
<tr>
<td>p</td>
<td>neg- n -nadzr</td>
</tr>
</tbody>
</table>

Note: *In this version there is no way of specifying general rules in doubling consonants, so this parsing information would have to be provided for each suffix and each consonant that gets doubled.

**Context Sensitivity Preventing Incorrect Parses**

Even when there is no change to the root involved, it may still be beneficial to include part of the root in the alternate form of an affix if the affix is phonologically conditioned. This will help reduce the number of false parses.
For example, with the allomorphs of the prefix in- specified as follows:

\lx in-
\la im-
\la il-
\la ir-
\ps neg
\ge OPPOS

words like image, imam, iris and iron will be parsed incorrectly if they do not already have their own lexical entry. For example:

\tt iron
\tm in- on
\tg OPPOS- upon
\tp neg- prep

If morphophonemic notation is used to express the restriction on their occurrence then these false parses will not occur. For example:

\lx in-
\la imb-
\lu in+b
\la imm-
\lu in+m
\la imp-
\lu in+p
\la ill-
\lu in+l
\la irr-
\lu in+r
\ps neg
\ge OPPOS

Similarly, false ambiguities can be avoided by appropriately constraining allomorphs. For example, with unconstrained allomorphs, the following entry

\lx -s
\a -es
\a -es
\u e+s
\ps vinf
\ge PL

would cause words like hopes to be parsed as hop-s. But the -es suffix is phonologically conditioned. The entry should be modified as follows:

\lx -s
\a -ses
The above would solve the problem of \texttt{hop\texttt{s}} being falsely analyzed as \texttt{hop\texttt{-s}}. Since the conditioning environment for \texttt{-es} is not present, the word would be correctly analyzed as \texttt{hope\texttt{-s}}. (The entry also contains information about the orthographic use of oes for the plural of words ending in o.)
The entry should be modified as follows:

\l x   -s
\l a   -ses
\l u   s+s
\l a   -zes
\l u   z+s
\l a   -ches
\l u   ch+s
\l a   -shes
\l u   sh+s
\l a   -xes
\l u   x+s
\l a   oes
\l u   o+s
\l ps  vinfl
\l ge  PL

The above would solve the problem of hopes being falsely analyzed as hop-s. Since the conditioning environment for -es is not present, the word would be correctly analyzed as hope-s. (The entry also contains information about the orthographic use of oes for the plural of words ending in o.)
**Ambiguity: Recognizing**

In some instances, it is helpful to use morphophonemic notation to increase the number of parses. One situation where this is needed is the case where a whole root is ambiguous with a parse that divides it into pieces.

For example, the word *number* in English has two possible meanings. It could be *number* or *numb-er*. A whole root always overrides parses that split it, so in order to see the parse *numb-er*, it is necessary to force it to appear. To do this, use an entry like the following:

\[\text{l}x \text{number} \\
\text{l}u \text{number} \\
\text{l}u \text{numb} - \text{er}\]

The two \(\text{l}u\) fields of the same length cause the parse to show them as ambiguous. Note that it is necessary to include both fields. If only the \(\text{l}u\) field with *numb-er* is given, then that will be considered the only possible parse of *number*.

Here is another way to get this same effect:

\[\text{l}x \text{number} \\
\text{l}x \text{numb} \\
\text{a} \text{number} \\
\text{l}u \text{numb} - \text{er}\]

The \(\text{a}\) and \(\text{l}u\) fields show that *number* can be parsed *numb-er*. With this method, it is not necessary to have two \(\text{l}u\) fields.

Another way to deal with this problem is to use "Interlinearize (show all parses)". If "Longer override on show all" is turned off, then "Interlinearize (show all parses)" will show both *number* and *numb-er*.

When normal interlinearize does not show the parse you want, you can use show all parses to give you more parses to choose from.

Another setting that affects the number of ambiguities shown is "Longest affix override". For example the word *does* in English has two possible meanings. It could be *do-s* or *doe-s*. (This is an orthographic ambiguity, but the solution is the same as for any other type.)

If "Longest affix override" is on, then because *-es* is the longer suffix, Toolbox will prefer the parse of *do-es* to *doe-s*. But if "Longest affix override" is off, then both parses will show as an ambiguity.

**Word Formulas (Interlinear)**

When Toolbox produces both valid and invalid parses for a word because of limitations in its basic morphological parsing features, it can use "word formulas" to select the valid parses automatically.

A common problem in Toolbox parsing is invalid sequence of morphemes (especially involving wrong homographs). You can solve it by writing "formulas" that define valid sequences of morphemes (usually in terms of part of speech categories or glosses in lexical databases).

Word formulas can eliminate invalid parses, but they don't enable any additional valid parses. Note that Toolbox doesn't have "sentence formulas" that would use a word's surrounding context to resolve ambiguities in its meaning when there are multiple valid parses.
Examples of valid and invalid parses

In English, the morpheme -s has homographs: an inflectional affix for nouns (plural) and also for verbs (third person singular present tense). To simplify these examples, a third homograph (possessive) has been left out.

1. To the Toolbox parser, a word like tigers is ambiguous even though one of the parses is invalid. The invalid parse consists of an invalid sequence of morphemes: a noun root and a verb suffix (which is the wrong homograph).

```
\t tigers  tigers
\m tiger   -s  *tiger  -s
\p n    -nsuf  *n    -vsuf
```

2. A word like bears is truly ambiguous because it has two valid parses (as a noun and as a verb). In a specific context, you must select the correct parse—Toolbox can't select it for you automatically. There are also two invalid parses.

```
\t bears  bears
\m bear   -s  *bear  -s
\p n    -nsuf  *n    -vsuf
\t bears  bears
\m *bear  -s  bear  -s
\p *v    -nsuf  v    -vsuf
```

Examples of word formulas

A word formula consists of a symbol and one or more patterns. When a formula has multiple patterns—as is usually the case—they define alternatives (i.e. the formula can either match this, or that, or these others).

1. Here's a formula that selects the valid parses for tigers and bears. You can read it as, "A word consists of either a noun root (optionally followed by a noun suffix) or a verb root (optionally followed by a verb suffix)."

Symbol: Word
Patterns: n (nsuf)
         v (vsuf)

2. The formula can be made more precise by matching the suffixes by their glosses instead of by their part of speech categories.

Symbol: Word
Patterns: n (pl)
         v (3s)

```
\t bears  bears
\m bear   -s  bear  -s
\g Ursidae-pl  hold_up-3s
\p n    -nsuf  v    -vsuf
```

Comments in word formulas
To put a comment line in a word formula, put an asterisk followed by a space at the front of the line.

**Enabling word formulas In the Parse dialog box:**

- Check the **Enable word formulas** box. The formulas are kept with the rest of the Parse properties in the database type for interlinear texts. If you disable word formulas, they aren't deleted—the parser just doesn't use them.

- Choose **Formulas** to access the Word Formulas dialog box. From there, you can modify individual formulas by accessing the Word Formula Properties dialog box. The Ambiguity Selection dialog box also has a Formulas button.
Word Formulas: Selecting valid parses

If word formulas are enabled, here’s how the Toolbox parser uses them:

1. If a word has a single parse,
   - Toolbox uses that parse whether or not it matches the formulas.

2. If a word has multiple parses, the ones whose sequences of constituent morphemes match the formulas are considered valid.
   - If there’s one valid parse, Toolbox selects it automatically.
   - If there are multiple valid parses, Toolbox lists them in the Ambiguity Selection dialog box. You must select the correct one—the right analysis of the word in its context.
   - If there are no valid parses, Toolbox lists all the parses (with an asterisk * to the left of each—indicating that they are invalid).

Example  [Example lexical database records]

Because a word like bearers has multiple parses, Toolbox uses word formulas. If they don’t account for derivational affixes, none of the parses match (not even the correct one—third in the list).

* bear{Ursidae}{n} -er{vnzr}{vsuf} -s{pl}{nsuf}
* bear{Ursidae}{n} -er{vnzr}{vsuf} -s{3s}{vsuf}
* bear{hold_up}{v} -er{vnzr}{vsuf} -s{pl}{nsuf}
* bear{hold_up}{v} -er{vnzr}{vsuf} -s{3s}{vsuf}

To match the correct parse, add another rule to the formula:

Symbol: Word
Patterns: n (pl)
         v vnzr (pl)
         v (3s)

Word Formulas: Symbols

In Toolbox word formulas, a symbol is the name for a piece of a word (e.g. NounStem). In a pattern, it refers to the set of morpheme sequences that its formula matches.

Examples of symbols

1. When you add a pattern to account for the verb nominalizer suffix, it might duplicate information about the optional noun plural suffix, as follows:

Symbol: Word
Patterns: n (pl)
         v vnzr (pl)
         v (3s)
You can eliminate such duplication by defining noun stem as a symbol. When Toolbox uses the pattern containing NounStem to match a parse, it "replaces" the symbol by any of the patterns in its formula (either n or v vnzr).

Symbol:  Word
Patterns:  NounStem (pl)
          v (3s)
Symbol:  NounStem
Patterns:  n
          v vnzr

2. You can also add symbols just to make formulas easier to understand:

Symbol:  Word
Patterns:  NounWord
          VerbWord
Symbol:  NounWord
Pattern:  NounStem (pl)
Symbol:  NounStem
Patterns:  n
          v vnzr
Symbol:  VerbWord
Pattern:  v (3s)

Distinguishing symbols from data

Notice that all the examples follow a convention:

- Word formula symbols begin with uppercase letters (e.g. Word and NounStem).
- Lexical data items begin with lowercase letters (e.g. n and v; pl and 3s).

Although Toolbox doesn't require that you follow this convention, as you develop a set of formulas it is important to distinguish:

- symbols that define valid sequences of morphemes
- lexical data items that match sets of morphemes

Word Formulas: Patterns

In Toolbox word formulas, a pattern consists of one or more

- symbols
- lexical data items (from fields in Interlinear Lookup Processes—usually part of speech categories or glosses)

A symbol represents the set of morpheme sequences that its formula matches. An item represents one or more morphemes that have it in their lexical database record.

If a symbol or item is optional, enclose it in parentheses.

When a formula has multiple patterns—as is usually the case—they define alternatives (i.e. the formula can either match this, or that, or these others).
Example
Toolbox matches the correct parse of the word *philosophizes* with the formulas using the following step-by-step hypothesis-testing process. The lexical data items that match are shown in red.

\t philosophizes
\t \m philosophy \-ize \-s
\t \g learning \-nvzr \-3s
\t \p n \-nsuf \-vsuf

1. Is *philosophizes* a Word?
   Symbol: Word
   Patterns: NounWord VerbWord

2. Is *philosophizes* a VerbWord?
   Symbol: VerbWord
   Pattern: VerbStem (3s)

3. Is *philosophize* a VerbStem?
   Does *-s* have item 3s? Yes, it's a gloss.
   Symbol: VerbStem
   Patterns: v
           n nvzr

4. Does *philosophy* have item n? Yes, it's the part of speech.
   Does *-ize* have item nvzr? Yes, it's a gloss.

The parse matches the word formulas because
• all the symbols have been "replaced by" lexical data items
• for each morpheme in the word, either the \g or \p field contains an item that matches the formulas—with nothing left over

Relationship to formal grammars
Toolbox word formulas define valid sequences of morphemes using a formalism called a phrase structure grammar (also known as a context-free grammar).

Here are some alternate terms for elements of word formulas:
• rewriting rule or production corresponds to a symbol and one of its patterns (e.g. a VerbStem consists of a verb root, optionally followed by the third person singular present tense suffix)
• start symbol is the primary formula's symbol (by default, Word)
• nonterminal symbol is a word formula's symbol (e.g. VerbStem)
• terminal symbol is an item from a data field in an Interlinear Lookup Process—usually a part of speech category or a gloss (e.g. n, nvzr, and 3s)
Word Formulas: Recursion

Repeated use of word formulas to match a parse is a specific instance of a concept called "recursion". Derivational morphology is best described by recursive formulas. For example, the following words give an indication of a recursive relationship:

- **compute** is a verb
- **computer** is a noun
- **computerize** is a verb
- **computerization** is a noun

Example

In English, the morpheme -er has homographs: a derivational suffix for verbs (nominalizer) and a suffix for adjectives (comparative degree).

To the Toolbox parser, a word like **computerization** is ambiguous even though only one parse is valid (the one that contains the nominalizer, shown in red).

```
\t computerization
\m compute -er -ize -ation
\g calculate -vnzr -nvzr -vnzr
\p v -vsuf -nsuf -vsuf
```

In the following formulas, **NounStem** and **VerbStem** are mutually recursive:

Symbol: Word
Patterns: NounWord
         VerbWord
Symbol: NounWord
Pattern: NounStem (pl)
Symbol: **NounStem**
Patterns: n
         VerbStem vnzr
Symbol: VerbWord
Pattern: VerbStem (3s)
Symbol: **VerbStem**
Patterns: v
         NounStem nvzr

Toolbox matches the correct parse of **computerization** with the formulas using the following step-by-step hypothesis-testing process. The lexical data items that match are shown in red.

```
\t computerization
\m compute -er -ize -ation
\g calculate -vnzr -nvzr -vnzr
\p v -vsuf -nsuf -vsuf
```
1. Is computerization a Word?
2. Is computerization a NounWord?
3. Is computerization a NounStem?
   Assume that the optional noun plural suffix doesn't occur.
4. Is computerize a VerbStem?
   Does -ation have item vnzr? Yes, it's a gloss.
5. Is computer a NounStem?
   Does -ize have item nvzr? Yes, it's a gloss.
6. Is compute a VerbStem?
   Does -er have item vnzr? Yes, it's a gloss.
7. Does compute have item v? Yes, it's the part of speech.

Endless Recursion
This problem occurs if a symbol is used recursively in a pattern that doesn't contain at least one lexical data item. If the pattern doesn't match at least one morpheme in the parse, the recursion might never end.

Notice that the two recursive patterns from the previous example do contain items—shown in red:

VerbStem vnzr
NounStem nvzr

Avoiding needless complexity
To avoid writing formulas that are more complicated than necessary, remember that Toolbox uses word formulas for automatic selection of valid parses.

- The formulas don't have to describe the morphology of words that don't have invalid parses.
- The Toolbox parser was designed for text that is assumed to be correct—because it has usually been elicited from fluent vernacular language speakers. Therefore, the formulas don't have to exclude parses of invalid words.

Word Formulas: Troubleshooting
In Toolbox parsing, there are two main places to find the cause for problems:

- lexical databases (missing or incorrect information)
- word formulas (matching invalid parses or eliminating valid parses)

To see if it's the formulas, use the version of the Interlinearize command that shows all parses in the Ambiguity Selection dialog box. In the list of parses, the invalid ones—that don't match the formulas—have an asterisk * to the left.

1. If Toolbox doesn't produce the correct parse for a word:
   - Enable it by adding or modifying information in the lexical database (e.g. missing morphemes or morphophonemic changes).
   - If it was mistakenly eliminated by the formulas, modify them so that it matches.
2. If Toolbox produces an invalid parse:
   - Modify the information in the lexical database that allowed the parser to produce it.
   - Modify the word formulas so that it doesn't match. However, be sure that its sequence of morphemes is truly invalid in all contexts (e.g. combining a noun root with a verb suffix is invalid).
Spelling and Interlinear Check (Checks menu)

This command allows you to check for spelling errors or inconsistencies within the various fields assigned to a specific language by using the first interlinear process defined in the Interlinear tab (Properties command, Database menu). Toolbox checks every field whose language encoding is the same as the "From" marker specified in the first interlinear process.

Note: You do not have to interlinearize the text to do spell check, but you must have at least one process set up so Toolbox knows which language to use, and which dictionaries, etc. Spell check runs on any field whose language is appropriate, not just the field specified in the interlinear process. If you have interlinearized the text, the Spell Check command will run the Verify process.

To check spelling

Do the following:

1. In order for spell check to work, you must set up an interlinear process.
2. From the Checks menu, choose Spelling and Interlinear Check or use the shortcut keys, ALT+C.

Spell check starts with the word at the insertion point and checks from there to the end of the record. If text is selected, it checks only the selected text. If a word does not parse, spell checking stops and the failed word is selected. To ignore the failure, press a right arrow (this moves off the word and clears the selection), and start spell check again. To correct the word, either type the correct word in place of the selected word, or place the insertion point at the place that needs to be changed and correct it.

Jump Insert can be used to add the word to the database (Ctrl+right click in the word or choose Jump Insert from the Edit menu).

To add the root of the word to the database, select (or highlight) the root, and use Jump Insert. If no spelling problems are found, a message appears saying that spell check is complete.

To check multiple records in a database quickly, use the shortcut keys, ALT+N to move to the next record and ALT+C to spell check the record.

Interlinear Check

In interlinear text, spell check performs Interlinear Verify to verify the acceptability of all interlinear annotations.

Verify Interlinear Text

This command allows you to verify the consistency of your interlinear text without having to choose among all the ambiguities again. It also allows you to correct or update the interlinear text according to changes that have been made in the lexicon.

To Verify interlinear text

1. Go to the very top of the file (click the First Record button: )
2. From the Checks menu, choose Spelling and Interlinear Check or use the shortcut keys, ALT+C. If the text has been interlinearized, Toolbox will do interlinear Verify instead of the spell check.

3. When Verify stops, give the interlinearize command to process the word.

4. Give the Spell Check command again to continue checking / verifying the text. (Note: If you wish to stop verifying, don't give the command to continue.) **What Verify does...**

Verify does three passes through the text. The first two passes are primarily for those still working on the analysis of the language.

**Pass one**

On the first pass, **if interlinearization of a word**

* succeeded before:
  - if there are no changes to the lexicon or word formulas that affect the parsing of the word, Verify will pass on to process the next word
  - if a new gloss has been added to the lexicon that might be ambiguous for this word, but the existing glosses and parsing still succeed, Verify will pass on to process the next word
  - if the only gloss for some part of this word has been changed then depending on the setting of the **Automatic Interlinear Update** option, Verify will
    - (if the box is not checked) stop and allow you to interlinearize the word
    - (if the box is checked) replace the old gloss with the new and pass on to process the next word
  - if the parsing of the word no longer succeeds or if it produces a different result than before, Verify will stop and allow you to interlinearize the word.

* if interlinearization of a word failed before:
  - if the failed word can now be analyzed successfully, Verify will stop and allow you to interlinearize the word.

**Pass two**

The second pass allows you to update failures for which there may be a better guess than before on the root or affixes. It stops at any failure for which the guess will change if you interlinearize again. You can interlinearize the word if you want to see the changed guess.

You can tell you are in the second pass if it stops at a failure and interlinearizing produces a different, but still failed parse.

**Automatic update of changed failures**

If Automatic Interlinear Update is on, then you can request automatic update of all changed failures.

To do this, during pass two, request Spell Check again while a previous failure is highlighted. All further failures in pass two will be updated automatically. You will see each one flash briefly on the screen as it is being updated.
Pass three

The third pass is primarily for those who are preparing the text for publication. Verify will stop at every failure and will refuse to go on until you either repair the failure or place the cursor in the next bundle. If you thought your text was completely interlinearized, you will want to interlinearize these failures, possibly adding new entries to the lexicon. If you are relatively early in the process of interlinearizing, you do not have to continue verifying and interlinearizing at this point if there are failures in the text that you cannot resolve.

You can tell you are in the third pass if Verify stops at a failure and interlinearizing the word produces exactly the same result. If you choose to leave it as a failure, when you press ALT-C again, it stops at the same failure.

Progress display in status bar

During interlinear check, the status bar shows the number of the record currently being checked to indicate progress.

When a given pass is completed, the record number will return to one and will start to count up again on the next pass.

"Spell Check Complete"

If you start Verify at the top of a text, and it produces the Spell Check Complete message, it means there are no failures in your text. If you did not start at the beginning of the text, there may be failures before the point at which you began.

Adaptation: An Overview

In Toolbox, adaptation refers to the conversion of text between related languages (from a source to a target). You can use its interlinear features for adaptation. You are not limited to word-for-word translations, because Toolbox can give glosses to phrases and change the order of morphemes and words.

There are two ways to do adaptation with Toolbox. One way is for a single person to do interlinearization and adaptation at the same time. With this method, you start from the source text and either substitute whole words or analyze the words into morphemes and substitute morphemes. The other way is for one person to do interlinearization, and another person to do adaptation from the interlinearized text. With this method, a person who understands the source language can break the text into morphemes and gloss it. Then a person who understands the target language can see the morpheme breaks and glosses as helps in understanding the source text.

The second approach tends to be more work because it takes more effort to interlinearize and gloss a text than just to adapt it to another language. But the gain from the extra work is that it is shared by two people, one who analyzes the source, and another who adapts to the target. This second approach helps a person who does not know the source language to understand the source text in order to adapt it.

Four of the Interlinear processes can be used for adaptation:

- The Given process is for the interlinear lines of a text that has already been interlinearized by someone else.
- The Lookup process can convert words or morphemes from the source to the target language.
- The Rearrange process changes the order of morphemes in a word and words in a phrase.
The Generate process uses a set of phonological rules to convert from underlying forms to
surface forms (in the target language). It is usually the last Interlinear process.

**Adapt (Tools menu)**
The adapt command performs interlinear processes starting with the first adaptation process.
Adaptation usually does the following:

1. A Lookup process from the source gloss or morpheme to the target.
2. A Rearrange process to get a better target order.
3. A Generate process to form target words.

The adapt command is normally done on text that has already been interlinearized. If the adapt
command is done on text that has not been interlinearized, then if there are interlinear processes
before the adapt processes, the interlinear processes are performed before the adapt processes. This
is done so that the interlinear processes can produce the lines needed for adaptation.

**To adapt *** (Check wording, found some missing in
following.)**
Do one of the following:

- On the toolbar, click the Interlinearize button: 
  (When an Adapt process is defined for
  a particular database type, the Interlinear button on the toolbar actually functions as if it were an
  Adapt button.) □ Use the shortcut keys, ALT+A.
- From the Tools menu, choose Adapt. There is no dialog box associated with this command. If
  you have both interlinear and adapt processes defined, and no interlinearizing has yet been
done on the text, then the Interlinear button or the Adapt command will cause both the interlinear
and the adapt processes to occur.

**Adapt All**
Sometimes in adaptation you want to process a whole file at once. This is possible because the
Rearrange and Generate processes do not produce ambiguities. The Lookup process of an adaptation
setup may produce ambiguities, but in adaptation one often tries to keep the number of those to a
minimum by carefully disambiguating the analysis and being careful to have unique glosses in the
analysis linked to entries in the target dictionary. If your adaptation has few or no ambiguities, then
you can use the Adapt All command to perform the adaptation processes on the entire file. If there are
ambiguities, it will stop at each one for you to choose, then proceed on. Typically you would Adapt All
if you have made changes to the Rearrange or Generate process that might change the result in a
number of places. To do Adapt All, go to the very top of the file, and then choose Tools, Adapt All.

**Jump Target (View menu)**
When making repeated jumps to the same field in a database, it might become quite tedious for
duplicate windows to constantly be opened on the database. Unless you are cautious to close these
windows when you are finished with them, layers of duplicate windows could soon result in a cluttered
display within the Toolbox window. This is especially true when interlinearizing text.

This Toolbox feature allows you to "target" the active database window as the destination for the
Jump To command.

**Without the Jump Target set**
Toolbox moves to the record matching the jump text by opening a duplicate window on the database positioned at that particular record.

**With the Jump Target set**
Toolbox moves to the record matching the jump text by positioning the database in the target window at that particular record.

**To set the Jump Target**
Do the following:

1. First sort the active database window by the marker specified in the Jump Path. Do not include any other marker in the sort.*

2. From the View menu, choose Jump Target. When enabled a check mark appears beside the command name. Only the active database window is affected.

**To turn the Jump Target setting on or off**
From the View menu, choose Jump Target. When enabled a check mark appears beside the command name. To disable, choose the command again and the check will disappear. Only the active database window is affected.

**Note:**
*The duplicate window opened during a jump is automatically sorted only on the marker matched in the jump path. Therefore, if you target the window opened as a result of the jump, it will become the destination of future jumps matching that particular marker and database.

Remember, if a database is sorted by any field other than the record marker field, that field and its contents are displayed on the status bar.

**Spelling and Interlinear Check (Checks menu)**
This command allows you to check for spelling errors or inconsistencies within the various fields assigned to a specific language by using the first interlinear process defined in the Interlinear tab (Properties command, Database menu). Toolbox checks every field whose language encoding is the same as the "From" marker specified in the first interlinear process.

Note: You do not have to interlinearize the text to do spell check, but you must have at least one process set up so Toolbox knows which language to use, and which dictionaries, etc. Spell check runs on any field whose language is appropriate, not just the field specified in the interlinear process. If you have interlinearized the text, the Spell Check command will run the Verify process.

**To check spelling**
Do the following:

In order for spell check to work, you must set up an interlinear process.

From the Checks menu, choose Spelling and Interlinear Check or use the shortcut keys, ALT+C.
Spell check starts with the word at the insertion point and checks from there to the end of the record. If text is selected, it checks only the selected text. If a word does not parse, spell checking stops and the failed word is selected. To ignore the failure, press a right arrow (this moves off the word and clears the selection), and start spell check again. To correct the word, either type the correct word in place of the selected word, or place the insertion point at the place that needs to be changed and correct it. Jump Insert can be used to add the word to the database (Ctrl+right click in the word or choose Jump Insert from the Edit menu).

To add the root of the word to the database, select (or highlight) the root, and use Jump Insert. If no spelling problems are found, a message appears saying that spell check is complete.

To check multiple records in a database quickly, use the shortcut keys, ALT+N to move to the next record and ALT+C to spell check the record.

**Interlinear Check**
In interlinear text, spell check performs Interlinear Verify to verify the acceptability of all interlinear annotations.

My Questions:
Does the full interlinear have to be set up to parse correctly (assuming that whole inflected words are not in the dictionary as such)?

If a working interlinear is not needed, then:
How does it know what to do with running, when only run and -ing are in the dictionary?

How does it know that parsing is OK if only parse and -ing are in the dictionary? – or does it?

How does it know that thoughting is incorrect?

**Technical Details**

**Sorting: An Overview**
Toolbox allows you to establish the proper order for sorting data for each language. Toolbox is also able to sort data in a variety of ways. Toolbox can sort the records in a database by different fields; it also can sort from the end of the word, as well as by generalizations such as Consonant vs Vowel. The default phonetic sort order (which is also used as the default vernacular sort order) has the option of sorting “alphabetically” or by place of articulation or manner of articulation. Non-roman scripts and syllable-based sorting have been successfully accommodated.

Sorting by various fields and specifying the details of the sort order are two quite different topics. Typically the sort order for a language is set up once and then is not modified. Changing the fields or direction of sorting is done frequently, sometimes many times in the same day.

Specifying how to change the fields or direction of sorting is discussed under Data, Sorting. The details of establishing the sort order is discussed below.
### Setting up the Sort Order

**Terminology: “Primary” and “Secondary”**

*Primary sort characters* are characters which define the main divisions in sorting. They are the “alphabet” in a roman script. In a dictionary, generally all words beginning with the same primary character are grouped together. Within that list, the words are further sorted according to the proper order of the alphabet for the second character. (Sorting for some languages, with overstriking vowels or syllable-ordered sorting gets much more complex, but think of languages using roman script.)

*Secondary sort characters* are characters that are mostly ignored except in the case of minimal pairs, and words that begin as if they were minimal pairs. In English, these are quite rare, but hyphen and apostrophe can illustrate. Does “we’ll” come before “well” or after? What about co-op (a verbal shortening of “cooperative”)? Does it come before or after “coop”?

The concept of secondary sorting characters is very important for establishing the sort order in the Language Encoding. Occasionally, if the script being used is inadequate to express the full number of vowels or consonants in the vernacular language, a (linguistically) primary character that does not have an independent alphabetic symbol in the script is represented using an overstrike or is combined with another character (eg ç, ñ, ö, ch, ng). Such characters are often given primary character status in the sorting but are sometimes sorted by the national orthography as closely as possible.

Another relevant factor is how the data was keyed – that is, what the underlying codes are. Unicode allows two forms for many characters (sometimes more). Generally speaking, an overstruck character, such as ñ can be formed (and encoded) as an “n” with a tilde overstriking – two separate Unicode codes – or as a single picture which shows the ñ already combined. The code for ñ has nothing to do with the code for n or for tilde. So you must know what is in the data or allow for both possible forms.

If you are establishing the keyboarding, you may wonder if there is any reason to choose one form over another? Yes. If a vowel (or consonant) with an overstrike is really the same character as the form without the overstrike – perhaps the overstrike represents tone, for example –, then having the overstrike as a separate character will make some searches simpler. But if it’s an entirely different character, as n and ñ usually are, then having the combined form will prevent some accidental “finds” since there is no separate “n” to be found in the ñ code.

### Sort Orders tab

This tab allows you to define the sort orders for the language encoding. The options provided in this tab include:

- **Add button** Adds a new sort order.
- **Copy button** Copies existing sort order to a new name.
- **Modify button** Modifies existing sort order currently highlighted. Double-clicking on any sort order within the list is the same as choosing the Modify option.
- **Delete button** Removes the highlighted sort order from the list making it unavailable.
- **Default sort order** Specifies the language encoding which contains the sort order to be used for sorting the secondary field(s).
The default sort order is an important option as the Interlinearizer usually uses the default sort order to determine which characters are word forming characters. It assumes that anything not in the default sort order of the language is punctuation or other word separators. Secondary characters and ignore characters are considered part of the sort order.

Choosing Add, Copy or Modify opens the Sort Order Properties dialog box, where sort orders are defined or modified.

**Sort Order Properties dialog box**

This dialog box is where sort orders are defined and modified. It offers the following options:

**Name**

Specifies the name of a sort order—more than one sort order can be defined to use for the language.

**Description**

Describes the sort order and how it functions. This description provides a way for the user to distinguish between the various sort orders that might be defined.

**Primary characters**

Lists the orthographic characters considered most important or primary in the sort. These are distinct characters like A and B, and they specify the order for the sorting.

Characters which are identical in the sort order are entered on the same line as the primary character they are identical to and are separated by one or more spaces. The order of characters within this grouping is significant—it is referred to as the secondary grouping.

For example:

- A a
- B b
- C Ç c ç
- Ch ch
- D d

For “A a” and “B b”, this says that capitalization should be considered for minimal pairs (i.e., words which differ only by their capitalization), but otherwise A and a should be treated the same as each other, and B and b should be treated the same as each other. “Ba” will follow “az”

In the case of “C Ç c ç”, the C or c with a cedilla is to be treated like the C or c without. However, if minimal pairs occur, then a capitalized word is sorted before a lower-case word, whether or not the cedilla is present. If both words are the same case, then the word without the cedilla is first.

The lack of space between the c and the h indicates that Ch and ch are to be treated as single characters for sorting and words beginning with Ch occur after words beginning with cz and before words beginning with Da. This is true even though both c and h occur separately as Primary sort characters.

**Secondary Characters**

Secondary characters are often overstrikes. When that is the case, you can choose whether to put the secondary form in the same line as the Primary, as the following:
A Á a á
or to put only the A and a in the Primary order and list the overstrikes in the Secondary. Where there
are a lot of overstrikes used, it is generally a good idea to list the overstrikes rather than trying to get all
the secondary forms included in the Primary. It is too easy to miss one, or to specify a different order
for the secondary forms of A than of U. (I’ve seen both problems.)

Note, if you only have an acute accent over i and u, for example, it doesn’t hurt that you have allowed
for it occurring over a, e, and o by listing it as a Secondary. In general, characters that never occur in
the data are not a problem in the Sort Order.

If you have combination forms, as mentioned earlier, such that the acute accent over the a is not a
separate character but the a-acute is a single picture and a single Unicode code, then you have no
choice but to list the combined form as a secondary in the Primary order.

If you are not sure, enter both combined and separate characters.

Note: When entering isolated overstriking diacritics into the list of secondary or ignore characters (or
even into the primary characters list) they should be preceded by two spaces. A single space may
leave them appearing to overstrike their neighbor and might even conceal them.

Secondary characters ordered before unmarked primaries
Secondary characters specified within this option are sorted before unmarked primaries. For
example, in English with ' (apostrophe) defined in this option, the following list might occur:

she’ll with secondary character
shell unmarked primary
we’d with secondary character
wed unmarked primary

Compare this list to the same set of words under “Secondary characters ordered after”, following.

Secondary characters ordered after
Secondary characters specified in this option are sorted after unmarked primaries. For example, in
English with ' (apostrophe) defined in this option, the following list might occur:

shell unmarked primary
she’ll with secondary character
wed unmarked primary
we’d with secondary character

Compare this list to the same set of words under “Secondary characters ordered before”, above.

Secondary characters follow their corresponding primaries
This option allows you to specify how the secondary characters actually appear in the database file.
Toolbox assumes that secondary characters precede their corresponding primaries unless this
option is chosen. For example, if you have an acute-a – is the a first in the data or the acute?
Ignore characters
Lists the characters which will be ignored in the sort sequence. When entering, remember characters are separated by spaces. For example:

- ' =

Using CC with the Sort Order
Some sorting requirements are just too complex to lay out as combinations of Primary and Secondary characters. This is particularly true for syllable-based sorting schemes. At the moment, there is no documentation for this process and there is no edit box for specifying such a table. However, if you have the MDF files that are included in the Toolbox New Project, you can see that the Language Encoding "Date" has a CC table specified. If you are familiar with CC, then examining this table and locating it in the Date.lng file will give you clues about how to approach this problem. Help is also available from Toolbox Support – contact Toolbox@sil.org

Case and Punctuation tab
Case and Punctuation are mostly related to Interlinear.

Punctuation
Punctuation is an alternative to having a Sort Order, mostly for eastern scripts with a "gazillion" characters (Korean, Japanese, and Chinese particularly). If there is something in the Punctuation box, any Sort Order will be ignored and the Unicode code-order will be used instead.

But it also has implications for Interlinearizing. When interlinear is given a text to parse, there is often punctuation included with the text. How can Toolbox identify which codes refer to a word and which are punctuation? Mostly Toolbox consults the Sort Order to determine if a character belongs to a word. But for some scripts, it’s nearly impossible to list everything. So the Punctuation box provides an alternative: list everything that is not part of a word instead.

If anything is placed in the Punctuation box, the Sort Order will be ignored for determining word-building characters for Interlinear. That means that, for English, you have to list something like the following:

.,?!:();+-={}

plus both opening and closing single and double quotation marks. It's extremely easy to miss something, and I make no guarantees that this list is complete. (Look at all the punctuation above the digits that I didn’t include... are any in your text?)
You need to be sure that you include the characters that Interlinear uses for the forced glosses when it's parsing. Those characters are (by default) the curly brackets. These can be changed in the Parse Properties dialog box. See the Advanced Topics, Database Types, Interlinear Setup.

**Case**

Text to be interlinearized also can contain capitalization if the script allows for such. If the script you are working with does not have special upper case forms, then you don’t need to have anything in this list. For pure English, this can just be cut-and-pasted from the Sort Order. But if there are any diacritics, then the relationships need to be clarified for Interlinear. (Toolbox can’t see the character shapes.) So if you sort order has

\[
A \quad \tilde{A} \quad a \quad \tilde{a}
\]

the Case list needs to have

\[
A \quad a \quad \tilde{A} \quad \tilde{a}
\]

so Interlinear will know how to deal with \(A\) or \(\tilde{A}\) when it finds it.

**Note:** Some orthography conventions don’t put accents on upper case characters. This is a situation that Interlinear does not handle directly. The Case box is expecting matched **pairs**. You cannot do three:

\[
A \quad a \quad \tilde{a}
\]

Nor can you match plain cap \(A\) with two different lower case forms on separate lines.

\[
A \quad a
\]

In this latter situation, Toolbox will give you an error message for duplicating the \(A\). In the former situation, Toolbox ignores anything past the pair but no error message is given; it just doesn't do what you want.

The solutions are to quickly scan the beginnings of sentences, or to enter words which occur frequently at the beginnings of sentences as an alternate form (thus creating an ambiguity).