User's Guide
to the
GloTEX Program

Version 1.1

Abstract

The GloTEX program is used to automate the generation of a Glossary in a LATEX document. It uses the `.glo` file generated by the `\makeglossary` command and one or more Glossary Definition Database Files to create a file which is `\input` in the document to generate the Glossary.

R L Aurbach
CR&DS MIS Group
Monsanto Company
800 N. Lindbergh Blvd.
St. Louis, MO 63167
(314)–694–5453

August 27, 1986

[File: GloTeX.QMS]

User's Guide
to the
GloTEX Program

Why Automate Glossaries?

Several programs already exist to augment the capabilities of LATEX to produce better documents — BibTEX supports the formatting and maintenance of bibliographies, IdxTEX automates the generation of an Index, etc. Why is a program needed to automate the generation of Glossaries? There are several reasons:

1. Consider how you might actually make a glossary in a real document.

    - As you write the document, you come across words or phrases which should be included in a glossary. You jot them down on paper somewhere.
    - When the document is nearly finished, you take your list of terms, alphabetize it, and write definitions for each one.
    - Then you write these definitions into your document and complete it.

    Think how much easier it would be if you could completely automate documents with glossaries:

    - As you write the document and come across a word or phrase to be included in the glossary, you place a simple `\glossary` command into the document.
    - You format the document with LATEX and use GloTEX to build the glossary. The GloTEX program will tell you about any terms you haven't yet defined.
    - The definitions go into an external database, rather than directly into the document. This means that you can use them again in another document without retyping them.

    So GloTEX can be used to make the generation of a document which contains a glossary easier.

2. Suppose you are writing a series of documents about related topics and you decide that each should have a glossary. You could carefully write the definitions for each term into each document.

But, wouldn't it be easier to write the definitions only once and pull those definitions which are needed into each document? If you often write about a certain topic, wouldn't it be easier to create a **database** of definitions of those terms you commonly use?

GloTEX helps you maintain a database of definitions for inclusion in glossaries. It supports the use of multiple definition files in a single document, so you can keep definitions for different terms in different places. For example, if you are writing about an application program, you might want to have one glossary of computer terms and another which describes the vocabulary appropriate to that application.

3. The standard LATEX document styles already support the most important features needed to automate glossaries — the `\makeglossary` and `\glossary` commands. Before GloTEX, there was nothing available to allow you to take advantage of these LATEX features.

Building a Glossary — Step by Step

To automate the generation of a Glossary, here is what you do:

- In the preamble of your document, include the command `\makeglossary`. This will cause a file to be generated when you run LATEX which has the same filename as your LATEX file and the file type `.glo`.

- In your document, for every word or phrase which you wish to appear in the glossary, include a `\glossary` command. The command format is

  ```
  \glossary{label}
  ```

  where "label" is an identifier which is compared with the contents of the Glossary Definition File to locate the correct definition. This will be discussed in more detail later.

- If the definitions you wish to include are not already in a Glossary Definition File, create the appropriate file. This is discussed below.

- Run LATEX on you document. This produces the `.glo` file.

- Run GloTEX on the `.glo` file, specifying the name of one or more Glossary Definition Files from which definitions are to be drawn. GloTEX will produce a file which has the same filename as you LATEX file, but with the file type `.gls`.

- In your document, insert an `\input` command, such as

  ```
  \input{myfile.gls}
  ```

  to include the `.gls` file in your document at the appropriate place.

- Run LaTeX twice. The second time is needed to get the Table of Contents entry for the Glossary to appear properly.

Command Syntax

The GloTeX program is run as a foreign DCL command.[1]

The GloTeX program is run using a command line of the form

```
$ glotex  file [/STYLE={Article | Report | Special}]
                                        [/GLOSSARY=(def1[,def2[,...]])]
```

The "file" parameter is required. It is the name of the `.glo` file which is GloTeX's input file. If you don't specify an input file, GloTeX will prompt you for one. You need not specify the `.glo` file type. The default file specification for the input file is file type `.glo` in your current default directory.

The `/STYLE` qualifier is optional. It describes how you wish GloTeX to format the glossary it builds. There are three possibilities:

Article The `Article` keyword is used if your document uses the standard LaTeX `article` document style or any other section oriented style. If you specify this keyword, then the glossary is built as a regular, numbered section in your document. It will appear in your Table of Contents automatically.

Report The `Report` keyword is used if your document uses the standard LaTeX `report` or `book` document styles or any other chapter oriented style. If you specify this keyword, then the glossary is built as an unnumbered chapter in your document. It will appear in your Table of Contents automatically.

[1] On the Monsanto BB1T VAXcluster, the GloTeX symbol is defined by issuing the CRLSETUP command.

Special

The `Special` keyword is used if your document uses any of the Monsanto CR&DS document styles (`pamphlet`, `manual`, or `memo`). They already define commands to format the glossary appropriately — using the `Special` keyword tells GloTEX not to include commands in its output file to redefine these commands for you.

In section-oriented Monsanto CR&DS document styles, the glossary is formatted as a numbered section. In the chapter-oriented `manual` document style, the glossary is an unnumbered chapter, with page numbers of the form

"Glossary–nn".

If you do not include the `/STYLE` qualifier, `Special` is assumed.

The `/GLOSSARY` qualifier is used to specify the list of Glossary Definition Files which GloTEX will search to find the definitions you wish to include. You may include as many files as you wish — GloTEX will search them in the order you specify. For each file, GloTEX will look for a file with the file type `.gdf` in your current default directory, unless you override the defaults.

The `/GLOSSARY` qualifier is optional. However, you must use it unless you are using a Monsanto CR&DS document style, because there is no other way for GloTEX to know where your definitions are to be found.[2]

Note

> GloTEX prints a list of all those labels which are present in your input file, but for which it can't find any definition. This report is output both on your terminal and to a Glossary Log File (`.glg` file).
>
> You can use this feature to make a list of terms needing definition.

Building a Glossary — More Detail

This section discusses the steps needed to build a Glossary in more detail. In this section, we assume that the source file for your document is named `myfile.tex`.

`makeglossary` **Command**

LATEX creates a list of those terms to be included in the Glossary in a file named (in this example) `myfile.glo`. This file is not created automatically. If you want one, you must include the `\makeglossary` command in your document.

[2] If you use a Monsanto CR&DS document style, special commands are available to allow you to build the glossary definition file specifications into your document. They are described later.

The `\makeglossary` command must be placed in the preamble of your source file — that is, between the `\documentstyle` command and the `\begin{document}` command.

`glossary` Command

When you come across a word or phrase in your document which you would like to define in the Glossary, insert a `\glossary` command in your source file. The format of this command is

```
\glossary{label}
```

where “label” is an identifier you assign to this word or phrase. This identifier is used to locate the definition in a Glossary Definition File which you wish to be included. That is, if “label” matches the identifier belonging to a definition in the database, then that definition will be included in the Glossary.

For example, in this document, the entry `\glossary{glo-file}` appears. As you can see by examining the glossary, there is a definition for the term “`glo` File”, which was generated as a result of this reference.

Generating the `glo` File

Having prepared your document, the next step is to run it through LaTeX. This will generate a file `myfile.glo` which contains the collected information from the `\glossary` commands which you placed in your document.

Building the `gls` File

The next step is to use the GloTeX program to generate the glossary information, using the `myfile.glo` file and the appropriate Glossary Definition Files.

In our example, we will make several assumptions.

1. Lets assume that your document `myfile.tex` uses the standard LaTeX `article` document style.
2. Lets also assume that you are using definitions from three sources — a local glossary of application specific terms, a public glossary of biological terms, and a public glossary of computer terms. The biological glossary is assumed to be located in the `GenLocDoc:` area, while the computer glossary is located in the `CRL_Documentation:` area.

Then, the command line

```
$ glotex myfile /STYLE=Article -
$_    /GLOSSARY=(myglos,GenLocDoc:BioDefs,-
$_               Crl_Documentation:EDPDefs)
```

will process the `myfile.glo` file, using the three specified Glossary Definition Files, and produce the file `myfile.gls`, formatted appropriately for the standard LaTeX `article` document style.

Including the `gls` File

Normally, a Glossary is placed in the document at the end, just before the index. Most people place the Glossary **before** the Bibliography, if both exist in the document, but that is a matter of personal choice.

In any case, once you decide where to put the Glossary, it is merely necessary to add the command

```
\input{myfile.gls}
```

in the appropriate place.

Generate the Final Document

Now you are ready to generate the final document. To do this, run LaTeX on the document twice. The first time you run LaTeX, the Glossary will be included in your document and a reference will be made to include a Table of Contents entry for it. The second time you run LaTeX, the Table of Contents will be updated to include an entry for the Glossary.

Glossary Definition Files

A Glossary Definition File is a text file which contains one or more definitions of words or phrases (an “entry”) in a special, defined format. A Glossary Definition File is also called a `.gdf` file, because the default file type for these files is `.gdf`.

A entry in a `.gdf` file has three parts.

label

The *label* is an identifier used to associate a particular definition with a particular `\glossary` command from the document.

The label is a **case sensitive** string. If the label of an entry in the `.gdf` file matches a label specified in a `\glossary` command exactly (character for character, including any whitespace characters), then the term will be included in the Glossary.

item

The *item* is the word or phrase being defined by the entry. Visually, the item appears on the left-hand portion of the page, separated from the definition.

Glossary entries are formatted in an `infomap` environment,[3] a special variation of the `description` environment.[4] In an `infomap` environment, the item is placed in a `\parbox`. If the item is too long for the space allotted it, it will be typeset in multiple lines. You can often help this process along by careful inclusion of `\\` commands in the item to control where the lines are split.

The default formatting in an item is **boldface**. Other possibilities are available, however. In this document, for example, the item "glo File" was entered as "`{\tt glo} File`". Notice that the term appears in the correct alphabetical order, in spite of the presence of the special formatting characters.

definition The definition is the body of the glossary entry. It consists of as much normal LaTeX source as you wish — multiple paragraphs, lists, etc., are all supported. However, experience has shown that a glossary entry which is two or three sentences long is normally plenty. Very long entries lose their effectiveness with the reader.

The format of an entry in a `.gdf` file is

```
@entry{label, item} definition
definition (continued...)
```

A new entry in the file begins with "`@entry`" in lower case, and as the first characters on a new line. The `@entry` command takes two arguments — the label and the item strings. The item string is optional — if it is not specified, then the label string will be used as the item string as well.

The definition may begin on the `@entry` line, although that is not required. All text is considered as definition, beginning with the first character after the closing brace of the `@entry` command up to the next `@entry` command or the end of the file.

As an example, below is a portion of the `.gdf` file used to generate the Glossary of this document.

```
@entry{glo-file, {\tt glo} File} The {\tt .glo} file is
produced by \LaTeX\ and contains a list of labels for
terms to be included in the Glossary.  It is one of the
inputs to \GloTeX.
```

[3] for those familiar with the CR&DS Extensions to LaTeX.

[4] If you specify /STYLE:ARTICLE or /STYLE:REPORT in your command line, GloTeX will automatically include in its output file the commands needed to define this new environment.

```
@entry{item} An {\em item\/} is the word or term to be
defined in the glossary.
```

LaTeX Extensions Which Support Glossaries

Standard LaTeX document styles define the `\makeglossary` and `\glossary` commands described above. If you are using standard LaTeX document styles, these are the only commands available to you.

If you use one of the CR&DS document styles (`pamphlet`, `manual`, or `memo`), then additional commands are defined which make glossary generation even nicer.[5]

`theglossary` **Environment**

The `theglossary` environment is the context for a glossary. That is, a glossary begins with the `\begin{theglossary}` command and ends with the `\end{theglossary}` command. Each CR&DS document style defines this environment so that the formatting of the Glossary is appropriate to the document. When you specify `/STYLE = Article` or `/STYLE = Report`, GloTeX includes in the `.gls` file the necessary commands to define this environment for you.

`glossaryfile` **Command**

The `\glossaryfile` command is used to specify one or more `.gdf` files to be used to locate definitions. The format of the command is

```
\glossaryfile{filelist}
```

where filelist is either a single file specification or a list of file specifications, separated by commas.

You may specify more than one `\glossaryfile` command if you wish. The `.gdf` files you specify are searched in the following order:

1. Any `.gdf` files specified on the command line using the `/GLOSSARY` qualifier are searched in the order specified.
2. Any `.gdf` files specified in the first `\glossaryfile` command in the source document are searched in the order specified.
3. Addition `.gdf` files are searched in order as specified by additional `\glossaryfile` commands.

[5] This section is only relevant at sites which use the Monsanto CR&DS special document styles. It should be ignored otherwise.

`insertglossary` **Command**

The `\insertglossary` command is used to place a glossary at a particular place in the document. It is equivalent to issuing the `\input{myfile.gls}` command, where `myfile` is the name of the source document.

Glossary

definition

The *definition* is the text used to indicate what a particular item means. A definition may be as short as a phrase, or as long as a dissertation. However, short definitions (a sentence or two) are usually most effective in the context of a Glossary.

`gdf` **File**

The `.gdf` file is the Glossary Definition File.

`glg` **File**

The `.glg` file is a Glossary Log File. It lists those labels for which no definition was found. This file is only generated if needed.

`glo` **File**

The `.glo` file is produced by LaTeX and contains a list of labels for terms to be included in the Glossary. It is one of the inputs to GloTeX.

Glossary Definition File

A Glossary Definition File is represents a database of definitions. Each entry in a Glossary Definition File contains a *label*, an *item*, and a *definition*.

GloTeX

The GloTeX program is a tool used to collect and format a glossary for inclusion in a LaTeX document. It uses `\glossary` commands in the LaTeX source file to determine which words or phrases should be included in the Glossary, and one or more Glossary Definition Files which contain the definitions of the terms.

`gls` **File**

The `.gls` file is the output of GloTeX. It is the file which should be `\input` in the document to generate the Glossary.

item

An *item* is the word or term to be defined in the glossary.

label

A *label* is an identifier which is associated with a particular definition. Labels are specified in `\glossary` commands to indicate which definitions should be included in the Glossary.

Index

— B —

Building a Glossary, 2ff, 4ff; *see also*
• Glossary Definition File
Create `.gdf` File, 2
Create `.glo` File, 2, 5
Create `.gls` File, 2, 5
Generate Final Document, 3, 6
`\glossary`, 2, 5
Include in Source File, 3, 6
`\makeglossary`, 2, 4

— C —

Command Syntax, 3ff
`/GLOSSARY` Qualifier, 4
`/STYLE` Qualifier, 3
`Article` keyword, 3
`Report` keyword, 3
`Special` keyword, 4
File Parameter, 3

— G —

`gdf` file, 6ff
Glossary Definition File, 6ff
Definition, 7
Example, 7
Format, 7
Item, 6
Label, 6
GloTeX, 1ff
raison d'être, 1
Complete a partial implementation, 2
Make writing easier, 1
Re-use definitions, 1

— N —

New Glossary Commands
`glossaryfile` Command, 8
`insertglossary` Command, 9
`theglossary` Environment, 8

Master Index

— **B** —

`\begin{theindex}` Command
- *IdxTeX*, 4

Building a Glossary
- *GloTeX*, 2ff, 4ff

see also
- Glossary Definition File

Create `.gdf` File
- *GloTeX*, 2

Create `.glo` File
- *GloTeX*, 2, 5

Create `.gls` File
- *GloTeX*, 2, 5

Generate Final Document
- *GloTeX*, 3, 6

`\glossary`
- *GloTeX*, 2, 5

Include in Source File
- *GloTeX*, 3, 6

`\makeglossary`
- *GloTeX*, 2, 4

— **C** —

Combining References
- *IdxTeX*, 10

see also
- Cross References

Command Syntax
- *GloTeX*, 3ff

see also
- Syntax

`/GLOSSARY` Qualifier
- *GloTeX*, 4

`/STYLE` Qualifier
- *GloTeX*, 3

`Article` keyword
- *GloTeX*, 3

`Report` keyword
- *GloTeX*, 3

`Special` keyword
- *GloTeX*, 4

File Parameter
- *GloTeX*, 3

Cross References
- *IdxTeX*, 9ff

— **E** —

`\end{theindex}` Command
- *IdxTeX*, 4

Entry
- *IdxTeX*, 4

— **G** —

`gdf` file
- *GloTeX*, 6ff

Glossary Definition File
- *GloTeX*, 6ff

Definition
- *GloTeX*, 7

Example
- *GloTeX*, 7

Format
- *GloTeX*, 7

Item
- *GloTeX*, 6

Label
- *GloTeX*, 6

GloTeX
- *GloTeX*, 1ff

raison d'être
- *GloTeX*, 1

Complete a partial implementation
- *GloTeX*, 2

Make writing easier
- *GloTeX*, 1

Re-use definitions

        • *GloTeX*, 1

— **H** —

Highlighting
    A **boldface** entry
        • *IdxTeX*, 5
    An *italic* entry
        • *IdxTeX*, 5

— **I** —

IDX File
        • *IdxTeX*, 1, 3
IdxTeX Features
    Alphabetized, Two-Column Index
        • *IdxTeX*, 2, **4**
    `\verb` Command Supported
        • *IdxTeX*, 2, **6**
    Complete Index Generation
        • *IdxTeX*, 2, **4**
    Cross References
        • *IdxTeX*, 2
    Index Item Highlighting
        • *IdxTeX*, 2, **5**
    Master Index Generation
        • *IdxTeX*, 2
    Page Number Highlighting
        • *IdxTeX*, 2, **6**
    Page Number Ranges
        • *IdxTeX*, 2
    Three Levels of Indexing
        • *IdxTeX*, 2, **4**
IND File
        • *IdxTeX*, 1, 3–4
`\index` Command
        • *IdxTeX*, 1–2, 4
    Use of `#`
        • *IdxTeX*, 9
    Use of `&`
        • *IdxTeX*, 10
    Use of `>`
        • *IdxTeX*, 4
    Use of `^`
        • *IdxTeX*, 6
    Use of `_`
        • *IdxTeX*, 6
    Use of `~`
        • *IdxTeX*, 6
`\indexentry` Command
        • *IdxTeX*, 1
`\input{file.IND}`
        • *IdxTeX*, 3

— **M** —

`\makeindex` Command
        • *IdxTeX*, 1–2
Master Index
        • *IdxTeX*, 10–12
    Definition
        • *IdxTeX*, 10
    Generation
        • *IdxTeX*, 12
    The `.MDX` File
        • *IdxTeX*, 11
    The `.MND` File
        • *IdxTeX*, 11

— **N** —

New Glossary Commands
    `glossaryfile` Command
        • *GloTeX*, 8
    `insertglossary` Command
        • *GloTeX*, 9
    `theglossary` Environment
        • *GloTeX*, 8

— **P** —

Page Numbers
    boldface
        • *IdxTeX*, **6**
    italics

- • *IdxTEX*, *6*
- underlined
  - • *IdxTEX*, <u>6</u>
- Page Ranges
  - • *IdxTEX*, 7ff
  - Explicit
    - • *IdxTEX*, 8ff
  - Implicit
    - • *IdxTEX*, 8ff
- Preamble
  - • *IdxTEX*, 1–2

— S —

- SubIndex (>) Command
  - • *IdxTEX*, 4
- Syntax
  - • *IdxTEX*, 3
  - /MASTER Qualifier
    - • *IdxTEX*, 11
  - /TOC Qualifier
    - • *IdxTEX*, 3
  - Filename Parameter
    - • *IdxTEX*, 3, 11

— T —

- theindex Environment
  - • *IdxTEX*, 2
- Top Level Item
  - SubItem
    - SubSubItem
      - • *IdxTEX*, 5

Contents

Why Automate Glossaries? 1

Building a Glossary — Step by Step 2

Command Syntax 3

Building a Glossary — More Detail 4

Glossary Definition Files 6

LaTeX **Extensions Which Support Glossaries** 8

Glossary 9

Index 10

Master Index 11