Sample Document Using the datagidx Package

Nicola L. C. Talbot

April 3, 2025

Contents

Summary

This is a sample document illustrating the use of the `datagidx` package to create document indices, glossaries and lists of acronyms without the use of external indexing applications, such as `makeindex` or `xindy`. Please read the user guide (datatool-user.pdf) for further guidance.

Chapter 1

Introduction

Words can be indexed.

A glossary is a useful addition to any technical document, although a glossary can also simply be a collection of glosses, which is another thing entirely. Some documents have multiple glossaries.

A bravo is a cry of approval (plural bravos) but a bravo can also be a hired ruffian or killer (plural bravoes).

1.1 Characters

When defining entries be careful of commas (,) and equal signs (=) so they don't interfere with the key=value mechanism. The sort and label keys get expanded so be careful of special characters, such as ampersand (&), underscore (_), circumflex (^), dollar ($) and tilde (~).

Since we're not using `makeindex`, we don't need to worry about `makeindex`'s special characters, such as double quote ("), exclamation mark (!) and vertical bar (|). (Unless they've been made active by packages such as `ngerman` or `babel`.)

Non-alphabetical characters are usually grouped at the start of an index, and are usually followed by the number group. That is, the group of entries that are numerical, such as 0 (zero), 1 (one), 2 (two) and 3 (three).

1.2 Custom Fields

You can add custom fields. For example, this document has added three custom fields to the 'index' database.

1.3 Plurals

The plural of glossary is glossaries. The plural of index is indices. Some words have an alternative plural. For example, an alternative to indices is .

1.4 Sorting

The only type of sorting available is letter sorting. If you want word sorting you'll need to use `makeindex` or `xindy`. So "sea lion" comes after "seal".

The default sort is case-insensitive so kite before Knuth and Knuth before koala.

1.5 Using without indexing

Here's a defined entry that won't get into the glossary. Name: page list. Description: a list of individual pages or page ranges (e.g. 1,2,4,7–9). (Unless I later reference it using a command like `\gls`.)

1.6 Links to Entries

You can reference and index entries using `\gls`, `\Gls`, `\glspl`, `\Glspl`, `\glssym` and `\Glssym`. (Note, if you're used to using the `glossaries` package the syntax is different.)

Or you can reference a particular field using `\useentry` or `\Useentry`. So here's the description for seal: sea mammal with flippers that eats fish.

If the `hyperref` package has been loaded, commands like `\gls` will link to the relevant entry in the glossary or index. Referencing using `\glsdispentry` and `\Glsdispentry` won't have hyperlinks.

1.6.1 Enabling and Disabling Hyperlinks

If the `hyperref` package has been loaded, hyperlinks can be enabled and disabled. Either globally (here's a reference to seal without a hyperlink and here's a reference to seal with a hyperlink) or locally (here's a reference to seal without a hyperlink and here's a reference to seal with a hyperlink).

1.7 Acronyms

Here's an acronym referenced using `\acr`: hyper-text markup language (html). And here it is again: html. If you're used to the `glossaries` package, note the difference in using `\gls`: hyper-text markup language (html). And again (no difference): hyper-text markup language (html).

Now let's switch to displaying acronyms with a footnote. First use: xml[1]. Next use: xml.

However it would look better if the footnote text started with a capital letter, so let's tweak things a bit. Try with another acronym: css[2]. Next use: css.

Reset: Here are the acronyms again: hyper-text markup language (html), extensible markup language (xml) and cascading style sheet (css). Next use: html, xml and css. Full form: hyper-text markup language (html), extensible markup language (xml) and cascading style sheet (css).

Reset again. Start with a capital. Hyper-text markup language (html). Next: Html. Full: Hyper-text markup language (html).

Prefer small-caps? Cascading style sheet (CSS). Next: CSS. Full: cascading style sheet (CSS).

Prefer capitals? Extensible markup language (XML). Next: XML. Full: extensible markup language (XML).

1.8 Conditionals

You can test if a term has been defined using `\iftermexists`. For example: seal exists. Another example: jabberwocky doesn't exist.

You can test if a term has been used via `ifentryused`. For example: seal has been used Another example: pglist hasn't been used.

1.9 Symbols

Terms may have an associated symbol. The symbol can be accessed using `\glssym` or if you don't want to add information to the location list you can use `\glsdispentry`. Here's the symbol associated with the cardinality entry: $|\mathcal{S}|$.

A set (denoted $\mathcal{S}$) is a collection of objects. The universal set is the set of everything. The empty set contains no elements. The cardinality of a set (denoted $|\mathcal{S}|$) is the number of elements in the set.

1.10 Location Ranges

A range is formed if a location sequence contains more than 2 locations. Here's seal again.

[1] extensible markup language.
[2] Cascading style sheet.

Glossaries

Glossary

Bravo 1) cry of approval (pl. bravos). 1 2) hired ruffian or killer (pl. bravoes). 1

Glossary 1) collection of glosses. 1 2) list of technical words. iii, 1, 2

Index an alphabetical list of names or subjects with references to their location in the document (pl. indices or indexes). iii, 2

Sea lion large seal. 2

Seal sea mammal with flippers that eats fish. 2, 3

List of Acronyms

CSS Cascading Style Sheet. 3

HTML Hyper-text Markup Language. 2, 3

XML Extensible Markup Language. 3

Notation

Set a collection of distinct objects. ($\mathcal{S}$) 3

Cardinality the number of elements in the set $\mathcal{S}$. ($|\mathcal{S}|$) 3

Index

Locations in bold indicate primary reference. Locations in italic indicate definitions in the glossaries.