Creating alphabetized *see also* cross-references using Index-Manager

Donald Howes

Because Index-Manager produces XE fields for the creation of an index in a Microsoft Word document, its default cross-reference field creation has the same problems of placement and alphabetization as fields created by the Word indexing module. Previous solutions to this problem in a Word index require a search-and-replace routine of relevant XE fields after the indexing module creates them in the document. However, it is possible to correct these problems in Index-Manager, and this article explains the process. It involves the creation of a specialized subheading for the generation of *see also* cross-references. Examples of the subheading are shown and its syntax is explained.

Introduction

As Index-Manager creates XE fields in the source manuscript, it has the same deficiencies as the Microsoft Word indexing module when it comes to the creation of *see also* cross-references (xrefs).[1] First, one has no control over the placement of xrefs. The Word indexing module places a *see also* xref as the first subheading following the main heading. Second, if there are multiple *see also* xrefs, the xrefs are merged but not alphabetized.

See also xrefs and Word

In a series of articles in *The Indexer*, Walter Greulich has exhaustively explored the use of the Word indexing module in the creation of indexes (Greulich, 2020a; 2020b;

Donald Howes has an academic background in archaeology, specializing in Maya archaeology and the archaeology of the American Southwest. He worked as a software developer for companies large and small and has owned and operated an online antiquarian bookstore. He currently works as a freelance indexer, specializing in back-of-book and embedded indexes for scholarly and trade publications. He is a member of the Indexing Society of Canada/Société canadienne d'indexation (ISC/SCI), and of the American Society for Indexing (ASI).
Email: dwhowes@shaw.ca

 https://doi.org/10.3828/index.2024.28

2020c; 2021a; 2021b; 2021c). In Greulich (2020c: 382–5) he shows how to create a *see also* xref as the final subheading. As he explains,

> The following case is given:
>
> InDesign 315–30, siehe ouch FrameMaker
> Gesamtindexerzeugung 324–30
> Indexmarken 317–23
> Platzierung von Dokumenten 316
> Zwischenuberschriften 327
>
> This is the usual syntax for cross-references. To achieve the following arrangement:
>
> InDesign 315–30
> Gesamtindexerzeugung 324–30
> Indexmarken 317–23
> Platzierung von Dokumenten 316
> Zwischenuberschriften 327
> siehe ouch FrameMaker
>
> the XE field has the syntax
>
> { XE ,,InDesign:siehe ouch FrameMaker;zzz" \t ,," }
>
> Greulich (2020c: 382–3)

However, in order to achieve this, as a final step *before* generating the index, a search-and-replace routine must be done in the Word file to change the syntax of the XE field.

It is possible to get the same result in Index-Manager by creating a correctly formatted subheading entry.

See also xrefs in Index-Manager

Normally, *see* and *see also* xrefs would be entered on the Cross-references tab (Figure 1).[2] In this instance, however, because *see also* xrefs are created as subheadings in the index window, only *see* xrefs should be created on this tab.

See also *xrefs as subheadings*

Index-Manager takes the contents of the *Entry* and *Subentry* fields and translates them into an XE field in the Word document. The goal is to create a text string in the

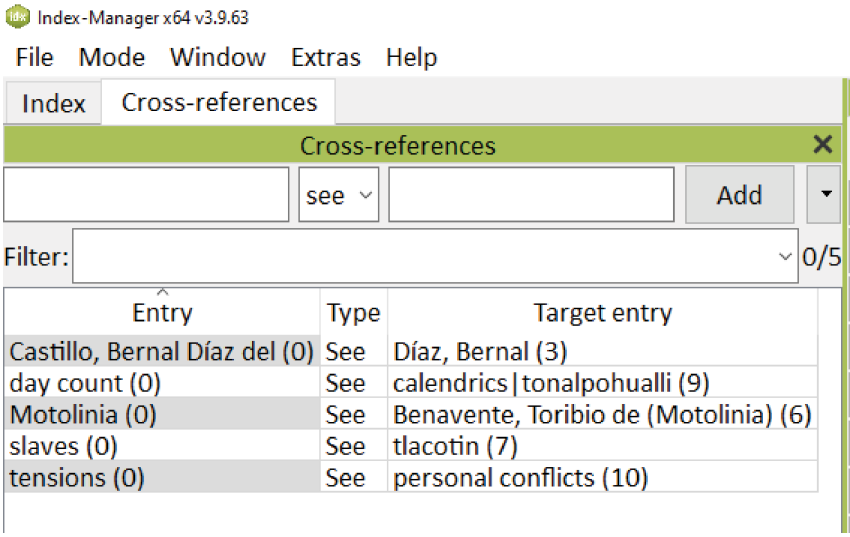

Figure 1. Index-Manager Cross-references tab

Subentry field that produces an output XE entry that matches the format of the XE field created by Greulich.

Figure 2 shows an example of the creation of a simple, one-element *see also* xref. The subheading entry has the syntax

```
<i>See also</i> land tenure;zzz\" \t
```

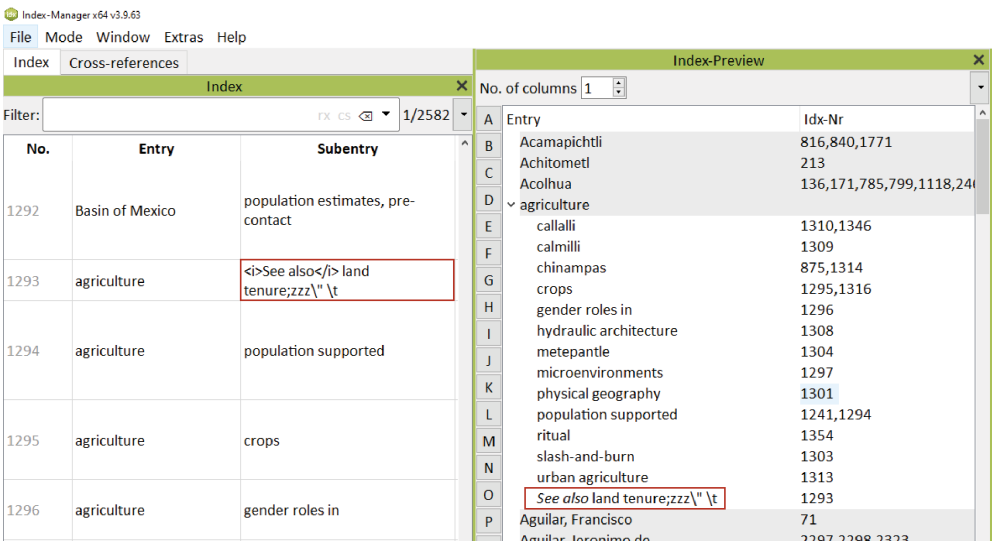

Figure 2. *See also* xref as a subheading in Index-Manager

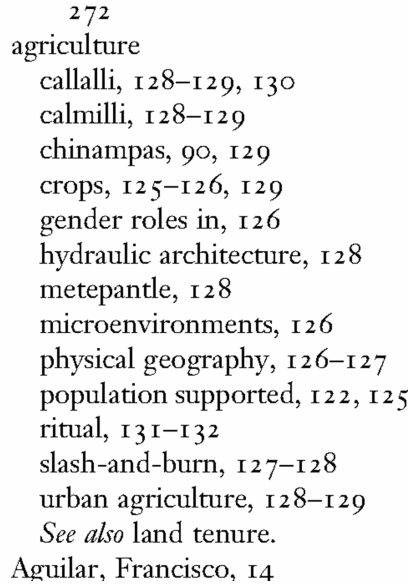

Figure 3. Formatted output with *see also* xref as the final subheading in the printed index (Kellogg, 2024: 352)

Figure 3 shows the result as formatted for print. An example of a multi-element *see also* xref subheading is

```
<i>See also</i> Aztecs\; Maya\; Tolteca;zzz\" \t
```

This gives the formatted output

See also Aztecs; Maya; Tolteca

Additional elements are added to the xref by editing the *Subentry* field for the xref and adding the new information to the text string.

Structure of a see also *xref subheading*

The breakdown of the xref syntax is as follows.

- `<i>See also</i>`: sets the text 'See also' in italic. For additional control over text formatting, change this to `<i>See</i><i> also</i>` so that each word can be formatted independently. Capitalize as required by press guidelines.
- `Aztecs\; Maya\; Tolteca`: when there are multiple entries in the xref, sort them manually as they are entered, then escape each semi-colon (;) within the text string with a backslash (\). A semi-colon (;) is a special character in the Word indexing module and is used to indicate a sort string for forced sorting. Change the element separator character to match press guidelines.

- `;zzz\" \t`: this control sequence does two things. First, the text string following the semi-colon (`;zzz`) forces the xref to sort as the final subheading, and second, the `\t` flag suppresses the printing of a page number for the subheading. The sequence should be entered exactly as shown (including the space before the `\t` flag). However, replace `zzz` with `aaa` to make the xref the first subheading.

This method can be used to create *see also* xrefs, formatted to match press requirements.[3]

Acknowledgements

I would like to thank Alison Craig, who edited early drafts of this article. Her attention to detail was invaluable. I would also like to thank Katharina Munk (Klarso GmbH), who directed me to the solution proposed by Walter Greulich and suggested some ideas as to how this could be accomplished in Index-Manager. Any remaining errors are, of course, my own.

Notes

1 Microsoft and Microsoft Word are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Index-Manager is a trademark of Klarso® GmbH.

2 All examples come from the index I created for *A concise history of the Aztecs* by Susan Kellogg (Kellogg, 2024).

3 The Word indexing module has a hidden and undocumented maximum length for an index entry (length of main heading and all sub-headings combined.) If this maximum length is exceeded, the module fails silently and, while the xref is added as a subheading, it is not placed correctly as the final subheading.

References

Greulich, W. (2020a) 'Embedded indexing with Word: new light on an old topic. Part 1: how to monitor creation of an index', *The Indexer* **38**(2), 207–18. Available at https://doi.org/10.3828/indexer.2020.18.

Greulich, W. (2020b) 'Embedded indexing with Word. Part 2 – editing entries and making work easier by using macros', *The Indexer* **38**(3), 291–306. Available at https://doi.org/10.3828/indexer.2020.27.

Greulich, W. (2020c) 'Embedded indexing with Word. Part 3 – shifting method and field codes for cross-references and page ranges', *The Indexer* **38**(4), 381–98. Available at https://doi.org/10.3828/indexer.2020.36.

Greulich, W. (2021a) 'Embedded indexing with Word. Part 4 – page reference annotations and functional ebook indexes', *The Indexer* **39**(1), 85–100. Available at https://doi.org/10.3828/indexer.2021.8.

Greulich, W. (2021b) 'Embedded indexing with Word. Part 5 – locators other than page numbers', *The Indexer* **39**(2), 183–98. Available at https://doi.org/10.3828/indexer.2021.18.

Greulich, W. (2021c) 'Embedded indexing with Word. Part 6 – sorting and export of entries', *The Indexer* **39**(3), 263–82. Available at https://doi.org/10.3828/indexer.2021.25.

Kellogg, S. (2024) *A concise history of the Aztecs*. Cambridge: Cambridge University Press.