Scribus Index Feature
Phase 2 Specification

John R. Culleton, Jr.

Wexford Press

Eldersburg

Contents

Summary

1.1 Purpose

This document privides a specification of Phase 2 of the Scribus Indexing Project. Phase I utilized the program makeindex and a tcl/tk program tyro.tcl as a gui front end. Phase 2 will eliminate the tyro.tcl program and use Scribus as the gui front end instead. The program makeindex will still do the heavy lifting.

1.2 Components

The Phase 2 approach requires that 5 steps be programmed:

A. Create or open foo.idx file.

B. Accept index item from author and reformat.

C. Add index item to foo.idx

D. Run makeindex program using foo.idx and creating foo.ind

E. import foo.ind at end of scribus document.

Step A. occurs when scribus is called or renamed.

Steps B. and C. occur when the author chooses via an icon.

Step D. can occur after each occurrence of Step C. Alternatively it can be deferred until just before Scribus processes Step E.

Step E can be a separate user action with a separate icon. Or it can be triggered after each occurence of step C.

The Steps in Detail

2.1 Step A: Create or open foo.idx file.

This should occur when Scribus is started. A plain text file is created with the same name as the sla file but with the suffix idx. If the file already exists then no action is taken.

If the name of the file is changed via a **Save as** action e.g., file **foo.sla** is saved as file **bar.sla**, then the .idx file is renamed from **foo.idx** to **bar.idx** and if file **foo.ind** exists then it is renamed as **bar.ind**.

2.2 Step B: Accept index item and reformat, simple version.

This is the step that requires the most programming. Initially it can be programmed as a pop up window with a single text entry item and a Submit button. When used on e.g.,page 32 and the submit button is pressed then the text item e.g., **dog** is reformatted as **\indexentry{dog}{32}**.

If the author wants a subentry then the entry in the text box would be e.g., **dog!poodle**. Then the index tag would be **\indexentry{dog!poodle}{32}**

2.3 Step C: Add index item to foo.idx

First a test is made to see if file **foo.idx** exists. If it does not then it is created. The file is opened and the index tag is appended to the file. The sequence of entries in the file does not matter (see the complex version of Step B for an exception). An index entry made on page 100 can be followed by an entry made on page 1. An entry for zebra can precede an entry for aardvark.

2.4 Step D: Run makeindex program–create foo.id.

This step calls a script which in turn contains the commnd: `makeindex -s alpha.ist -l foo.idx`

The `-s alpha.ist` parameter calls a plain text executable file containing the specific formatting values needed for Scribus (different from the TeX default). The `-l` parameter changes the sorting of the file from word ordering to letter ordering. Different publishers may require one or the other. If word ordering is needed then the `-l` parameter should be omitted in the script file.

The description of the difference between word and letter ordering is a bit technical. Here is a quote from the makeindex man page:

```
 By default, makeindex assumes word ordering;
if the  -l  option  is  in effect,  letter  ordering  is used.
In word ordering, a blank precedes any letter in the alphabet,
whereas in letter  ordering, it  does  not count at all.
This is illustrated by the following example:

              word order                        letter order
              sea lion                          seal
              seal                              sea lion

Numbers are always sorted in numeric order.  For instance,

             9 (nine),  123 10 (ten), see Derek, Bo

Letters are first sorted without regard to case; when
words are identi- cal, the uppercase version precedes its
lowercase counterpart.

 A special symbol is defined here to be any character not
appearing  in the  union  of  digits and the English alphabetic
characters.  Patterns starting with special symbols precede
numbers, which  precede  patterns starting  with  letters.   As
a special case, a string starting with a digit but mixed with
non-digits is considered to be a pattern  starting with a special
character.
```

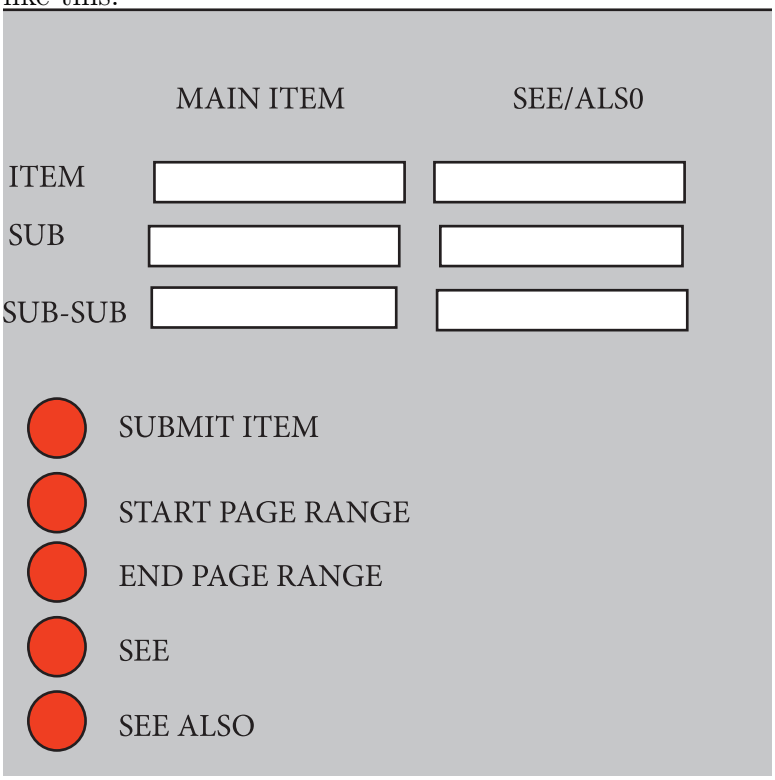
2.5 Step E: importing the index.

The output of Step D is a text file in single column format. It needs to be imported and put in two-column format at the end of the document.

Step B: complex version

3.1 Pop-up window

For the more complex version of Step B a bigger pop-up window is needed, something like this:



The five circles at the bottom are radio buttons.They generate:

- A normal index item–
- The start of a renge of page entries for an index item–
- The close of a range of page entries for an index item–
- A *see* item–
- A *See also* item–

depending on which is pushed. For regular index items the first column marked MAIN ITEM is only used and for *See* or *See also* both columns must be used.

3.2 Example files.

Thus far we have discussed two very simple formats for index items in the **foo.idx** file. For the complex version of Step B we need to discuss several other formats. Here are some examples:

```
(1)
\indexentry{dog!poodle!toy}{33} (2)
\indexentry{dog!poodle!alpha@toy}{33} (3)
\indexentry{dog!poodle!beta@miniature{34} (4)
\indexentry{dog!poodle!charlie@standard{35}
```

Example (1) above shows an index item with a sub-sub-entry. while seldom used we should provide for it.

Examples (2), (3) and (4) show how the sorting of (in this case) sub-subitems can be altered in the final index, to fit a logical sequence instead of an alphabetic sequence. Alpha, Beta and Charlie are sort keys. They could have alse been 1,2,3. The author enters the sort key in the same window as the item in the form of a prefix to the item separated by a "@" character as above e.g., `alpha@toy` etc.

3.3 Additional formats

Here are some more formats:

```
(5)
\indexentry{future|(}{3} (6) \indexentry{dichotomies}{8} (7)
\indexentry{future|)}{11} (8)
\indexentry{analysis!archetypical|indexsee{archetypical
criticism}}{99999} (9) \indexentry{archetypical criticism}{14}
(10)\indexentry{archetypical criticism!elements
of|indexseealso{dichotomies}}{99999}
```

Examples (6) and (10) are just ordinary entries.

Examples (5) and (7) respectively open and close a range of pages.

Example (8) is a **See** entry.

Example (10) is a **See also** entry.

These last two entries use a dummy page number of 99999.

These entries resulted in a foo.ind file like this:

```
--A--


analysis archetypical, \indexsee{archetypical criticism}{99999}
archetypical criticism, 14 elements of,
\indexseealso{dichotomies}{99999}

   --D--


dichotomies, 8

   --F--


future, 3--11
```

The output obviously needs to be massaged further to substitute *See* for `\indexsee` and *See also* for `\indexseealso` And the dummy page numbers 99999 and all braces need to be removed. But this must occur after Step D.

This ends the definition of Scribus Index Specification: Phase 2 as envisioned by the author. Comments and corrections can be posted on the scribus mailing list.