

The `ArcIdx` package*

Alan Robert Clark
a.clark@ee.wits.ac.za[†]

2002/12/09

Abstract

The `ArcIdx` package was originally conceived by André Fourie (a.fourie@ee.wits.ac.za), but tweaked and documented by me (a.clark@ee.wits.ac.za), with changes to the logic of the commandset, where this was necessary, and changes to the terminology (eg main versus primary :-)

Effectively the `ArcIdx` package attempts to smooth the production of a comprehensive index, which with the ordinary `\index` commands is a tedious process, needing particular care with % after lines so that extra space does not occur in the text etc.

In addition, one often finds that a “double entry” is necessary: Dipole antenna and Antenna, dipole. Note initial case changes. These macros facilitate this kind of thing, with a zillion different cases.

Thus there are a few basic macros, and then several variations on the theme.

1 Introduction

The construction of an index is an artform. It really requires getting inside someone else’s head in order to answer the question: “When looking for topic *A*, what *keywords* would be searched for?” I have met excellent books in which I can find nothing, since the index constructor and I seem to be wired differently, and I have met mediocre books with an excellent, and extensive index. I strongly prefer the latter!

In $\text{\LaTeX}_{2\epsilon}$, the `makeindex` programme is an add-on extra which processes the outputs of `\index{}` commands embedded in the text. There are a few difficulties:

- The `\index` command must be placed very close to the actual occurrence of the instance, otherwise a page break could easily force a wrong page number in the index.
- As a direct result, paragraphs and even lines are broken to provide the `\index` command, leading to messy input files. In order to neaten the process, it is often useful to have these “stick out” by blank lines, but these obviously have to be commented, leading to a less than satisfactory result.
- It is tedious to re-type, or cut-and-paste the phrase if it is meant to appear in the index as it appears in the text.
- Often a “double entry” is required: as in Bandwidth, helix and Helix, bandwidth (Note case changes). This type of entry might not appear verbatim in the text.

There are a number of philosophical points regarding an index. It is preferable to do the index *AFTER* constructing the text, as then the “full picture” is available. In the Stone Age, this was true, and it is certainly the proper way to approach the artform. The text, especially in an e-book, is always evolving, and it is valid to attempt to populate the index “on-the-fly” as this package enables. However, it is to be remembered that a mixture of the

*This file has version number v1.2, last revised 2002/12/09.

[†]adapted from the original style file by André Fourie, a.fourie@ee.wits.ac.za

two approaches is the correct one. One has to have OverSight at some point in order to get the Bigger Picture, whilst still having the lower level stuff automated.

2 Summary of user commands

Because the range of possible options is large, a large number of commands are available. The simplistic commands are a small subset of these, and already make the production of an index simpler and more efficient, but inevitably, the need to do “different” things requires a bit more complexity.

A consistent command-naming paradigm has thus been adopted to be able to wade through the zillions of options.

3 The macros

In this section the actual macros are documented at a source code level.

Internal Internal commands not meant to be called by the user. Basically, they provide the ability to upper and lower case the initial letter to allow Main, sub and Sub, main type entries. Can override the `\formatrest` command to enforce lowercase, for example, but that plays hell with VSWR :-)

```

1 \newcommand*{\formatfirst}[1]{\MakeUppercase{#1}}
2 \newcommand*{\formatrest}[1]{#1}
3 \makeatletter
4 \newcommand*{\startcap}[1]{%
5   \expandafter\formatfirst\expandafter{\@car #1\@empty\@nil}%
6   \expandafter\formatrest\expandafter{\@cdr #1\@empty\@nil}%
7 }
8 \makeatother
9 \newcommand{\bb}[1]{\textbf{#1}}
10 \newcommand{\ii}[1]{\textit{#1}}
```

`\mid{x,s,e}[q]` The (main) primary index commands below allow you to just add a command in front of a regular word in your document and enclose that word in `{}`. The word will then still appear in the main document and will be indexed as the Main single (in text) index item (for a most important reference to the keyword). Use the command with the `s` (start) and `e` (suffices) to get range of pages. All three commands can be postfixed with a `q` (for quiet) to suppress the word in the document.

Thus used in text as `\midx{indexWord}`, which produces a main capitalised index entry, and puts the `indexWord` in the document; `\mids{indexWord}` puts the `indexWord` in the document and records the start of a range, whereas `\mide{indexWord}` records the end of the important range in the index, and does *NOT* produce the `indexWord` in the document, obviously :-)

The quiet forms of the command with an appended `q` do not put the `indexWord` into the document. `\mideq` is redundant, but provided for completeness.

```

11 \newcommand{\midx}[1]{#1\index{#1\startcap{#1}|bb}}
12 \newcommand{\mids}[1]{#1\index{#1\startcap{#1}|(bb)}}
13 \newcommand{\mide}[1]{\index{#1\startcap{#1}|)}}
14 \newcommand{\midxq}[1]{\index{#1\startcap{#1}|bb}}
15 \newcommand{\midsq}[1]{\index{#1\startcap{#1}|(bb)}}
16 \newcommand{\mideq}[1]{\index{#1\startcap{#1}|)}}}
```

`\id\{x,s,e}\{q}` Produces the *ordinary* primary index entry, with all the variations of the “main” indexing commands. Since defining a range for a keyword seems to imply that it is important, the range commands simply invoke the (main) range commands, ie they are provided for completeness.

```
17 \newcommand{\idx}[1]{#1\index{#1\startcap{#1}}}
18 \newcommand{\ids}[1]{#1\index{#1\startcap{#1}|(bb)}}
19 \newcommand{\ide}[1]{\index{#1\startcap{#1}|)}}
20 \newcommand{\idxq}[1]{\index{#1\startcap{#1}}}
21 \newcommand{\idsq}[1]{\index{#1\startcap{#1}|(bb)}}
22 \newcommand{\ideq}[1]{\index{\startcap{#1}|}}}
```

`\sid\{x,s,e}\{q}` Produces the *ordinary* sub-item index (of in-text word with two arguments). First argument will be reproduced in text where it occurred. Attempt to resurrect the optional argument....

```
23 \newcommand{\sidx}[2]{#2\index{#2\startcap{#2}!#1\MakeLowercase{#1}}%
24 \index{#1\startcap{#1}!#2\MakeLowercase{#2}}}
25 \newcommand{\sids}[2]{#2\index{#2\startcap{#2}!#1\MakeLowercase{#1}|(bb)}%
26 \index{#1\startcap{#1}!#2\MakeLowercase{#2}|(bb)}}
27 \newcommand{\side}[2]{\index{#2\startcap{#2}!#1\MakeLowercase{#1}|)}}%
28 \index{#1\startcap{#1}!#2\MakeLowercase{#2}|)}}
29 \newcommand{\sidxq}[2]{\index{#2\startcap{#2}!#1\MakeLowercase{#1}}%
30 \index{#1\startcap{#1}!#2\MakeLowercase{#2}}}
31 \newcommand{\sidsq}[2]{\index{#2\startcap{#2}!#1\MakeLowercase{#1}|(bb)}%
32 \index{#1\startcap{#1}!#2\MakeLowercase{#2}|(bb)}}
33 \newcommand{\sideq}[2]{\index{#2\startcap{#2}!#1\MakeLowercase{#1}|)}}%
34 \index{#1\startcap{#1}!#2\MakeLowercase{#2}|)}}}
```

`\Sid\{x,s,e}\{q}` The Capitalised form of the sub-indexing code leaves the second term capitalized (eg for VSWR)

```
35 \newcommand{\Sidx}[2]{#2\index{#2\startcap{#2}!{#1}}%
36 \index{#1\startcap{#1}!{#2}}}
37 \newcommand{\Sids}[2]{#2\index{#2\startcap{#2}!{#1}|(bb)}%
38 \index{#1\startcap{#1}!{#2}|(bb)}}
39 \newcommand{\Side}[2]{#2\index{#2\startcap{#2}!{#1}|)}}%
40 \index{#1\startcap{#1}!{#2}|)}}
41 \newcommand{\Sidxq}[2]{\index{#2\startcap{#2}!{#1}}%
42 \index{#1\startcap{#1}!{#2}}}
43 \newcommand{\Sidsq}[2]{\index{#2\startcap{#2}!{#1}|(bb)}%
44 \index{#1\startcap{#1}!{#2}|(bb)}}
45 \newcommand{\Sideq}[2]{\index{#2\startcap{#2}!{#1}|)}}%
46 \index{#1\startcap{#1}!{#2}|)}}}
```

`\eidx\{x}\{q}` For equations, a special form of sub-index is provided, which only takes One argument, the primary term being “Equations”. Only quiet forms exist, since you would not ordinarily want the phrase in the text. Entries in the index are then Equations!link, and Link!equation.

There is also a double form, which takes two arguments. Note that produces 3 index entries!

```
47 \newcommand{\eidxq}[1]{%
48 \index{Equations!#1\MakeLowercase{#1}}%
49 \index{#1\startcap{#1}!equation}%
50 }
51 \newcommand{\eidxxq}[2]{%
52 \index{Equations!#1\MakeLowercase{#1}!#2\MakeLowercase{#2}}%
53 \index{#1\startcap{#1}!#2\MakeLowercase{#2}}%
54 \index{#2\startcap{#2}!#1\MakeLowercase{#1}}%
55 }
```

`\Eidx\{x}\{q}` Use Capital Eidx if you want the terms to remain capitalized

```

56 \newcommand{\Eidxq}[1]{%
57   \index{Equations!#1}%
58   \index{#1@startcap{#1}!equation}%
59 }
60 \newcommand{\Eidxq}[2]{%
61   \index{Equations!#1!#2}%
62   \index{#1@startcap{#1}!#2}%
63   \index{#2@startcap{#2}!#1}%
64 }

```

`\fidx[x]q` For figures, a special form of sub-index is provided, similar to that provided for equations, which only takes One argument, the primary term being “Figures”. Only quiet forms exist, since you would not ordinarily want the phrase in the text. Entries in the index are then Figures!link, and Link!figures.

There is also a double form, which takes two arguments. Note that produces 3 index entries!

Finally, with André going completely mad, a triple argument form is provided, with 7 index entries! Use at your peril!

```

65 \newcommand{\fidx}[1]{%
66   #1\index{Figures!\#1@MakeLowercase{#1}|ii}%
67   \index{#1@startcap{#1}|ii}%
68 }
69 \newcommand{\fidxx}[2]{%
70   #2\index{Figures!#1@MakeLowercase{#1}!#2@MakeLowercase{#2}|ii}%
71   \index{#1@startcap{#1}!#2@MakeLowercase{#2}|ii}%
72   \index{#2@startcap{#2}!#1@MakeLowercase{#1}|ii}%
73 }
74 \newcommand{\fidxq}[1]{%
75   \index{Figures!\#1@MakeLowercase{#1}|ii}%
76   \index{#1@startcap{#1}|ii}%
77 }
78 \newcommand{\fidxxq}[2]{%
79   \index{Figures!#1@MakeLowercase{#1}!#2@MakeLowercase{#2}|ii}%
80   \index{#1@startcap{#1}!#2@MakeLowercase{#2}|ii}%
81   \index{#2@startcap{#2}!#1@MakeLowercase{#1}|ii}%
82 }
83 \newcommand{\fidxxxq}[3]{%
84   \index{Figures!#1@MakeLowercase{#1}!#2@MakeLowercase{#2}|ii}%
85   \index{#1@startcap{#1}!#2@MakeLowercase{#2}|ii}%
86   \index{#2@startcap{#2}!#1@MakeLowercase{#1}|ii}%
87   \index{#1@startcap{#1}!#3@MakeLowercase{#3}|ii}%
88   \index{#3@startcap{#3}!#1@MakeLowercase{#1}|ii}%
89   \index{#2@startcap{#2}!#3@MakeLowercase{#3}|ii}%
90   \index{#3@startcap{#3}!#2@MakeLowercase{#2}|ii}%
91 }

```

`\Fidx[x]q` Use Capital Fidx if you want argument to remain capitalized.

```

92 \newcommand{\Fidx}[1]{%
93   #1\index{Figures!#1|ii}%
94   \index{#1@startcap{#1}|ii}%
95 }
96 \newcommand{\Fidx}[2]{%
97   #2\index{Figures!{#1}!#2|ii}%
98   \index{#1@startcap{#1}!#2|ii}%
99   \index{#2@startcap{#2}!#1|ii}%
100 }
101 \newcommand{\Fidxq}[1]{%
102   \index{Figures!#1|ii}%
103   \index{#1@startcap{#1}|ii}%
104 }

```

```

105 \newcommand{\Fidxxq}[2]{%
106   \index{Figures!#1!#2|ii}%
107   \index{#1@\startcap{#1}!#2|ii}%
108   \index{#2@\startcap{#2}!#1|ii}%
109 }
110 \newcommand{\Fidxxxq}[3]{%
111   \index{Figures!#1!#2|ii}%
112   \index{#1@\startcap{#1}!#2|ii}%
113   \index{#2@\startcap{#2}!#1|ii}%
114   \index{#1@\startcap{#1}!#3|ii}%
115   \index{#3@\startcap{#3}!#1|ii}%
116   \index{#2@\startcap{#2}!#3|ii}%
117   \index{#3@\startcap{#3}!#2|ii}%
118 }

```