

Program and package xindex

—

1.07 (August 13, 2026)

Herbert Voß

hvoss@tug.org

Contents

1. Introduction	2
1.1. Syntax	2
1.2. How it works	3
1.3. The .idx file	3
2. Language	5
3. Default sorting by the UCA (Unicode Collection Algorithm)	7
3.1. Examples	8
3.1.1. French language	8
3.1.2. German language	9
3.1.3. German language (DIN2)	9
3.1.4. Japanese language	10
3.2. Case sensitive index entries	10
3.3. Ignore space for sorting	11
4. Pagenumbers	12
4.1. Compressing pagenumber series	12
4.2. Modify Pagenumber	12
4.3. Suppress Pagenumber	13
5. The config file	13
6. hyperref	16
7. Including commands into the .idx file	17
8. Headings	18
9. Automatic index creation	18
10. Labels	19
11. Demerits	20
A. Examples	20



Thanks to:

Mark Baudoin; Denis Bitouzé; Andreas Deininger; Brian Dunn; Daniel Flipo; Yukitoshi Fujimura; Michal Hoftich; Dohyun Kim; Mykel Kochenderfer; Alexey Kuznetsov; Frank Mittelbach; Peter Müller; Rolf Niepraschk; Heiko Oberdiek; Matteo Paolini; Stefan Pinnow; Florent Rougon; Gonzalo Pesquero Serrano; Martin Sievers; Simon Spiegel; Jürgen Spitzmüller;

1. Introduction

The Lua program `xindex` is a unicode aware program for creating an index (`.ind`) file from an `.idx` source file. It is completely compatible to the current `makeindex` program, but can handle UTF-8, 16, 32, and 64. The $\LaTeX$ package `xindex` is a package which provides a $\LaTeX$ command which writes additional text into the index file. This text (comments and/or macros) will be accepted by the program `xindex`.

The general structure of a data element in the Lua table is:

```
data = { Entry = <text>, -- like the input line without command \indexentry
  pages = {
    { number = <roman/arabic number or text>,
      special = <macro> }, -- the part after | in the input
    [...]
    { number = <roman/arabic number or text>,
      special = <macro> }
  },
  sortChar = <unicode codepoint>, -- of the first character of Entry
  Macro = <TeX macro> -- only useful with LaTeX package xindex
}
```

After reading the input file the Lua table `pages` has only one element for the number and the so-called special command. When the pages are compressed the table will collect all pages which refer to the same entry name.

1.1. Syntax

The syntax is `xindex [...] <file(s)>` where `[...]` are optional arguments, either in short or long form which, of course, can be mixed:

```
xindex
  [-q,--quiet ]
  [-h,--help ]
  [-v ] verbose
  [-V,--version ]
  [-a,--no_casesensitive ] default is false
  [-b,--no_labels ] default is false
  [-c,--config ] default is cfg
  [-e,--escapechar ] default is "
  [-f,--fix_hyperref ] default is false
  [-g,--no_pagenumber] default is false
  [-i,--ignoreSpace ] default is false
  [-k,--checklang ] default ist false
```

```

[-l,--language ]           default is en
[-n,--noheadings ]         default is false
[-o,--output ]             default is <input>.ind
[-p,--prefix]              default L
[-r,--restricted]          default is false
[-s,--use_stdin ]          default is false
[-r,--restricted]          default is false
<files...> (default stdin) file(s)[.idx]  one or more files

```

For example:



```

xindex -q -l fr -b myDoc
xindex -l de -c DIN2 demo1 demo2 demo3
xindex -c norsk -o index.ind demo1 demo2.bdx demo3.adx

```

1. -q: quiet; -l fr: french language setting; -b: no labels; myDoc: input data myDoc or myDoc.idx
output data will be myDoc.ind and logfile myDoc.ilg
2. -l de: German language setting; -c DIN2: config file xindex-DIN2.lua; demo1 demo2 demo3: input data files
with or without extension .idx
output data will be demo1.ind and logfile xindex.ilg
3. -c norsk: config file xindex-norsk.lua; -o index.ind: output file; demo1 demo2.bdx demo3.adx: input data files
with or without extension .idx
output data will be xindex.ind and logfile xindex.ilg

It is also possible to use standard input for the index data, which needs the -s parameter:



```
cat myDoc.idx | xindex -q -l fr -b -s xindex -l de -c DIN2 < myDoc.idx
```



The language has to be chosen as an international abbreviation in lower- or uppercase letters, see https://en.wikipedia.org/wiki/ISO_3166-2

1.2. How it works

xindex creates by default an output file <input>.ind which can be read by the $\LaTeX$ document with the default command `\printindex`. One can use another output filename, which makes only sense if one doesn't use the `\printindex` command for typesetting the index. The default sorting is given by the configuration file, which defines replacements for accented characters, like `ö→o`.

1.3. The .idx file

There are three (four) characters which must be escaped if used in the command `\index`: `!`, `@`, or `|` and the current escape character itself. These characters have a special meaning for the index.



The default escape character is the double quote `"`. The braces `{` and `}` cannot be used as argument for the command `\index`. Use `\braceLeft` and `\braceRight` instead (defined in the package xindex).

Escaping characters

Exclamation mark !
 Vertical bar |
 Doublequote "
 Double doublequote ""
 At character @
 Left parenthesis {
 Right parenthesis }
 index -l fr <file.idx>

Index**Symboles**

!, 1
 ", 1
 """, 1
 {, 1
 }, 1
 @, 1
 |, 1

Nombres

123, 1

A

At (@), 1

E

Etage, 1
 Étagé, 1
 Exclamation (!), 1

F

<file.idx>, 1

V

Vertical bar (|), 1

It is by design that the braces { and } cannot be used as index entry. The *package* xindex defines the two commands \braceLeft and \braceRight which can be used instead (see examples above and below).

For the German language the double quote is an active character and it makes life easier if one chooses another character. The escape character can be changed easily by the optional argument -e "<char>" or --escapechar "<char>". The following example shows how it works for the escape character »>« (greater). By default the expression »>>« will be a T_EX ligature with the output »>>«.

With the beginning of xindex the escaped chars are converted into the internal strings and later back to the original meaning. The two characters {} cannot be used as \{\} inside the argument of \index. The package xindex defines the two helper macros



```
\providecommand\braceLeft{\{}
```

```
\providecommand\braceRight{\}}
```

The following example shows how to use it:

Escaping characters with >

Exclamation mark !
 Vertical bar|
 Escapechar >
 Double escapechar >>
 At character @
 Group start {
 Group end {
 index with xindex -e ">" -n

Index

{, 1
 }, 1
 <file.idx>, 1
 >, 1
 », 1
 123, 1
 At (@), 1
 exclaim (!), 1
 Vertical bar (|), 1

2. Language

The language is only important for the first two headers in the output of the index data. They are by default *Symbols* followed by *Numbers*. In a new version of xindex it will be customizable. The predefined language is »en« and currently the following languages which its alias are defined:

```
<id> = {<symbols>, <numbers>, <alias language name>, ...}
indexheader = {
  cs = {"Symboly", "Čísla", "czech"},
  da = {"Symboler", "Tal", "danish"},
  de = {"Symbole", "Zahlen", "austrian", "german", "germanb", "ngerman", "naustrian"},
  en = {"Symbols", "Numbers", "english", "USenglish", "american", "UKenglish", "british", "canadian", "australian", "newzealand"},
  es = {"Símbolos", "Números", "spanish"},
  fr = {"Symboles", "Nombres", "french", "français", "canadien", "acadian"},
  it = {"Simboli", "Numeri", "italian"},
  jp = {"シンボル", "番号", "japanese"},
  nl = {"Symbolen", "Nummers", "dutch"},
  no = {"Symboler", "Tall", "norsk", "nynorsk"},
  ru = {"Символы", "Числа", "russian"},
}
```

The following example was run with `xindex -l it <file>.idx`:

1 Escaping simboli con >

- punto esclamativo !
- linea verticale |
- escapechar >
- doppio escapechar >>
- At simboli @

Initio xindex con `xindex -l it -e ">"`

Indice analitico

Simboli

>, 1

», 1

Numeri

123, 1

A

At (@), 1

E

exclaim (!), 1

V

Vertical bar (|), 1

The following example was run with `xindex -k <file>.idx`. In this case xindex tries to detect the language from the aux file(s). This is only possible if package babel or polyglossia are used.

1 Escaping characters with ?

- Exclamation mark !
- Vertical bar|
- Escapechar ?
- Double escapechar ??
- At character @

Run xindex with `xindex -k -e "?"`

Index

Symbolen

?, 1

??, 1

@, 1

Nummers

123, 1

A

At (@), 1

E

exclaim (!), 1

V

Vertical bar (|), 1

For the Russian language you have to choose the language and the config file. This allows to have different indexes with different language.

войные:	<code>\index{Хвойные}</code>
торрея,	<code>\index{Хвойные!тисовые!торрея (Torreya)}</code>
тис ягодный,	<code>\index{Хвойные!тисовые!тис!ягодный (Táxus baccata)}</code>
ливанский кедр,	<code>\index{Хвойные!сосновые!кедр!ливанский (Cedrus libani)}</code>
ель обыкновенная.	<code>\index{Хвойные!сосновые!ель!обыкновенная (Pícea ábies)}</code>
од колючей ежевикой	<code>\index{Ежевика (Rúbus) }</code>
ил ушастый ёж.	<code>\index{Ёж!ушастый (Hemiechinus auritus)}</code>

Алфавитный указатель

ушастый (Hemiechinus auritus), 1	— — — обыкновенная (Pícea ábies), 1
ика (Rúbus), 1	— — кедр
	— — — ливанский (Cedrus libani), 1
	— тисовые
ные, 1	— — тис
основые	— — — ягодный (Táxus baccata) , 1
ель	— — торрея (Torreya) , 1

3. Default sorting by the UCA (Unicode Collection Algorithm)

This Lua library from Michal Hoftich is part of T_EXLive and is used by default for sorting. The supported languages and variants are listed in the file `lua-uca-languages.lua`:



af, am, ar, as, az, be, bg, bn, bs, bs_cyrl, ca, chr, cs, cy, da, de, de_din2, dsb, dz, ee, el, en, eo, es, et, fa, fi, fil, fo, fr, fr_backward_accents, ga, gl, gu, ha, haw, he, hr, hi, hsb, hu, hy, id, ig, is, it, ja, ka, kk, kl, km, kn, ko, kok, ky, lb, lkt, ln, lo, lt, lv, mk, ml, mn, mr, ms, mt, my, nb, ne, nl, nn, no, om, pa, pl, ps, pt, ro, ru, se, si, sk, sl, smn, sq, sr, sr_latn, sv, sw, ta, te, th, tk, to, tr, ug, uk, ur, uz, vi, vo, wae, wo, yi, yo, zh, zu

The sorting order can be easily modified. Read the documentation of the package `LUA-UCA` on how to do it and what languages are supported so far. Any additional code setting for UCA should be done in the file `xindex-cfg-uca.lua`, which will automatically be read by `xindex`. Language-specific sorting orders can also be defined in a config file.

```
Sorted with -l cs ahoj crha, čaj, chachar,
rak, řeka, srp, šutr, hudba, linux, zebra,
žába, 7 dubů
```

Index	L
	linux, 1
A	R
ahoj, 1	rak, 1
C	Ř
crha, 1	řeka, 1
Č	S
čaj, 1	srp, 1
H	Š
hudba, 1	šutr, 1
Ch	Z
chachar, 1	zebra, 1

Modifications can be done in a config file which is then loaded by the option `-c`. For example: the file `xindex-cfg-uca.lua` has modification for french and norwegian. For french the standard sorting rules `fr_backward_accents`, are a bit special and should be the default also for the language `fr`:

```
languages.fr = function(collator_obj)
  -- reverse search for accents in French (recommended):
  collator_obj.accents_backward = true
  local tailoring = function(s) collator_obj:tailor_string(s) end
  tailoring("&æ=ae")
  tailoring("&ø=oe")
  tailoring("&th<ph<<<þ")      -- Canadian, see SGQRI004.pdf
  return collator_obj
end
```

Add any additional modifications to this file or create an own config file and load it with -c.

3.1. Examples

3.1.1. French language

	coté, 1	lésé, 1
e, 1	Coté, 1	
e, 1	COTÉ, 1	M
ien, 1	côté, 1	MÂCON, 1
	Côté, 1	maçon, 1
is, 1	CÔTÉ, 1	P
t, 1	coté, 2 sq.	pêche, 1
TM, 1	cœur, 1 sq.	péché, 1
1	E	PÉCHÉ, 1
ent, 1, 3	Élève, 1	péché, 2
, 1	élevé, 1	pécher, 1
ence, 1	G	pêcher, 1
	gène, 1	R
1	gêne, 1	relève, 1
	géné, 1	relevé, 1
	L	révèle, 1
1	lèse, 1	révélé, 1

3.1.2. German language

The default sorting where Umlauts are identical to the base latter: ö→o and »ß→s«

Index

F	
Fluss	Oberin, 1
- Oder, 1	Obstler, 1
	Ödem, 1
	oder, 1
G	Oder, 1, <i>siehe auch</i>
Göbel, 1	Fluss
Goethe, 1	Öl, 1
Goldmann, 1	ölen, 1
Göthe, 1	Oligarch, 1
Götz, 1	Öresund, 1
	Ostern, 1
O	Österreich, 1
Ober, 1	

3.1.3. German language (DIN2)

The same sorted with the language setting »German DIN variant 2«, which can be set. with --language de_din2 or l de_din2. Umlauts are now converted into »ö→oe« and »ß→ss«:

Index

F	
Fluss	Oberin, 1
- Oder, 1	Obstler, 1
	oder, 1
	Oder, 1, <i>siehe auch</i>
G	Fluss
Göbel, 1	Ödem, 1
Goethe, 1	Öl, 1
Göthe, 1	ölen, 1
Götz, 1	Öresund, 1
Goldmann, 1	Österreich, 1
	Oligarch, 1
O	Ostern, 1
Ober, 1	

3.1.4. Japanese language

The following runs with `xindex -l jp <file>`:

指数

シンボル //, 1	と ドイツ, 1
番号 4711, 1	ふ プリンタ, 1
B bar, 1	わ ワープロ, 1
F foo, 1	印刷, 1
	天王, 1
	広島, 1
こ コンピュータ, 1	日本, 1, 1
	病院, 1
す スイミングプール, 1	車, 1
	車道, 1

In Japanese, sorting by character code (Unicode) does not follow standard dictionary or index conventions. Because Kanji (ideographic characters) often have multiple phonetic readings depending on context, they must be sorted based on their pronunciation (Kana) rather than the character itself.

For example, a term like 車 should be sorted under 'ku' (kuruma), but its Unicode position is completely different. To make `xindex` viable for professional Japanese publications, it is essential to emphasize the use of the `SortEntry@Output` syntax, such as:

指数

い 印刷, 1	に 日本, 1
く 車, 1	ひ 病院, 1
× 車道, 1	広島, 1
て 天王, 1	わ ワープロ, 1

Without these explicit sort keys, the resulting index will not be readable for Japanese users.

3.2. Case sensitive index entries

By default `foo` and `Foo` are two different entries and will be handled differently by `xindex`: `Foo` will be as an own entry *before* `foo`. Let's see a more complex example. In the index the entry `xindex-DIN2.lua` is the first one of the `xindex-???` series because uppercase letters are sorted before lowercase letters.

Index

X

- xindex package, 2, 15
- xindex program, 4, 13f
- xindex-cfg-common.cfg file, 9
- xindex-cfg-common.lua file, 14
- xindex-cfg.lua file, 6, 10
- xindex-DIN2.lua file, 6
- xindex-dtk.lua file, 12
- xindex-HAdW-eKO.lua file, 10
- xindex-newfile.lua file, 6

The same example sorted with the `-a` or `--no_casesensitive` has another output: now `xindex-cfg-common.lua` is the first one of the `xindex-???` series.

Index

X

- xindex package, 2, 15
- xindex program, 4, 13f
- xindex-cfg-common.cfg file, 9
- xindex-cfg-common.lua file, 14
- xindex-cfg.lua file, 6, 10
- xindex-DIN2.lua file, 6
- xindex-dtk.lua file, 12
- xindex-HAdW-eKO.lua file, 10
- xindex-newfile.lua file, 6

3.3. Ignore space for sorting

By default »alpha sort« will be sorted *before* »alphaA«:

Index

A

- alpha sort, 1
- alphaA, 1

This can be changed with the optional argument `-i` or `--ignoreSpace`:

Index

A

- alphaA, 1
- alpha sort, 1

4. Pagenumbers

4.1. Compressing pagenumber series

By default page sequences of an entry are compressed to

8f page 8 and 9

8ff page 8, 9, and 10

8-12 page 8, 9, ..., 12

The so-called folio abbreviation is language dependent and defined in the file `xindex-cfg-common.cfg`:

```
folium = {
  cs = {"f.", "ff."},
  da = {"f", "ff"},
  de = {"f", "ff"},
  en = {"f", "ff"},
  es = {"f", "ff"},
  fr = {"\\", "sq.", "\\", "sq."},
  it = {"f", "ff"},
  jp = {"シンボル", "番号"},
  nl = {"f.", "ff."},
  no = {"\\", "f.", "\\", "ff."},
  ru = {"", ""},      -- doesn't exist
}
```

Index

B

bar, 1 sqq.

baz, 1-4

F

foo, 1 sq., 5

foobar, 1-5

4.2. Modify Pagenumber

Every page can be combined with an additional macro, like `\index{foo|fbox}`, the page number will be set into a framebox. If we have on the same page the two commands:

```
foo\index{foo} and foo\index{foo|bar}
```

then we have two *different* index entries which will not be compressed to one entry. In the following example we have four different entries for *foo* which is the reason that we do not get an output like *foo*, 1--4. Only the first two entries are of the same type, so we get 1f in the output.

Index

F

foo, 1f, 2, 3, 4

4.3. Supress Pagenumber

Instead of printing an index in the default way, one can also print a glossary without the pagenumbers. This is possible with the optional argument `-g` which is equivalent to the long form `--no_pagenumber`. The following example uses an own config file for the definition of the description environment:

```
itemPageDelimiter = ""
compressPages     = true
fCompress         = false
minCompress       = 2
rangeSymbol       = "--"
numericPage       = true
sublabels         = {"", "\\item", "\\item", "\\item"}
pageNoPrefixDel   = ""
idxnewletter      = "\\textbf"
envStart          = "\\begin{description}"
indexOpening      = "\\makeatletter\\z
                    \\def\\subitem{\\@idxitem} \\def\\subsubitem{\\@idxitem}\\z
                    \\def\\subsubsubitem{\\@idxitem}\\z
                    \\makeatother\\z
                    \\itemsep 0.1ex" -- commands after envStart
envStop           = "\\end{description}"
```

ARPES — Angle resolved photo electron spectroscopy.

EXAFS — Extended x-ray absorpion fine structure.

HTSC — High temperature superconductivity.

MOCVD — Metalorganic chemical vapour deposition.

PES — Photo emission spectroscopy.

PLD — Pulsed laser deposition.

SCES — Strongly correlated electron systems.

XAFS — Extended x-ray absorption fine structure.

XANES — X-ray absorption near edge structure.

XAS — X-ray absorption spectroscopy.

ARPES — Angle resolved photo electron spectroscopy.

Bosons

Boson — a particle with an integer spin S .

Meson — a hadronic subatomic particle composed of an equal number of quarks and antiquarks: $S = 0$.

Photon — a quantum of the electromagnetic field: $S = 1$.

Pion — any of π^0 , π^+ , π^- mesons consisting of quark and antiquark: $S = 0$.

EXAFS — Extended x-ray absorpion fine structure.

Fermions

Electron — a subatomic particle with a negative elementary electric charge: $S = 1/2$.

Fermion — a particle with a half-odd-integer spin S .

Neutrino — an elementary electrically neutral particle with very small rest mass: $S = 1/2$.

Neutron — a subatomic electrically neutral particle: $S = 1/2$.

Positron — a particle with a positive elementary electric charge, and the same mass as an electron: $S = 1/2$.

Proton — a subatomic particle with a positive elementary electric charge: $S = 1/2$.

HTSC — High temperature superconductivity.

MOCVD — Metalorganic chemical vapour deposition.

PES — Photo emission spectroscopy.

PLD — Pulsed laser deposition.

5. The config file

The main config file is `xindex-cfg.lua` and used by default. A new config file must have the prefix `xindex-` and the file extension `.lua`, for example: `xindex-HAdW-eK0.lua` which can be used with `--config HAdW-eK0`. The

file must be saved in the documents directory or in one which is known to kpsewhich, for example¹
 $\$TEXMFLOCAL/tex/lualatex/xindex/$ Do not forget to update the filename database.

A new config file must declare at least the variables which are part of the default config file: the translation tables and

```
itemPageDelimiter = ","      -- Hello, 14
compressPages      = true
  -- something like 12--15, instead of 12,13,14,15. the |( ... |) syntax is still valid
fCompress          = true    -- 3f -> page 3, 4 and 3ff -> page 3, 4, 5
minCompress        = 3       -- 14--17
rangeSymbol        = "--"
numericPage        = true    -- for non numerical pagenumbers, like "VI-17"
sublabels          = {"", "-\\-", "--\\-", "---\\-"}
  -- for the sub(sub(sub-items, first one is empty
pageNoPrefixDel    = ""      -- a delimiter for page numbers like "VI-17" -- not used !!!
indexOpening       = ""      -- commands/text after \begin{theindex}
```

The new config file can define own functions for compressing the pagelist for a given entry and for the formatting of the output. They must be called `specialCompressPageList` and `specialGetPageList`.

For example:

```
function specialCompressPageList(pages)
  if (pages[1]["number"] == "") then pages[1]["number"] = " " end
  if (#pages <= 1) then
    pages[1]["number"] = pages[1]["number"]:gsub('-',':~')-- replace "-" with ":~"
    return pages
  end -- only one pageno
  local sortPages = {}
  local roman
  local volume
  local page
  local i
  for i=1,#pages do
    roman = string.gsub(pages[i]["number"],'%U*', '') -- only uppercase to catch VII/1-123f and VII/3-123ff (folium pages)
    if romanToNumber(roman) then
      roman = string.format("%05d",tonumber(romanToNumber(roman))) -- only roman part VII
    else
      roman = ""
    end
    volume = string.gsub(pages[i]["number"],'%a*', '') -- only the number /2 123 or /2-123
    if volume then volume = volume:gsub('-', '%d*', '') end -- delete - char to get /2
    page = string.gsub(pages[i]["number"],'.*-', '')
    page = string.format("%5s",page)
    sortPages[#sortPages+1] = {
      origin = pages[i],
      sort = roman..volume.." "..page } -- no minus between Roman/Volume and first page
  end
  table.sort(sortPages, function(a,b) return a["sort"] < b["sort"] end )

  [...]

  return pages
end
end
```

The above code is a special function which can handle page numbers like VII-17, VIII/2/1-186. Internally exists a function `compressPageList` which is used if no `specialCompressPageList` is defined.

¹The directory `xindex` must be created before saving the file.

verzeichnis

nes von	VII/1 : 215	K	
.....	VII/2/1 : 1003, 1012	Karl	
.....	VII/2/1 : 864, 991, 1048, 1067, 1156	- II., Kaiser	VII/1 : 147
.....	VII/2/1 : 1223, IX/1 : 1228	- III., Kaiser	VII/1 : 149
.....	VII/1 : 514, XI/1 : 515	- IV., Kaiser	VI/1 : 12, VII/1 : 34, 147
.....	VII/1 : 426, 434, 453, 455, 466f.	- V., Kaiser	VI/1 : 84, 284, 654, VI/2 : 708, 1014, 1043, 1131, 1210, VII/1 : 34
		- VI., Kaiser	VII/1 : 296
		- IX., Kaiser	VII/1 : 296
		- X., Kaiser	VII/1 : 149
		- der Große, Kaiser	VI/2 : 987, 989, 1028
Wolfenbüttel		O	
or von, Herzog	VI/1 : 83	Osnabrück	
a, Erzbischof	see Sachsen-Lauenburg	- Heinz von, Bischof	see Sachsen-Lauenburg
		S	
		Schleswig-Holstein	
, römischer Kaiser	VII/2/1 : 904	- Rudolf von, Herzog	VII/2/1 : 758–761, 765
zantinischer Kaiser	VII/1 : 326, 734,	Z	
2/1 : 1011		Zwingl, Haldrich	IX : 479, 692

The .idx file of the above example looks like

```
\indexentry{Karl!V., Kaiser}{VI/2-1210}
\indexentry{Braunschweig-Wolfenbüttel!Karl Viktor von, Herzog}{VI/1-83}
\indexentry{Schleswig-Holstein!Rudolf von, Herzog}{VII/2/1-758}
\indexentry{Schleswig-Holstein!Rudolf von, Herzog}{VII/2/1-759}
[...]
```

The config file xindex-dtk.lua defines a special page output:

```
function specialGetPageList(v,hyperpage) -- Entry table, boolean
  local Pages = {}
[.]
  if (Pages[1]["special"] == nil) or (Pages[1]["number"] == nil) then return "" end
  if #Pages == 1 then
    return "\\relax"..Pages[1]["number"].."\\@nil"
  else
    pageNo = "\\relax"..Pages[1]["number"]
    for i=2,#Pages do
      if Pages[i]["number"] then
        pageNo = pageNo..", "..Pages[i]["number"].."\\@nil"
        Pages[i] = {}
      end
    end
  end
[.]
end
```

The following example runs xindex -c dtk -l de -n <input> and the .idx file looks like

```
\indexentry{BährendtsenElke@Elke Bährendtsen!\Email {elke"@xyz.de}}{14}
\indexentry{JacekJonasson Jared@Jonasson Jared Jazek!\Email {mail"@jones.net}}{20}
[...]
```

enliste

ährendtsen yz.de	[14]	Eike Schulter Haussteig 15 36396 Stuttgart eike.schulter@kabel.de	[40]
Fannt hallee 74 neu-Isenburg fannt@gmxnet.de	[48]	Markus Severs siehe Seite 1	[4]
on Jared Jazek ones.net	[20]	Herbert Voß Wasgensteig 12 10127 Potsdam herbert@xyz.de	[3, 5]
Koon -Links-Weg 16 Neckar z.org	[24, 31]	Michael Ziegenda Lokostr. 19 20713 Kallin ziegenda@mail.com	[9]

There are three predefined sublabels for \subitems. The program itself can handle more, there is no limit for xindex.

Index

F

foo, 1
 - bar, 1
 - baz, 1
 — foobar, 1
 — Kuba, 1

6. hyperref

Using the package hyperref is no problem:

Index

F	Oberin, 1
Fluss	Obstler, 1
- Oder, 1	Ödem, 1
	Oder, 1
G	oder, 1
Göbel, 1	Oder, <i>siehe auch</i>
Goethe, 1f	Fluss
Goldmann, 2	Öl, 1
Göthe, 1	ölen, 1
Götz, 1	Oligarch, 1
	Öresund, 1, 1
O	Ostern, 1
Ober, 1	Österreich, 1

The following example fixes a problem with hyperref and escaping the | character, e.g."|. In such a case hyperref ignores the vertical bar. With the optional parameter -f | --fix_hyperref, which is still experimental,

xindex tries to fix this problem. However, instead of using this problematic vertical character, you can use \textbar, which also solves the problem.

Index

Symbols	<code>'</code> , 1	<code>*</code> , 1	<code>=</code> , 1	F
<code>_</code> , 1	<code>”</code> , 1	<code>/</code> , 1	<code>></code> , 1	Foo, 2
<code>-</code> , 1	<code>”</code> , 2	<code>\</code> , 2	<code> </code> , 1f	
<code>,</code> , 1	<code>(</code> , 1	<code>&</code> , 1	<code>~</code> , 2	P
<code>;</code> , 1	<code>)</code> , 1	<code>#</code> , 1	<code>\$</code> , 1	Post, 2
<code>:</code> , 1	<code>[</code> , 1	<code>%</code> , 1		
<code>!</code> , 1	<code>]</code> , 1	<code>‘</code> , 1		
<code>?</code> , 1	<code>,</code> , 2	<code>^</code> , 2		Z
<code>.</code> , 1	<code>,</code> , 2	<code>+</code> , 1	Numbers	Z, 2
<code>...</code> , 2	<code>@</code> , 1	<code><</code> , 1	123, 2	Zeppelin, 2

7. Including commands into the .idx file

The command \addtocontents doesn’t work for the index file. With the L^AT_EX package xindex (same name as the Lua program xindex) defines a macro \writeidx which writes its argument into the .idx file. This can be useful to insert a pagebreak/columnbreak before a new letter in the output of the index file:

```
\documentclass{article}
\usepackage{makeidx}
\makeindex
\usepackage{xindex}
\begin{document}

\index{foo}foo and
\writeidx{\clearpage}
\index{bar}bar

\printindex
\end{document}
```

Such commands are then taken into account by the program xindex. With the often used program makeindex such commands are ignored. In the following example we put an horizontal line after the first entry:

Index

B	
bar, 1	
F	
foo, 1	
G	
gex, 1	

8. Headings

By default the output uses the English headings: *Symbols*, *Numbers*, and *A* ... There are three predefined languages en, de, and fr. The definition is in the file `xindex-cfg-common.lua` (see also section 2 on page 5). It can easily be extended for other languages. Sometimes the headers are not needed, for example in a name list. With the optional argument `-n` or `--noheadings` the created `.ind` file has only the vertical space between different first letters:

```
xindex -n <file>
```

Index

bar, 1–4

foo, 1f, 2, 3, 4

The headings are printed by default as `\textbf`. This can be changed in the config file by setting the variable `idxnewletter`, for example: `idxnewletter = "\\textit"`. If you need some more code here then define an own macro for it, which can be seen in the following example. It has an own config file `xindex-header.lua` which has the line

```
idxnewletter = "\\idxnewletter"
```

In the documents preamble there is the definition:

```
\newcommand\idxnewletter[1]{\textbf{\textit{#1}}}
```

Index

Symbols

, 1

”, 1

<file.idx>, 1

Numbers

123, 1

A

At (@), 1

E

exclaim (!), 1

V

Vertical bar (|), 1

9. Automatic index creation

With package `xindex` one can define several different index files, e.g. an index of names. With the optional argument `imakeidx` the package itself loads `imakeidx` and adds the program `xindex` as the default program to `imakeidx`.

Index of names

Fairbairns, Robin,
Feuerstack, Thomas,

Kew, Jonathan,
Knuth, Donald,
Kohm, Markus,

Lamport, Leslie,
Lemberg, Werner,

Mittelbach, Frank,

Niepraschk, Rolf,

Pégourié-Gonnard, Manuel,
Preining, Norbert,

Roux, Élie,

Schenk, Christian,

Thành, Hàn Thê,
Theiling, Henrik,
Tobin, Geoffrey,

Volovich, Vladimir,

Wilson, Peter,

You have to run $\LaTeX$ with the `--shell-escape` option to run `xindex` from within the $\LaTeX$ document.

10. Labels

By default `xindex` creates labels in the index for the symbols, numbers, and other parts (letters) to which one can refer. with `\ref{label}`. The labels are named `L-xindex-<name>`. The prefix `L` can be changed by the config file. `<name>` may be symbols, numbers, or `A` (a letter). For example

```
\begin{theindex}  
\par\textbf{Symbols}\label{L-xindex-symbols}  
\nopagebreak[4]  
  \item @, \hyperpage{3}  
  \item (, \hyperpage{3}  
  \item !, \hyperpage{3}  
  
\indexspace  
\textbf{A}\label{L-xindex-A}  
[...]
```

The labels can be used to create a reference to a specific part in the index, for example the letter `X` is in the index on page [23](#) (`\pageref{L-xindex-X}`).

With the optional argument `-b` for the run of `xindex` one can suppress the creation of the labels, e.g. `xindex -b -l fr ...`

11. Demerits

- For more than 5000 entries in the `.idx` file the internal Lua function for sorting may take some time.
- The `.idx` file is not checked for $\LaTeX$ errors in the argument of `\indexentry`.

A. Examples

Correct french sorting with UCA:

Index

B

bohème, 1
Bohême, 1
Bohémien, 1

C

caennais, 1
cæsium, 1
CÆSIUM, 1
cafard, 1
coefficient, 1, 3
CŒUR, 1
coexistence, 1
cote, 1
Cote, 1
COTE, 1
Côte, 1
côte, 1
CÔTE, 1

coté, 1

Coté, 1

COTÉ, 1

côté, 1

Côté, 1

CÔTÉ, 1

coté, 2 sq.

cœur, 1 sq.

E

Élève, 1

élevé, 1

G

gène, 1

gêne, 1

gêné, 1

L

lèse, 1

lésé, 1

M

MÂCON, 1
maçon, 1

P

pêche, 1
péché, 1
PÉCHÉ, 1
péché, 2
pécher, 1
pêcher, 1

R

relève, 1
relevé, 1
révèle, 1
révélé, 1

Index

A

alpha, 1

- beta, 1

- gamma, 1

F

first level, 1

- second level, 1

Index

A

A

- Test, 1

B

B

- Test, 1

Index

F

foo

- bar1
 - baz1, 1
 - baz2, 1
- bar2
 - baz1, 1
 - baz2, 1

Index

Symbols

!, 3
", 3
@, 3
\{, 3
\}, 3
>>, 3

A

accented characters, 3
\addtocontents, 17
argument, 2
aux file, 5

B

babel package, 5
\braceLeft, 3
\braceRight, 3

C

columnbreak, 17
config file, 15

D

data element, 1
description environment, 13

E

entry name, 1
escape character, 3

H

hyperref package, 16

I

.idx file extension, 1, 15, 17, 20
imakeidx package option, 18
imakeidx package, 18
.ind file extension, 1, 18
\index, 3
index of names, 18
\indexentry, 20

K

kpsewhich program, 14

L

label, 19
language, 2, 4, 18
L^AT_EX errors, 20
.lua file extension, 13
lua-uca-languages.lua file, 7

M

makeindex program, 1, 17

N

numbers, 19

O

output, 3

P

page number, 1, 14
pagebreak, 17
\pageref, 19
polyglossia package, 5
\printindex, 3
\providecommand, 3

R

\ref, 19

S

Shell escape, 19
sorting, 3, 20
\subitems, 16
symbols, 19
syntax, 2

T

\textbar, 17
\textbf, 18

U

unicode, 1
UTF-8, 1

W

\writeidx, 17

X

xindex package, 1, 3, 17f
xindex program, 3f, 7, 10, 16–19
xindex-cfg-common.cfg file, 12
xindex-cfg-common.lua file, 18
xindex-cfg-uca.lua file, 7
xindex-cfg.lua file, 13
xindex-dtk.lua file, 15
xindex-HAdW-eKO.lua file, 13
xindex-header.lua file, 18