



USER DOCUMENTATION (ALEPHINO 4.0)

How to create a new index

Table of contents

1Introduction.....	3
2Server control tables.....	3
2.1Index file definition – file etc/marcddata.int.....	3
2.2Internal index definition – file etc/marcindex.int.....	3
2.3External index definition – file etc/marcindex.ext.....	4
2.4Enable search term in OPAC – file etc/alipac.cfg.....	5
2.5Naming of search terms – files etc/marctext.<language>.....	5
3Create or rebuild index.....	6
3.1You have setup a new index.....	6
3.2You have modified index definitions for an existing search term.....	7

1 Introduction

In Alephino, records are indexed by field resp. subfield; several fields can be summarised in an index. A field may be indexed several times. For text fields, phrase or keyword indexing is selectable. The keyword indexing may be affected by the entries in a stop word list. Indexing takes place real-time.

In case you set up an index for a field that already existed you need to run the function "Build Index" from the Alephino Web Service Module/Administration to retrospectively index the previous data. In the following example, we create a new index for the new field with name **A90**, subfield **\$a**. The subfield contains the text of a remark.

2 Server control tables

2.1 Index file definition – file etc/marcdata.int

The files for indices of bibliographic fields are defined in the subdivision *****BIB***** of the section ***Index lists***. For the file number it is very important that it is not being used already, also please make sure that there is enough distance to the last used number as otherwise the number might be used in new releases. As file number we choose 900, the TYPE for an index is always LIST, the index length should correspond to the max length as defined by TERM see below. For date fields, the index length is 4 as dates are saved binary.

```
*****
* Index lists
*****
* Bibliographic
*****
* NOT
FILE=340,TYPE=LIST,INDEXLEN=30
* REM
FILE=900,TYPE=LIST,INDEXLEN=30 ← add line at end of section
...

```

2.2 Internal index definition – file etc/marcindex.int

The indices for bibliographic fields are defined in the section (titindex), the section is subdivided by title, ISBN etc, imprint, series and number and code fields. For each field an index number INDEX is defined. For the index number it is very important that it is not being used already, also please make sure that there is enough distance to the last used number as otherwise the number might be used in new releases. We choose the number 300 for it. As file number FILE we choose 900 (see above). In our example we consider TERM=2 the most suitable and the SOURCE is the field A90\$a.

We place the new entry at the end of the section under a new subdivision *Remark.
 To determine which TERM to choose study the section "Terms" at the top of the file, here the minimum and the maximum length is defined, for text fields IDXWORD translates the index word by word, IDXPHRASE translates the index phrase by phrase. For date fields choose the TERM=1

```
*****
* Terms
*****
TERM=0,TYPE=SORT,MAXLEN=100,ALIGN=L,TRANSL=IDXSORT
* Default for dates
TERM=1
TERM=2,MAXLEN=30,ALIGN=L,STOP=1:1,TRANSL=IDXWORD
TERM=3,MAXLEN=30,ALIGN=L,TRANSL=IDXWORD
TERM=4,MAXLEN=100,ALIGN=L,TRANSL=IDXPHRASE
...

*****
* Search terms
*****
(titindex)
*
* Title Index / Type = Word
*
INDEX=1,SOURCE=130$a,TERM=2,FILE=300          * TIT
INDEX=2,SOURCE=130$d,TERM=2,FILE=300
INDEX=3,SOURCE=130$f,TERM=2,FILE=300
...
INDEX=300,SOURCE=A90$a,TERM=2,FILE=900      ← add line at end of section
...

```

2.3 External index definition – file etc/marcindex.ext

You also need to define an index code, which is used by the clients. For bibliographic fields this is done in the section (titindex). The index code is a three-digit short name. We decided to use index code REM (INDEX=REM), the internal number of the index definition DEFIN in our case is 300, FIND=Y determines that the search aspect is available in the search mode "simple search", with SCAN=Y the index is available in the index search, with ITEMS=Y the index is available in Search for Title, with DIRECT=Y the index is available in the Multi-field search.

Both with ITEMS and with DIRECT you have to be careful:

The search masks for **Search for Title** and **Multi-field search** contain only up to 8 search fields. ITEMS=Y and DIRECT=Y should therefore only be defined each for 8 search aspects in total. This means if you choose e.g. for your new search aspect ITEMS=Y, another search aspect must be changed from ITEMS=Y to ITEMS=N, as in the standard 8 search aspects are already available in this mask (NAM, SUB, TIT, ISN, IDN, LOC, ACC, BAR).

```

*****
* DEFINITION OF SEARCH TERMS
*****
(titindex)
* All fields index (WRD)
INDEX=WRD,TYPE=G,DIRECT=Y
...

INDEX=GDC,DEFIN=180,FIND=N
INDEX=REP,DEFIN=181,FIND=N
...
INDEX=REM,DEFIN=300,FIND=Y,SCAN=Y      ← add line at end of section
...

```

2.4 Enable search term in OPAC – file etc/alipac.cfg

To make the index available in OPAC, add the search aspect in the section **(SearchTerms)**.

FIND defines that the search aspect is available in the basic search mode, with **SCAN** the index is available for browsing, and with **REFINE** the search aspect is available for modifying a set of records.

```

...
(SearchTerms)
FIND = WRD,TIT,NAM,SUB,PER,COR,MEE,SCO,SME,TOT,GEO,PUB,YEA,SRS,TRM,ISN,REM
SCAN = TIT,PER,COR,MEE,SPE,SCO,SME,TOT,GEO,PUB,PLA,YEA,SRS,ISN,REM
REFINE = WRD,TIT,PER,COR,MEE,SPE,SCO,SME,TOT,GEO,PUB,PLA,YEA,SRS,ISN,REM
...

```

2.5 Naming of search terms – files etc/marctext.<language>

Finally, the index should have a long name – which potentially is language dependent. Index names are defined in the section (Indexes). We call our index "Remark". For English we have to process file etc/marctext.eng:

```

...
(Indexes)
PST = Item process status
REM = Remark      ← add this line
REP = Report No.
...

```

3 Create or rebuild index

This step is not necessary when you set up an index for a newly created field (like the example in this how-to document) or when you set up an index for a field which was never filled in the existing records.

But as soon as a field is concerned which has been already filled in records, these records must be retrospectively indexed!

3.1 You have setup a new index

Both in case this new index was set up with a new, own search term (e.g. setting up a search term „CAT“ for the field CAT\$a) or in case the new index was added to an existing search term (e.g. setting up the field 222\$a “Key Title” as a new source for the title keyword index file), you have to run the function “Build Index” from the Alephino Web Service Module/ Administration to retrospectively index the current data in the concerned field.

Example from file etc/marcindex.int:

```
...
*****
* Search terms
*****
(titindex)
*
* Title Index / Type = Word
*
INDEX=1, SOURCE=130$a, TERM=2, FILE=300          * TIT
...
INDEX=41, SOURCE=830$v, TERM=2, FILE=300
INDEX=45, SOURCE=222$a, TERM=2, FILE=300        ← this was added
...
```

To make out new search term effective the respective index entries have to be added to the internal index file. For this Web-Service “Build index” has to be used as shown below:

Build index

master file:

index number(s):



3.2 You have modified index definitions for an existing search term

Normally you should NOT MODIFY the index definitions for existing search terms!

The predefined index terms do represent rules for index creation in many different ways, so normally there's no need to add more.

In case you still decide to do so, e.g. assign a different TERM, then you must delete the existing index file prior to rebuilding the indices in question:


- Detect internal index file number according to etc/marcdata.int

```
*****
* Index lists
*****
* Bibliographic
*****
* TIT
FILE=300,TYPE=LIST,INDEXLEN=30
...
```

- Delete that file using Web-Service “Delete file”

Delete files

file number(s):



- Detect index numbers that belong to that index file according to etc/marcindex.int

```
*****
* Search terms
*****
(titindex)
*
* Title Index / Type = Word
*
INDEX=1, SOURCE=130$a, TERM=2, FILE=300          * TIT
INDEX=2, SOURCE=130$d, TERM=2, FILE=300
INDEX=3, SOURCE=130$f, TERM=2, FILE=300
...
INDEX=44, SOURCE=830$v, TERM=2, FILE=300
...
```

- Create index file anew using Web-Service “Create index”

Build index

master file:

index number(s):

