

## **Control tables**

From: Swami Wiki

Date: April 28<sup>th</sup> 2011

### **Table of contents**

- 1 General
- 2 Data structure definitions
  - 2.1 Definition database
  - 2.2 Definition file
  - 2.3 Definition tag
  - 2.4 Definition virtual field
  - 2.5 Definition index rule
  - 2.6 Definition index term
  - 2.7 Definition translate table
  - 2.8 Definition link
  - 2.9 Definition link field
  - 2.10 Indication of a record field
  - 2.11 Definition doublet check
  - 2.12 Definition number assignment
- 3 Format definitions
  - 3.1 Definition database
  - 3.2 Definition file
  - 3.3 Definition format
  - 3.4 Definition columns
  - 3.5 Definition search aspect
  - 3.6 Definition display field
  - 3.7 Definition indicator table
  - 3.8 Definition link aspect
  - 3.9 Definition Script
  - 3.10 Definition localization
  - 3.11 Definition name
  - 3.12 Definition Layout
  
  - 3.13 Definition Statistics counter

## General

The data structure and format definitions can be saved in one or more text files.

Lines which have a \* in the first column will be interpreted as comments.

The lines with definitions begin in the first column and must not include blanks; to define texts with blanks, set them in single quotation marks. All text following a blank will be interpreted as a comment.

The maximal length for a line is 99 characters. Following lines are possible and must be initiated with one blank or tab at least. A division of a line must take place between two definition parameters, that means after a comma.

A label will be defined with **(labelname)**

The texts within the format definitions can be swapped out in a file which exists for special languages. In the following description you will find the note **(can be swapped out)**.

## Data structure definitions

### Definition database

The database definitions are included in file **marcdata.int**.

#### Syntax:

```
POOL=name, ACCDEL=glist, ACCDIS=glist, ACCINS=glist, ACCUPD=glist,  
CODE=code, DATABUF=count, FILE=label, INDEXBUF=count, PERIOD=number,  
SETUP=number, USERPERM=number
```

#### Explanation:

POOL= internal name of the database (10 characters maximum)  
ACCDEL= list of user groups be allowed to delete records (the groups will be arranged with + between, as 1+2+5; default: all)  
ACCDIS= list of user groups be allowed to display records (the groups will be arranged with + between, as 1+2+5; default: all)  
ACCINS= list of user groups be allowed to add records (the groups will be arranged with + between, as 1+2+5; default: all)  
ACCUPD= list of user groups be allowed to update records (the groups will be arranged with + between, as 1+2+5; default: all)  
CODE= internal character code (MAB stands for MAB2 character set)  
DATABUF= no. data blocks for data block buffer (default: 10)  
FILE= address of file definitions  
INDEXBUF= no. data blocks for index block buffer (default: 10)

PERIOD = internal file no. for circulation periods  
SETUP = internal file no. for setup parameters  
USERPERM= internal file no. for user permissions  
USERTAB= internal file no. for customer tables

### Example:

```
*****  
* Definition Pools  
*****  
POOL=MARC, FILE=filtab, CODE=MAB, USERPERM=006, SETUP=007, PERIOD=025,  
USERTAB=008
```

---

## Definition file

The definitions are included in file **marcdata.int**.

### Syntax:

```
FILE=number, ACCDEL=glist, ACCDIS=glist, ACCINS=glist, ACCUPD=glist,  
AUTONUM=label, DATALEN=length:VAR | FIX, DOUBLE=label, INDEX=label,  
INDEXDAT=length, INDEXLEN=length:VAR | FIX, ITEMLEN=length,  
LINK=label, MAIN=number, TAG=label, TYPE=type:itype
```

### Explanation:

FILE= internal file number (max. 999)  
ACCDEL= list of user groups be allowed to delete records (the groups will be arranged with + between, as 1+2+5; default: all)  
ACCDIS= list of user groups be allowed to display records (the groups will be arranged with + between, as 1+2+5; default: all)  
ACCINS= list of user groups be allowed to add records (the groups will be arranged with + between, as 1+2+5; default: all)  
ACCUPD= list of user groups be allowed to update records (the groups will be arranged with + between, as 1+2+5; default: all)  
AUTONUM= address of number assignment  
DATALEN= maximum / fixed data length (default: 3000:VAR)  
DOUBLE= address of doublets check table  
INDEX= address of index definitions  
INDEXDAT= length data portion on index level (default: 4 for list files, only for DB optimizer !)  
INDEXLEN= maximum / fixed index length (default: 4:FIX) (only fixed index length possible yet; maximum 64)  
ITEMLEN= length of entry in list file (default: 4, only for DB optimizer !)  
LINK= address of link definitions  
MAIN= file number mainfile (only for DB optimizer !)  
TAG= address of field definitions

TYPE= record type:internal type  
RECORD master file with placing of IDN  
DATA data file without IDN  
LIST list file  
(internal type= NORMFILE, MAINFILE, SUBFILE only for DB optimizer !)  
(default= RECORD:NORMFILE)

### Examples:

For master files:

```
(filtab)
*** Title records ***
FILE=011,TYPE=RECORD,TAG=tit,INDEX=titindex,LINK=titlink,DOUBLE=titdouble,DATALEN=10000
```

For link lists:

```
*** BIB-BIB ***
FILE=100,TYPE=LIST,INDEXLEN=4
```

For index lists:

```
*** TIT ***
FILE=300,TYPE=LIST,INDEXLEN=30
```

For special data files:

```
*** Setup-Parameter ***
FILE=007,TYPE=DATA,INDEXLEN=18,DATALEN=300
```

---

## Definition tag

The tag definitions are included in file **marcdata.int**.

### Syntax:

```
TAG=name (dimension) ,DELIM=x,INDIC1=text,INDIC2=text,MAX=count,
SUB=label, TYPE=type (length),VIRT=label
```

### Explanation:

TAG= internal tag name (3 characters maximum) dimension = number of repetitions allowed (\* unlimited; default: 1)

DELIM= delimiter  
 INDIC1= text which includes the first indicators allowed  
 INDIC2= text which includes the second indicators allowed  
 MAX= [reserved for further development (array-fields)]  
 SUB= address of definitions for subfields  
 TYPE= field type (optional field length)  
     CHAR          text field (length = maximum length)  
     DATE          date  
     DATETIME      date/time  
     IDNR          Identification number  
     MAIN          main field for subfields  
     NUMBER        Numerical field (length = position before comma:position after  
                   decimal point)  
     URL           URL (length = maximum length)  
     (default:  
     CHAR)  
 VIRT= address definition virtual field [reserved for further development]

**Example:**

```

*
* Number and Code Fields
*
TAG=010,TYPE=MAIN,SUB=sub010,INDIC1=' ',INDIC2=' '
TAG=013(*),TYPE=MAIN,SUB=sub013,INDIC1=' ',INDIC2=' '
TAG=015,TYPE=MAIN,SUB=sub015,INDIC1=' ',INDIC2=' '
TAG=016(*),TYPE=MAIN,SUB=sub016,INDIC1=' 7',INDIC2=' '
TAG=017(*),TYPE=MAIN,SUB=sub017,INDIC1=' ',INDIC2=' '

TAG=CAT(10),TYPE=MAIN,SUB=subach
TAG=005,TYPE=DATETIME
TAG=105,TYPE=NUMBER(9:2)
*
(sub010)
TAG=a
TAG=b(*)
TAG=z(*)
TAG=8(*)
  
```

---

**Definition virtual field**

[reserved for further development]

**Syntax:**

```

VIRT=tag$sub,ALIGN=L | R,FILL=x,FORM=A|B|X,LENG=length,NOEX=NONE|OBL,TEXT=text
  
```

---

## Definition index rule

The index definitions are included in **marcdata.int**.

### Syntax:

```
INDEX=number, FILE=number, PREFIX=text, SOURCE=tag$sub, TERM=number,  
NFINDIC=number
```

### Explanation:

INDEX= internal index number  
FILE= file number of index file  
PREFIX= constant text in front of index term  
SOURCE= source field for indexing  
TERM= Number of index term definition  
NFINDIC= position of the indicator to define the start position for indexing field contents (MARC)

### Examples:

```
INDEX=1, SOURCE=130$a, TERM=1, FILE=300  
INDEX=50, SOURCE=245$a, TERM=3, FILE=301, NFINDIC=2
```

---

## Definition index term

The index term definitions are included in file **marcdata.int**.

### Syntax:

```
TERM=number, ALIGN=align, FILL=x, INDIC=YES | NO, MAX=count, MAXLEN=length, MINLEN=length,  
STOP=file:number, TRANSL=name, TYPE=type
```

### Explanation:

TERM= internal term number  
ALIGN= term alignment:  
LEFT left-justified  
RIGHT right-justified  
(default: LEFT)

FILL= fill character:  
 SPACE blank  
 NULL null  
 0 null  
 LOW low value  
 HIGH high value  
 (default: SPACE)

INDIC= Y: Indexing with indicator  
 N: Indexing without indicator

MAX= max. number terms for selection (default: 200)

MAXLEN= max. term length (default: 240)

MINLEN= min. term length (default: 1)

STOP= file number and number of list with stop words

TRANSL= name of translate definitions

TYPE= term type:  
 ASCII  
 SORT preparation of sort term  
 IDLIST ID-list for link of strings  
 BINARY [reserved for further development]  
 (default: ASCII)

### Examples:

```
*****
* Terms
*****
TERM=1,TYPE=ASCII,MAXLEN=30,MINLEN=1,ALIGN=L,STOP=1:1,TRANSL=IDXWORD
TERM=2,TYPE=ASCII,MAXLEN=30,MINLEN=1,ALIGN=L,TRANSL=IDXWORD
TERM=3,TYPE=ASCII,MAXLEN=60,MINLEN=1,ALIGN=L,TRANSL=IDXPHRASE
TERM=4,TYPE=SORT,MAXLEN=30,MINLEN=1,ALIGN=L,TRANSL=IDXPHRASE
TERM=5,TYPE=ASCII,MAXLEN=20,MINLEN=1,ALIGN=L,TRANSL=IDXPHRASE
TERM=6,TYPE=ASCII,MAXLEN=14,MINLEN=1,ALIGN=L,TRANSL=IDXISN
TERM=7,TYPE=ASCII,MAXLEN=30,MINLEN=1,ALIGN=L,TRANSL=IDXNAME
TERM=8,TYPE=ASCII,MAXLEN=4,MINLEN=4,ALIGN=L,TRANSL=IDXWORD
TERM=9,TYPE=ASCII,MAXLEN=30,MINLEN=1,ALIGN=L,TRANSL=IDXNUMB
TERM=10,TYPE=ASCII,MAXLEN=15,MINLEN=3,ALIGN=L,TRANSL=IDXPHRASE
TERM=11
TERM=12,TYPE=ASCII,MAXLEN=18,MINLEN=1,ALIGN=L,TRANSL=IDXPHRASE,INDIC=YES
TERM=13,TYPE=ASCII,MAXLEN=4,MINLEN=1,ALIGN=L,TRANSL=IDXWORD
TERM=14,TYPE=IDLIST,MAXLEN=40
```

---

### Definition translate table

The translate tables are signed with the ending **\*.trl** and are included in the directory which is indicated within the parameter **PATH=** .

### Syntax:

```
TRANSL=name, PATH=externfile[, TYPE=UTF8]
```

### Explanation:

TRANSL= internal name of translate table  
PATH= external file name  
TYPE=UTF8 input data must meet the UTF8 character set

### Examples:

```
(indextrans)  
TRANSL=IDXWORD, PATH=../etc/utf2word.trl, TYPE=UTF8  
...  
TRANSL=EXTTOMAB, PATH=../etc/utf2mab.trl, TYPE=UTF8  
TRANSL=MABTOEXT, PATH=../etc/mab2utf.trl
```

### Format of the translate table:

Lines with a \* at the beginning will be interpreted as comments.

One translate entry per line :

Source character (decimal) TAB destination character (decimal)

More characters can be arranged with a + between. Characters that are not indicated will be translated 1:1 .

Translation tables with UTF8-input :

**Default** Characters that are not indicated will be suppressed.

**LETPASS=ASCII** All ASCII-characters (< 128 (decimal) will pass 1:1 if not indicated explicitly.

**LETPASS=ALL** All characters will pass 1:1 if not indicated explicitly.

Any other translation tables :

**Default** Characters that are not indicated will pass 1:1.

**LETPASS=UTF8** Only valid UTF8-characters will be passed to the output.

There are the following **specified values** for the destination characters:

000 source characters will be eliminated

001 separator for indexing

002 beginning elimination for indexing

003 end elimination for indexing

004 beginning and end elimination for indexing

005 adjust stop word for indexing

006 compound on indexing

007

008 wildcard for retrieval  
009 truncation for retrieval

### Examples:

```
*****
* NAME      : utf2word.trl
* PURPOSE   : Translation UTF8 to Word Index
* VERSION   : Alephino 3.1
* IDENTKEY  : 1.2
* UPDATE    : 08/11/08
* REMARKS   :
*****
LETPASS=ASCII
032 001 * Leerzeichen
033 001 * ! Ausrufungszeichen
034 000 * " Doppeltes Anfuhrungszeichen oben
042 009 * * Stern, Asterisk
045 006 * - Minus, Bindestrich
063 008 * ? Fragezeichen
195+132 065+069 * Ä Umlaut
```

---

### Definition link

The link definitions are included in file **marclink.int**.

### Syntax:

```
LINK=number,DESTREC=CHECK | NEW | OPT,FILE=number,
FORWARD=tag$sub,INDIC=YES | NO,LINKTAG=label,PUBLISH=YES | NO,
LIST=number,REFTAG=tag$sub,SEQU=number:number,SUB=YES | NO,SUBLINK='listofsubfields'
```

### Explanation:

LINK= internal link number  
DESTREC= destination record processing:  
CHECK check destination record on linking  
NEW add destination record  
OPT destination record / link is optional  
FILE= number of destination file  
FORWARD= field to forward link to destination file  
INDIC= YES: take over indicator to destination record  
LINKTAG= address of link fields  
PUBLISH= register updates for publishing (Primo-Pipe)  
LIST= number of link list  
REFTAG= reference field in destination file  
SEQU= number of sequence file : number of sequence  
SUB= YES: take all defined subfields to destination record  
SUBLINK= list of subfields to be taken to destination record

## Examples:

```
(titlink)
*** BIB ***
LINK=1, FILE=11, LIST=100, REFTAG=245$a, DESTREC=CHECK, LINKTAG=titlink01
*** Names ***
LINK=5, FILE=12, LIST=110, REFTAG=100$a, DESTREC=NEW, LINKTAG=titlink05,
    INDIC=YES
*** Subjects ***
LINK=9, FILE=13, LIST=120, REFTAG=148$a, DESTREC=NEW, LINKTAG=titlink09,
    INDIC=YES, SUB=YES
*** Thesaurus ***
LINK=16, FILE=40, LIST=161, REFTAG=101, FORWARD=230$L, DESTREC=CHECK, LINKTAG=titlink16

(mexlink)
LINK=1, FILE=11, LIST=140, REFTAG=245$a, DESTREC=CHECK, LINKTAG=mexlink01,
    SEQU=2:1
```

---

## Definition link field

The link field definitions are included in the file **marclink.int**.

### Syntax:

```
LINKTAG=tag$sub, BACK=tag$sub, DESTTAG=tag$sub, SEQTAG=tag$sub,
    SORT=tag$sub, SORT2=tag$sub, SOURCE=tag$sub, SUBLIB=tag$sub
```

### Explanation:

LINKTAG= field which contains the ID of the link  
BACK= field for back link in case of 1:1 link  
DESTTAG= destination field in destination file for field to field link  
SEQTAG= field which contains the sequence number  
SORT= field for sorting link list  
SORT2= alternate field for sorting link list  
SOURCE= field for creation link  
SUBLIB= field which contains the sub library

### Examples:

```
(titlink01)
LINKTAG=BIB$L, SOURCE=BIB$a, SORT=BIB$b
*
(titlink02)
LINKTAG=LKR$L, SOURCE=LKR$a, BACK=LKR$L
*
(titlink05)
LINKTAG=100$L, SOURCE=100$a
*
(mexlink01)
LINKTAG=BIB$L, SOURCE=BIB$a, SEQTAG=BIB$s
```

---

## Indication of a record field

The following notation will be used to address a single field inside the definition tables.

### Syntax:

```
Tag [ .TagOrder ] [ /Indicator ] [ $SubTag [ .SubTagOrder ] ]
```

The indicator blank can be defined with an underline    .  
TagOrder 999 stands for the last repetition of a field.

---

## Definition doublet check

The doublet checks are defined in the file **marcdoubl.int**.

### Syntax:

```
DOUBLE=tag$sub,FOUND=YES | COMPARE, INDIC=YES | NO,  
NOTFND=NO | CONTINUE,
```

### Explanation:

DOUBLE= field to check

FOUND= if this parameter exists, the system will execute an internal retrieval for the indicated field

YES records will be found, they count as doublets

COMPARE if records will be found, the fields without the definition FOUND or NOTFND will be compared in addition

INDIC=

YES indicator takes part in compare

NO indicator doesn't take part in compare

(default: NO)

NOTFND=

NO if records won't be found, no doublet will be identified

CONTINUE if records won't be found, the next entry will be checked

(default: CONTINUE)

### Example:

```
*** BIB ***
(titdouble)
DOUBLE=001, FOUND=YES
DOUBLE=010$a, FOUND=YES
DOUBLE=016$a, FOUND=YES, INDIC=YES
DOUBLE=020$a, FOUND=YES
DOUBLE=022$a, FOUND=YES
DOUBLE=024$a, FOUND=YES
DOUBLE=027$a, FOUND=YES
DOUBLE=245$a, FOUND=COMPARE, NOTFND=NO
DOUBLE=100$L
DOUBLE=110$L
DOUBLE=111$L
DOUBLE=130$L
DOUBLE=260$c
```

---

## Definition number assignment

The fields for automatic number assignment are defined in the file **marcnumb.int**.

### Syntax:

```
AUTONUM=number, FILE=number, TAG=tag$sub
```

### Explanation:

AUTONUM= internal definition number  
FILE= file number of number definitions  
TAG= field for number assignment

### Example:

```
(mexnum)
AUTONUM=1, TAG=100$a, FILE=4
AUTONUM=2, TAG=107$a, FILE=4
AUTONUM=3, TAG=115, FILE=4
```

## Format definitions

### Definition database

The databases are defined in the file **marcfile.ext**.

### Syntax:

```
POOL=name, DEFIN=name, FILE=label, COUNT=label,  
LONG=name, MAXLST=count, NAMI=label, SCRIPT=label,  
DATEFORM=number, TIMEFORM=number
```

### Explanation:

POOL= short name of the database (one-digit)  
DEFIN= Internal name of the database  
FILE= address of the file table  
NAMI= address of the name table internal/external  
SCRIPT= address of the script definitions  
COUNT= address of the table for statistics counter  
LONG= text for detailed name of the database (can be swapped out)  
MAXLST= max. number of search results in a list [presently unlimited]  
DATEFORM= no. format for date (default: 1; others are not valid presently)  
TIMEFORM= no. format for time (default: 1; others are not valid presently)

### Example:

```
(pooltab)  
POOL=M, DEFIN=MARC, FILE=filtab, NAMI=names, COUNT=statcount, SCRIPT=scripttab
```

---

## Definition file

The files are defined in the file **marcfile.ext**.

### Syntax:

```
FILE=name, DEFIN=number, INDEX=label, LINK=label, LOCATE=label, LONG=name,  
SFX=label, UPDATE=label, FILTER='query', FORM=reference, SUBSORT=N
```

### Explanation:

FILE= short name of file (three characters)  
DEFIN= Internal number of the file  
INDEX= address of search aspects table  
LINK= address of link aspects table  
LOCATE= address of locate table

LONG= text for detailed name of the file (can be swapped out)  
 SFX= address of SFX locate table  
 UPDATE= address of update formats (not yet used)  
 FILTER= CCL-query for "virtual" file  
 FORM= name of reference file for display- and print formats of belonging to a "virtual" file  
 SUBSORT=N subfields will be preserved in their input order. This is important for MARC fields. (Default: Y)

### Examples:

```
(filtab)
FILE=TIT,DEFIN=011,INDEX=titindex,LINK=titlnk,LOCATE=titloc,SFX=titsfx,
  SUBSORT=N
FILE=TIJ,DEFIN=011,INDEX=titindex,LINK=titlnk,LOCATE=titloc,SFX=titsfx,
  FILTER='JHR=2000 < 2005',FORM=TIT
```

---

## Definition format

The subsequent definitions are located in files with the ending **.ext**.

### Syntax:

```
FORM=type,FILE=name,COLUMN=label,COMMAND=text,HEAD=YES | NO,LAYOUT=number,LINK=name,
  LIST=name,LONG=name,NAME=name,PAGEFOOT=label,PAGEHEAD=label,RECFOOT=label,
  RECFORM=FIELD,RECHEAD=label,SUPPRESS=text,TAG=label,TYPE=printtype,STYLE=stylesheet
```

### Explanation:

FORM= format type:  
 DISPLAY Formatting of a single record (for display)  
 LIST Formatting of a list of records (multicolumn display)  
 PRINT Formatting of records for printing and E-mail dispatch  
 TABLE Definition of a table  
 SORT Composition of a sort key from a record  
 FILE= Master file  
 COLUMN= Address of the columns table  
 COMMAND= Command (for printing and others)  
 HEAD= YES: with header NO: without header  
 LAYOUT= number of the layout definition

LINK= link name for linked file  
 LIST= format name for linked file  
 LONG= text for detailed name of the format (can be swapped out)  
 NAME= internal name of the format  
 PAGEFOOT= address of page footer (print formats)  
 PAGEHEAD= address of page header (print formats)  
 RECHHEAD= address record header (print formats)  
 RECFOOT= address record footer (print formats)  
 RECFORM= FIELD for field format (with field name)  
 SUPPRESS= elimination of special characters:  
     NSO character which indicates no sorting  
     NST character which cancels stop words  
     DIA diakritika which can't be displayed  
     PRO special characters which can't be displayed  
     (More entries than one: SUPPRESS=NSO:NST:DIA:PRO)  
 TAG= field table for full format  
 TYPE= For print formats in combination with COMMAND:  
     PLAIN for simple plain text output  
 STYLE= XSL-stylesheet for HTML-formatted output (without ending **.xsl**)

### Examples:

Full format:

```

FORM=DISPLAY, FILE=TIT, NAME=001, RECFORM=FIELD, COLUMN=colfull,
LONG=FULL, TAG=titfull
FORM=DISPLAY, FILE=TIT, NAME=003, COLUMN=cardform,
LONG=ISBD, SUPPRESS=DIA:NST:PRO
  
```

List format:

```

FORM=LIST, FILE=TIT, COLUMN=titlwww, NAME=WWW,
SUPPRESS=DIA:NSO:NST:PRO
  
```

Print format:

```

FORM=PRINT, FILE=TIT, LONG=LMEX, LINK=MEX, LIST=ITMLIST, STYLE=list_mex,
PAGEHEAD=mexpagehdr, RECHHEAD=mexlisthdr, RECFOOT=recftr, NAME=011
FORM=PRINT, TAG=usercard, NAME=CARDW, COMMAND=WORD, TYPE=PLAIN
  
```

Sort format:

```

FORM=SORT, FILE=TIT, LONG=STITLE, TAG=tits2, NAME=002
  
```

Table:

```
FORM=TABLE,COLUMN=pccol,NAME='ITEM-MATERIAL',TAG=material
```

---

## Definition columns

### Syntax:

```
COLUMN=number,HEAD=text,LAYOUT=number[;number],  
LAYOUTH=number,TAG=label,SOURCE=tag,WIDTH=count
```

### Explanation:

COLUMN= no. of the column  
HEAD= text for column header / header or footer (can be swapped out)  
LAYOUT= no. of the layout definition, where applicable the number of the alternate layout definition additional  
LAYOUTH= no. of the layout definition for header (if deviating)  
TAG= address of the field table  
SOURCE= direct reference to a record field (instead of TAG=)  
WIDTH= Column width (in percent)

### Examples:

```
COLUMN=1,TAG=circtag,LAYOUT=7,WIDTH=100,HEAD=056HEADTIT  
COLUMN=2,LAYOUT=3;2,WIDTH=085,HEAD=CHKHEAD02  
COLUMN=3,SOURCE=803.1,LAYOUT=7,WIDTH=60,HEAD=014HEAD03
```

---

## Definition search aspect

The search aspect definitions are included in file **marcindex.ext**.

### Syntax:

```
INDEX=name,CASCADE=YES | NO,DEFIN=number,DIRECT=Y | N,  
  FIND=Y | N,FAM=name,FORM=label,INDIC=indicator,ITEMS=Y | N,  
  LINK=link,LONG=name,SCAN=Y | N,TAB=label,TYPE=type,SORT=name
```

## Explanation:

INDEX= short name of the search aspect (three characters)  
CASCADE= YES: YES: search query over links: index file is used recursive  
DEFIN= internal number of the index definition  
DIRECT= YES: index available in .multi-field. query form (default: NO)  
FIND= YES: index available in .simple. query form (default: YES)  
FAM= name of the family search aspect  
FORM= address of preparing the record display  
INDIC= indicator [presently not used]  
ITEMS= YES: index available in alternative query form for bibliographic records  
(default: NO)  
LINK= indication for search over linked files: [file number:]link number[:aspect name]  
LONG= text for detailed name of the search aspect (can be swapped out)  
SCAN= YES: index available in browse query (default: NO)  
TAB= address of short/long table  
TYPE= G: 'GOOGLE'-like search query  
SORT= name (number) of the sort format for sorting the records assigned to an index  
entry with the F3 resp. Ctrl\_F3 - browse list in the GUI catalog editor.

## Examples:

```
(titindex)  
INDEX=WRD,TYPE=G,DIRECT=Y  
INDEX=NAM,TYPE=F,SCAN=N,DIRECT=Y,ITEMS=Y  
INDEX=SUB,TYPE=F,SCAN=N,DIRECT=Y,ITEMS=Y  
INDEX=IMP,TYPE=F,SCAN=N,DIRECT=Y  
INDEX=TTT,DEFIN=1,FAM=WRD,SCAN=Y,ITEMS=Y,DIRECT=Y  
INDEX=SRS,DEFIN=100,FAM=WRD,SCAN=Y  
INDEX=PLA,DEFIN=140,FAM=WRD,SCAN=Y  
INDEX=PUB,DEFIN=145,FAM=WRD,SCAN=Y  
INDEX=YEA,DEFIN=146,SCAN=Y,DIRECT=Y  
INDEX=ISN,DEFIN=160,SCAN=Y,ITEMS=Y,DIRECT=Y  
INDEX=IDN,SCAN=N,DIRECT=Y,ITEMS=Y  
INDEX=LNG,DEFIN=182,SCAN=Y  
INDEX=PER,LINK=5,FAM=WRD,SCAN=Y  
INDEX=SME,LINK=12;MEE,FAM=WRD,FIND=N,SCAN=Y  
...  
(subindex)  
INDEX=TPP,DEFIN=380,LONG=TTP013,SCAN=Y,FORM=sub650c,SORT=090
```

---

## Definition display field

The syntax described in subsequent chapter is valid for all display- and print formatting of record fields.

### Syntax:

```
TAG=source,AFTER=text,BEFORE=text,CHARS=allowedchars,CASCADE=YES | NO,
CMPFLD=EQ | NE,CMPIND=EQ | NE,CMPLEN=length,COMPARE=text,
DELIM=text,DFORM=A | G | S | T | Y,EXIST=YES | NO, FILL=x,FILTER=SUBLIBREC,
FORM=label,INDEL=text,INDIC=indicator,INDPOS=A | B, INDTAB=label,
INDTEXT=YES | NO,INFO=text,LENG=length,LIMIT=number,LINK=link,
LOG=O | A | X, NAME=name,NOEX=YES | NO, SIGN=sign,SORT=ASC | DESC,
START=position,SUB=label,TAB=label,TEXT=text,TYPE=type,XTAG=xmlname
TABNO=filename:tablenumber
```

### Explanation:

TAG= source of preparing:  
field of record  
SYSID for record identification number  
RECNO for actual record number (within a set)  
RECALL for total number of records (of a set)  
SYSDATE for current date  
SYSTIME for current time  
USER for user name  
RECLIST for individual display of linked records  
COUNT for number of linked records  
ADDRESS for library address (depending on sub-library)  
TABLE reference to customer defined table stored in database (s. TABNO)  
000 else (for texts or sub-formats)

AFTER= text after entry (can be swapped out)  
BEFORE= text before entry (can be swapped out)  
CHARS= set of characters allowed for entry  
CASCADE= for TAG=COUNT: link will be applied recursive  
CMPFLD= operator for field content comparison (s. CMPLEN, COMPARE)  
CMPIND= operator for field indicator comparison (s. INDIC)  
CMPLEN= length for field content comparison (s. CMPFLD, COMPARE)  
COMPARE= text for field content comparison (s. CMPFLD, CMPLEN)  
DELIM= delimiter text (can be swapped out)  
DFORM= display format for fields with type DATE:  
A american format MM/DD/YYYY  
G german format DD.MM.YYYY  
S sort format YYYYMMDD  
T format as per template **Date=** from alephino.cfg  
V only valid date entry (i.e. for borrower addresses)  
Y age in years  
(standard: S)

Display format for numerical fields:

N without leading zeroes

EXIST= YES: check only if field exists, no output  
FILL= fill character on fixed length (s. LENG, TYPE)  
FILTER= SUBLIBREC: display field only if sub-library record=sub-library user  
FORM= address of sub format (TAG=000)  
INDEL= text for indicator separator [presently not used]  
INDIC= indicator (s. CMPIND)  
INDPOS= indicator position: A = after, B = before field content [presently not used]  
INDTAB= address for indicator table [presently not used]  
INDTEXT= [presently not used]  
YES replace indicator by text (s. INDTAB)  
NO don't replace indicator by text  
INFO= text for additional information (embedded in apostrophes)  
**ACC\_LINK,ASP** Link to browse list with field content as start value.  
(ASP stands for short name of search aspect.)  
**LKR\_LINK** Link to linked record for **distinct** assignment.  
**SET\_LINK,ASP** Link to linked records. (ASP stands for short name  
of link aspect.)  
**ALI\_LINK,[ASP],Linktag** Link to distinct linked record and browse list with  
field content as start value.  
Joins the abilities of ACC\_LINK and LKR\_LINK.  
Deviating from LKR\_LINK this link can point to  
different master files. Linktag must contain the ID-  
number of the linked record.  
If ASP is given (optional), the functionality of  
ACC\_LINK will be activated.  
**IDX\_LINK,ASP** Link to 'systematics browse list' in Web-OPAC.  
**GLOBAL\_ITEM** (Multiple-) link to item records  
**SUB\_LIB\_ITEM** (Multiple-) link to item records belonging to the  
sub-library given with COMPARE=.  
**EXTERNAL\_LINK,TYP** Preparation for external processing:  
HTM Preparation as URL (Hyper-link in Web-  
browser).  
IMG Preparation as Image-URL (embedded  
picture in browser display).  
ANY Alternative format for an URL.  
**OPAC\_LINK,TAG** Preparation as URL in Web-OPAC, whereas the  
URL itself comes from TAG.  
**MIRROR** Mirroring of the field content (to the hidden trailer)  
LENG= maximum or fixed length (s. FILL, TYPE)  
LIMIT=  
for TAG=RECLIST: upper limit for preparation of linked records to be  
displayed  
else: minimum number of records  
LINK= number of link definition for preparation of linked records:  
**[file number:]link number**  
LOG= logical combination to the current preparation (default: O)  
NAME= external field name (can be swapped out)  
NOEX= Check for existence of field:

YES editing entry if field doesn't exist, too  
 NO don't edit entry if field doesn't exist  
 (default: NO)

**SIGN=** display sign of numerical values:  
 YES display also positive sign  
 NO display no sign  
 CD display negative sign as **D**, positive as **C**  
 EXCH exchange sign  
 (default: display negative sign only)

**SORT=** sort order for repeatable fields

**SUPPRESS=** elimination of special characters:  
 NSO character which indicates no sorting  
 NST character which cancels stop words  
 DIA diakritika which can't be displayed  
 PRO special characters which can't be displayed  
 (More entries than one: SUPPRESS=NSO:NST:DIA:PRO)

**START=** start position in field entry (standard: 1 (= 1. digit))

**SUB=** address for editing subfields (s. FORM=)

**TAB=** address for short/detailed table

**TEXT=**  
 for  
**TAG=000:** text instead of field entry (can be swapped out)  
 SKIP for line feed

for valid list of subfields to be edited (alternative definition for MARC-tags  
 field: as substitute for sub-format with FORM=)

**TYPE=** editing type (s. FILL, LENG):  
 VAR variable length  
 FXL fixed length, left-justified  
 FXR fixed length, right-justified  
 (standard: VAR)

**XTAG=** name for special tagging for XML-formatted output

**TABNO=** reference to customer defined table stored in the database.  
 [filenumber:tablename]

## Examples:

```

TAG=SYSID,BEFORE=' [IDN=',AFTER=' ] '
TAG=SYSDATE,DFORM=T,XTAG='subj_date'
TAG=000,NAME=011HDR,FORM=kopf
TAG=000,LINK=11:18,LOG=U
TAG=000,NAME=056IVP,LINK=60:2,FORM=ivptotal,DELIM=' / '
TAG=100/b,NAME=011100b,LINK=4,INFO='ALI_LINK,AUT,100$L',SUB=person
TAG=QUE.1,NAME=011590,LINK=3,INFO='ALI_LINK,TIP,QUE.1$L',SUB=590.1
TAG=GT0,NAME=011GT0,LINK=1,FORM=GT0,INFO='ALI_LINK,TIP,GT0$L'
TAG=418,NAME=011418,SUB=sub418,INFO='ACC_LINK,VLG'
TAG=000,NAME=011Band,LINK=1,INFO='SET_LINK,BDE'
TAG=200$L,INDIC=' ',CMPIND=EQ,LINK=6,AFTER=': ',LOG=X
TAG=655$u,NAME=011655$u,INFO='EXTERNAL_LINK,HTM',XTAG='httpref'
TAG=COV,NAME=011COV,INFO='EXTERNAL_LINK,IMG',XTAG='httpimage'
  
```

```
TAG=800,NAME=012800,TEXT='abcdefghijklmnopqrstuv4',INFO='ACC_LINK,SER'
```

---

## Definition indicator table

[presently not used]

### Syntax:

```
INDIC=indicator,TEXT=text
```

### Explanation:

INDIC= indicator

TEXT= text for indicator (can be swapped out)

---

## Definition link aspect

The link aspect definitions are included in file **marclink.ext**.

### Syntax:

```
LINK=name,DEFIN=link,DIRECT=F | B,FORM=label,LONG=name
```

### Explanation:

LINK= short name of link aspect

DEFIN= internal number of link definition:

**[file number:]link number**

DIRECT= direction of link:

F forwards

B backwards

(default: F)

FORM= address for record editing for full format

FACET= address for record editing for OPAC facets

LONG= text for detailed name of link aspect (can be swapped out)

## Examples:

```
LINK=TIT,DEFIN=1  
LINK=SWT,DEFIN=9,FORM=schlagw  
LINK=UEB,LONG=UEB015,DEFIN=2  
LINK=MEX,DEFIN=20:1,DIRECT=B
```

---

## Definition Script

### Syntax:

```
SCRIPT=name,PATH=path
```

### Explanation:

SCRIPT= script name

PATH= external pathname of script file

### Example:

```
(scripttab)  
SCRIPT=CHECK,PATH='../etc/marcscript.txt'
```

---

## Definition localization

This definition serves for editing records as a search query for external data sources.

### Syntax:

```
LOCATE=name,INDEX=name,SOURCE=label
```

### Explanation:

LOCATE= name of destination file  
INDEX= name of destination aspect  
SOURCE= address of field table

### Examples:

```
(titsfx)
LOCATE=SFx, INDEX=title, SOURCE=sfxtit
LOCATE=SFx, INDEX=isbn, SOURCE=sfxisbn
*
(sfxtit)
TAG=245$a
*
(sfxisbn)
TAG=020$a
```

---

## Definition name

### Syntax:

```
NAMI=name, NAME=name
```

### Explanation:

NAMI= internal name (used by Alephino programs)  
NAME= definition name

### Example:

```
(names)
NAMI=BIBLS, NAME=TIT
NAMI=ITEMS, NAME=MEX
```

---

## Definition Layout

### Syntax:

```
LAYOUT=number, ALPHA=alpha, ATTRIB=attribute, COLOR=color, FONT=font,  
OFFSET=number; number
```

### Explanation:

LAYOUT= number of the layout definition  
ALPHA= character set (one-digit, standard = L(atin))  
ATTRIB=BMP Display bitmap icon in GUI  
COLOR= color code C## (s. GUI-configuration)  
FONT= font code ## (s. GUI-configuration)  
OFFSET= line offset [presently no used]

### Example:

```
LAYOUT=1, FONT=01, COLOR=C01 * 14 pt, black  
LAYOUT=7, FONT=02, COLOR=C02 * 12 pt, red  
LAYOUT=10, FONT=01, COLOR=C01, ATTRIB=BMP
```

---

## Definition Statistics counter

This definitions are included in file **marcstat.ext**.

### Syntax:

```
COUNT=number, ACTION=action, FILE=number, INCREM=incr, KEY=label,  
NAME=name, PERIOD=period, SOURCE=number, SUBLIB=YES | NO, TAG=label
```

### Explanation:

COUNT= no. of counter definition  
ACTION= action to be counted (in combination with SOURCE=)  
FILE= no. of the counter file  
INCREM= count operation; default: +=1  
KEY= key of counter  
NAME= name of counter (optional)  
PERIOD= Time period:  
    YEAR cumulate p.a.  
    MONTH cumulate p.a./month  
    DAY cumulate p.a./month/day  
    HOUR cumulate p.a./month/day/hour

SOURCE= number of file to count (in combination with ACTION=)  
SUBLIB= YES: cumulate per sub-library  
TAG= address for editing counter criterion

**Examples:**

```
COUNT=1, FILE=46, SOURCE=20, ACTION=NEW, KEY=MNS, TAG=cntmexsta, SUBLIB=YES,  
PERIOD=YEAR  
COUNT=16, FILE=46, SOURCE=26, ACTION=LOAN, KEY=VLS, TAG=cntvbusta, SUBLIB=YES,  
PERIOD=MONTH
```