AFO 134 – Conversions

Appendices

This document contains (technical) background information to help you use AFO 134 more efficiently.

Appendix A. Conversion actions

Data can be manipulated using conversion actions. In this chapter follows a description of every action, together with one or more examples about how to use it.

Note

The behaviour of the conversion actions depends upon the following settings:

- Apply to output data
- Modify input data

1. Apply to output data = 0 / Modify input data = 0

   The action is performed upon the current input data and the result is added to the current output data.

2. Apply to output data = 0 / Modify input data = 1

   The action is performed upon the current input data, which are replaced with the resulting data.

3. Apply to output data = 1 / Modify input data = 0

   The action is performed upon the current output data, which are replaced with the resulting data.

4. Apply to output data = 1 / Modify input data = 1

   The action is performed upon the current output data, and the result replaces the current input data.

An example, where a name is converted to mixed case:

Input buffer Output buffer

JOHNSON	{empty}	
JOHNSON	J	Action: extract position 1
johnson	J	Action: to lowercase, **apply to input data**
johnson	Johnson	Action: extract position 2-999

An example, where the leading zero's are removed from a number of 5 positions (but always leaving at least 1 digit):

Input buffer	Output buffer	
00050	{empty}	
00050	0005	Action: extract position 1-4
00050	5	Action: left trim 0-characters, **apply to output data**
00050	50	Action: extract position 5

(note that this yields "0" as result if the original number is "00000").

A.1 Arithmetic functions

The following arithmetic functions can be used:

Add a value to the current input data value

Function to use

"Add"

Usage

The current input data should be empty or contain a numeric value.

Example

Current input data : 10

Value to add : 5

Result : 15

Multiply the current input data value with some other value

Function to use

"Multiply"

Usage

The current input data should contain a numeric value.

Example

Current input data : 10

Value to multiply with : 5

Result : 50

Take percentage of the current input data value

Function to use

"Percentage"

Parameters to specify

Percentage number

Usage

The current input data should be empty or contain a numeric value.

Example

Current input data : 50

Percentage : 10

Result : 5

A.2 String functions

The following string functions can be used:

Add a special character to the current data

Function to use

"Add special character"

Parameters to specify

Decimal code of the character to add

Position before which the character should be added [*optional, default=end-of-data*]

Usage

Use this for characters that are not printable, for instance characters 136 and 137 that are used as non-filing markers in the UNIMARC format.

Example

Current data : The

Character to add : 136

Result : □The

Add a string to the current data

Function to use

"Add string"

Parameters to specify

String to add

Position before which the character should be added [*optional, default=end-of-data*]

Example

Current data : The connection

String to add : French

Position parameter : 5

Result : The French connection

Append a string to the current data

Function to use

"Append string"

Parameters to specify

String to append

Usage

Note that this is the equivalent of the "Add string" without position parameter.

Example

Current data : The French

String to append : connection

Result : The French connection

Append punctuation at the end of the previous subfield

Function to use

"Append punctuation to previous subfield"

Parameters to specify

Punctuation to append

Usage
This action will only have effect when there is some preceding action for the current subfield that resulted in data in the output buffer.
Example

Current data of previous subfield: The main title

Punctuation to append : /

Result : The main title /

Copy current data

Function to use

"Take complete data"

Example

Current data : The French connection

Result : The French connection

Extract a string from the current data

Function to use

"Extract string"

Parameters to specify

Start position [*optional, default=1*]

End position [*optional, default=end-of-data*]

Example

Current data : The French connection

Start position : 5

End position : 10

Result : French

Extract numeric part from the current data

Function to use

"Extract numeric part"

Example

Current data : Tel. 0736243400 (Geac)

Result : 0736243400

Extract year from the current data

Function to use

"Extract year"

Parameters to specify

Start position [*optional*]

Usage

If no start position is given, the whole data string is searched for a year. Any number between 1000 and 3000 will be considered as a year.

Example

Current data : 1940

Start position : 1

Result : 1940

Current data : ca. 1940

Result : 1940

Current data : ca. 1940

Start position : 1

Result : [empty string]

Extract a delimited string from the current data

Function to use

"Extract delimited data"

Parameters to specify

Start delimiter [*optional*]

End delimiters, separated by backslashes [*optional*]

Usage

If no start delimiter is specified, the extracted data start at the beginning of the current data.

If no end delimiter is specified, the extracted data end at the end of the current data.

Example

Current data : The (French) connection

Start delimiter : (

End delimiter :)

Result : French

Current data : The French connection; an example / by D. Eveloper

End delimiter : ; \ /

Result : The French connection

(the start delimiter is omitted, so the extracted data start at the beginning of the current data; there are 2 end delimiters, "/" and ";", of which the ";" is found first)

Left justify the current data

Function to use

"Left justify"

Parameters to specify

Length

Fill character [*optional, default=blank*]

Example

Current data : 19

Length : 4

Fill character : ?

Result : 19??

Right justify the current data

Function to use

"Right justify"

Parameters to specify

Length

Fill character [*optional, default=blank*]

Example

Current data : 123

Length : 5

Fill character : 0

Result : 00123

Extract or remove name prefix from the current data

Function to use

"Name prefix"

Parameters to specify

Action: 1=get, 2=remove, 3=move to end

Language [*optional, default*=*]

Usage

Name prefixes are defined in ^SysPar("Filing.WordLists","Author",…).

Example

Current data : van der Dungen

Action : 1

Result : van der

Current data : van der Dungen

Action : 2

Result : Dungen

Current data : van der Dungen

Action : 3

Result : Dungen van der

Get length of non-filing part from the current data

Function to use

"Get length of non-filing data"

Parameters to specify

Start delimiter of non-filing part [*optional*]

End delimiter of non-filing part

Usage

If no start delimiter is given, the non-filing part starts at the beginning of the data.

Both start and end delimiter may be special characters; in that case they are specified with their decimal value.

Example

Current data : The @title

End delimiter : @

Result : 4

Current data : □The □Title

Start delimiter : 136

End delimiter : 137

Result : 4

Get non-filing part from the current data

Function to use

"Get non-filing data"

Parameters to specify

Start delimiter of non-filing part [*optional*]

End delimiter of non-filing part

Usage

If no start delimiter is given, the non-filing part starts at the beginning of the data.

Both start and end delimiter may be special characters; in that case they are specified with their decimal value.

Example

Current data : The @title

End delimiter : @

Result : The

Current data : □The □Title

Start delimiter : 136

End delimiter : 137

Result : The

Remove non-filing part from the current data

Function to use

"Remove non-filing data"

Parameters to specify

Start delimiter of non-filing part [*optional*]

End delimiter of non-filing part

Usage

If no start delimiter is given, the non-filing part starts at the beginning of the data.

Both start and end delimiter may be special characters; in that case they are specified with their decimal value.

Example

Current data : The @title

End delimiter : @

Result : title

Current data : □The □Title

Start delimiter : 136

End delimiter : 137

Result : title

Remove characters from the current data

Function to use

"Remove characters"

Parameters to specify

Character(s) to remove

Example

Current data : The French connection

Characters to remove : eio

Result : Th Frnch cnnctn

Remove a delimited string from the current data

Function to use

"Remove delimited data"

Parameters to specify

Start delimiter [*optional*]

End delimiters, separated by backslashes [*optional*]

Usage

If no start delimiter is specified, the removed data start at the beginning of the current data.

If no end delimiter is specified, the removed data end at the end of the current data.

Example

Current data : The (French) connection

Start delimiter : (

End delimiter :)

Result : The () connection

Current data : The French connection; an example / by D. Eveloper

End delimiter : ; \ /

Result : ; an example / by D. Eveloper

(the start delimiter is omitted, so the removed data start at the beginning of the current data; there are 2 end delimiters, "/" and ";", of which the ";" is found first)

Remove punctuation characters from the current data

Function to use

"Remove punctuation"

Parameters to specify

Punctuation characters [*optional, default=.,:;()=+!?*]

Example

Current data : The (French) connection. -

Result : The French connection

Replace substring in the current data

Function to use

"Replace"

Parameters to specify

String to replace

Replacement string

Example

Current data : The French connection

String to replace : French

Replacement string : English

Result : The English connection

Start at number in the current data

Function to use

"Start at number"

Example

Current data : Tel. 0736243400 (Geac)

Result : 0736243400 (Geac)

Convert the current data to lower case

Function to use

"Convert to lower case"

Example

Current data : The French connection

Result : the french connection

Convert the current data to upper case

Function to use

"Convert to upper case"

Example

Current data : The French connection

Result : THE FRENCH CONNECTION

Remove leading/trailing characters from the current data

Function to use

"Remove characters at beginning and end"

Parameters to specify

Characters to remove [*optional, default=blank*]

Example

Current data : …The (French) connection…

Characters to remove : .

Result : The French connection

Remove leading characters from the current data

Function to use

"Remove characters at beginning"

Parameters to specify

Characters to remove [*optional, default=blank*]

Example

Current data : The (French) connection

Result : The French connection

Remove trailing characters from the current data

Function to use

"Remove characters at end"

Parameters to specify

Characters to remove [*optional, default=blank*]

Example

Current data : The (French) connection…

Characters to remove : .

Result : The French connection

Remove repeated characters from the current data

Function to use

"Remove double characters"

Parameters to specify

Double characters to remove [*optional, default=blank*]

Example

Current data : The French connection

Result : The French connection

Remove leading/trailing punctuation from the current data

Function to use

"Remove punctuation at beginning and end"

Parameters to specify

Punctuation characters [*optional, default=.,:;()=+!?*]

Example

Current data : . - The (French) connection;

Result : The (French) connection

A.3 Date and time functions

The following date and time functions can be used:

Convert date

Function to use

"Convert date"

Parameters to specify

Date format, where YY or YYYY is year, MM is month and DD is day [*optional, default=YYYYMMDD*]

Usage

If the current data is an empty string, the current date will be taken; else the current data should contain a valid date.

Example

Current data :

Format : DD/MM/YY

Result : 28/12/01

Convert time

Function to use

"Convert time"

Parameters to specify

Time format, where HH is hours, MM is minutes and SS is seconds [*optional, default=HHMMSS*]

Usage

If the current data is an empty string, the current time will be taken; else the current data should contain a valid time.

Example

Current data :

Format : HH:MM

Result : 13:12

A.4 Table functions

The following table functions can be used:

Convert current data using a conversion table

Function to use

"Use conversion table"

Parameters to specify

Table name

Default value if no match [*if default = equal sign, then default=input value*]

Example

Current data : UK

Table : Country

Result : gb

Current data : ZZ

Table : Country

Default : xx

Result : xx

Current data : ZZ

Table : Country

Default : =

Result : ZZ

Current data : ZZ

Table : Country

Result :

(in the second, third and fourth case the value was not found in the table)

A.5 Formatting functions

The following formatting functions can be used:

Convert currency data

Function to use

"Convert currency"

Parameters to specify

Format: A ? Z

- A is prefix [*optional*]
- ? is formatting, where every # is a digit, e.g.: ###.###,##
- Z is suffix [*optional*]

Example

Current data : 10000

Format : Hfl. ###.###.-

Result : Hfl. 10.000.-

Current data : 100,50

Format : ###.###,## Euro

Result : 100,50 Euro

Current data : 100

Format : ###.###,00

Result : 100,00

Insert hyphens into ISBN

Function to use

"Insert hyphens into ISBN"

Usage

The current data must contain a valid 10-character ISBN number.

Example

Current data : 2203143193

Result : 2-203-14319-3

Insert hyphens into ISSN

Function to use

"Insert hyphens into ISSN"

Usage

The current data must contain a valid 8-digit ISSN number.

Example

Current data : 13629387

Result : 1362-9387

A.6 Control functions

The following control functions can be used:

Follow link

Function to use

"Follow link"

Parameters to specify

Start delimiter for link-id [*optional,default=begin-of-data*]

End delimiter for link-id [*optional,default=end-of-data*]

Field/subfield in link record from which data should be retrieved – subfield may be omitted

Usage

This action is used to follows a link to the actual data, using the following steps:

- extraction of link from between specified delimiters
- retrieval of link record
- retrieval of data in specified field/subfield

Repeat this until no link could be extracted.

Example

Current data : #12345678#Titel link

Start delimiter : #

End delimiter : #

Field/subfield : 200/$a

Result : The contents of field 200, subfield $a of the record with id "12345678"

Get data from link record

Function to use

"Get from link record"

Parameters to specify

Field

Subfield [*optional*]

Usage

The field (and subfield) are retrieved from the record that was obtained by the last "Follow link" action.

Example

Current data :

Field : 260

Subfield : $c

Result : The contents of field 260, subfield $c of the link record

Get data from linked field

Function to use

"Get data from linked field"

Parameters to specify

Link field

Link subfield (may be empty)

Data subfield

Usage

The field is retrieved from the current record.

If link subfield is not empty, the current contents of the input buffer are compared with the contents of this subfield and if they are equal the field is retrieved. In the conversion rule the source subfield must be the one that contains the link id.

If link subfield is empty, the link field with the same occurrence as the current field occurrence is retrieved.

Examples

Current data : AX-123

Link field : 930

Link subfield : $5

Data subfield : $a

Result : The contents of field 930$a if 930$5 is equal to AX-123

Current data :

Link field : 930

Link subfield :

Data subfield : $a

Result : The contents of field 930$a if it exists with the current field occurrence

Repeat last ? actions

Function to use

"Repeat last … actions"

Parameters to specify

Number of actions to repeat

Usage

This is a special function, that indicates a repetition of the last # actions. It can **only** be used with *GetDelimitedData* actions. The group of actions will be repeated until there are no more group data results.

Groups cannot be repeated more than 10 times.

Example

Suppose that the last 2 actions were:

GetDelimitedData, end delimiter / \;

GetDelimitedData, start delimiter / , end delimiter ; \ :

Both add a prefix "+" to the data.

Current data : abc / def : xxx; ghi : xxx; jkl / mno

Number of actions : 2

Result : abc+def+ghi+jkl+mno

Set a conversion variable

Function to use

"Set conversion variable"

Parameters to specify

Name of the variable

Value of the variable [*optional,default=current data*]

Example

Current data : F

Name of the variable : RecordType

Result : The RecordType variable is set to "F"

If the current data are used, beware that they will be copied to the output buffer as well. So use the "Modify input buffer" flag if this is not wanted.

Get a conversion variable

Function to use

"Get conversion variable"

Parameters to specify

Name of the variable

Example

Current data :

Name of the variable : RecordType

Result : Contents of the variable RecordType

Check a conversion variable

Function to use

"Check conversion variable"

Parameters to specify

Name of the variable

Values to check for (multiple values are separated by backslashes)

Operator (= for equal, # for different)

Usage

This action can only be used for an *Action condition*.

Example

Current data :

Name of the variable : RecordType

Values to check for : B\F\G

Operator : =

Result : The condition is true, if the RecordType is "B", "F" or "G"

A.7 External functions

The following external functions can be used:

Call an external routine

Function to use

"Call external routine"

Parameters to specify

Name of the routine (e.g. DoMyStuff^MyRoutines) – note that the ^ character will show as %5E

Parameter 1 [*optional*]

Parameter 2 [*optional*]

Usage

This action can be used when the standard actions are not sufficient for the data transformation (for instance, if information from the database is needed). The external routine should be of the form:

RoutineName*(InputData,InputParameters,ReturnParameter*)**

InputData = the data from the input buffer

InputParameters = parameter 1 in subscript 1, parameter 2 in subscript 2

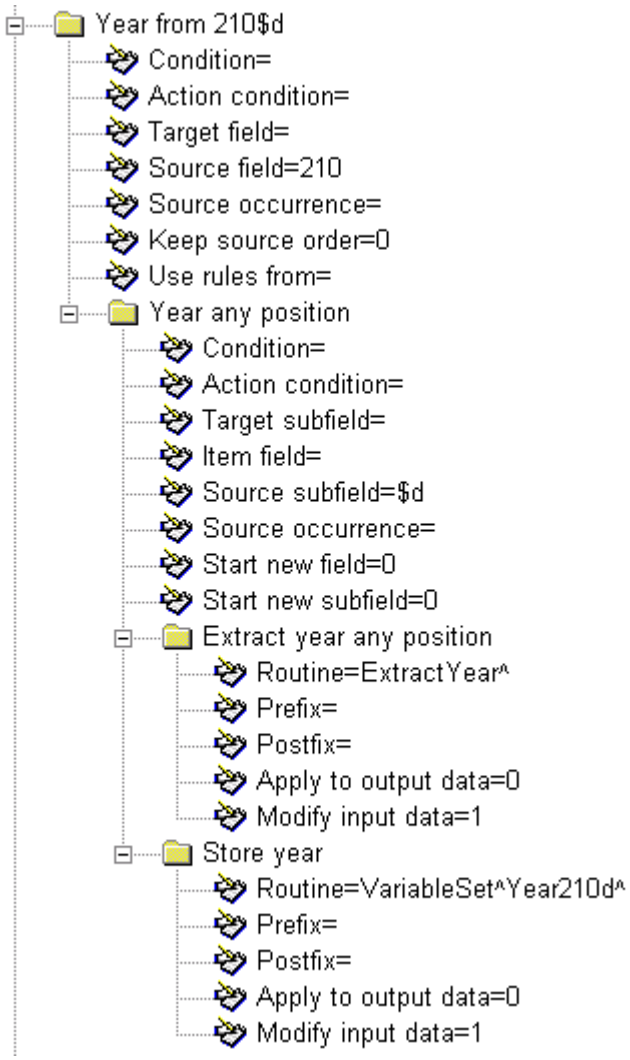
ReturnParameter = this is where the result of the transformation must be stored

Appendix B. Using the conversion actions

In the next chapter an explanation of how to use the various conversion actions is given.

B.1 Working with conversion variables

The following screen image shows how a conversion variable can be set:



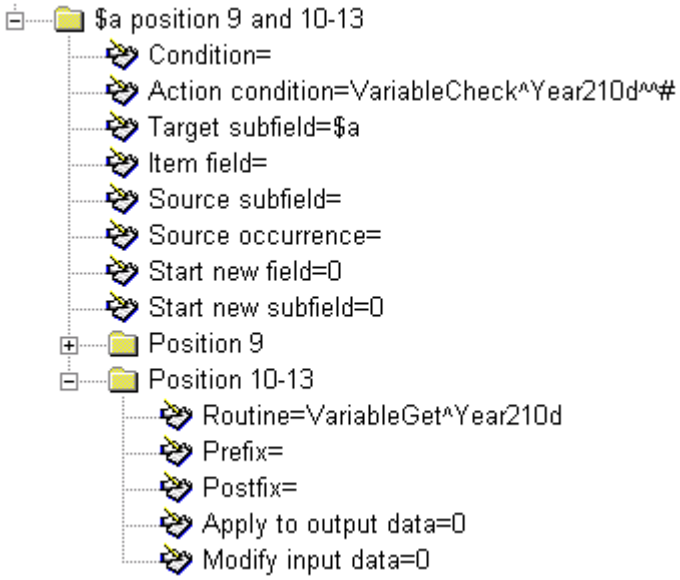
The conversion variable Year210d is set to the year as found in field 210 $d. Note the use of the "Modify input data" flag in the extraction of the year and in the assignment of the variable.

Later on this variable can be retrieved with the "Get conversion variable" action (see screen image on next page).

Note

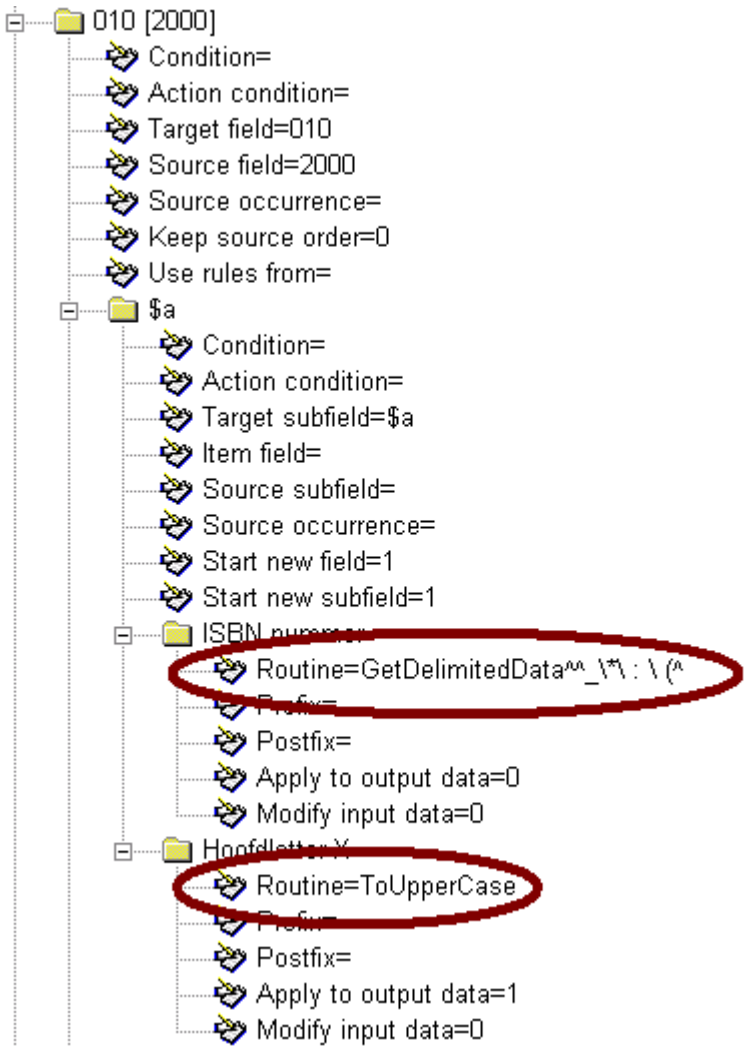
The strings "ExtractYear^" and "VariableSet^Year210d^" are obtained through a data entry form.

Usage of "Check conversion variable" and "Get conversion variable"; note that first is checked if the variable is not empty with the "Check conversion variable" action:



B.2 Extracting and manipulating data

In the following example an ISBN is extracted from the source data. The "Convert to upper case" action is used to make sure that an "X" at the end is in upper case - note that this action modifies the current output data.



Appendix C. Special fields

In this chapter an explanation is given of various special fields.

1. ###

There is a special field "###" in records that have been exported from Vubis Smart that contains control data of the record. Its structure is:

Record type~Creation date/time-Mutation date/time

Record type is a number.

Creation and mutation date/time are of the form YYYYMMDDHHMMSS.

2. 000

The "000" field can be used to indicate the record-type of the record as stored in ^BB.

It may also have the following special values:

- **D**: to mark the record for deletion
- **H**: only load holdings – do not update bibliographical data
- **R**: to indicate that this is a see-reference record

The "000" field has no subfields.

3. DBS

When authorities are loaded, the destination database can be different between records. Therefore it is possible to define the database-id in a special field if it is different from the one defined in the import profile. This special field has the fieldcode "DBS", and has no subfields. Its only contents are the database-id.

4. LNK

Sometimes bibliographic records that are loaded from an external source into the local database, need to be linked to other bibliographic records in the database. Basically there are two cases:

- linking to records that are loaded together with the record to be linked, and contain a pointer to their source record-id
- linking to records that already exist in the local database

In the conversion rules the target field "LNK" will have a special meaning: it is a link field. This field can contain the following subfields:

$a link type (as defined in ^SysDD)

$b link record-id (for linking to other record in same run)

$c matching profile (for linking to database record)

$d matching key (for linking to database record)

$e volume number

$x data to be copied to the record with which the link is established

$y data to be copied from the record with which the link is established

(Use $b only if you are sure that the record is loaded in the same run)

- The format of the subfield $x/$y is as follows:

  - destination fieldcode/subfield=source fieldcode/subfield, with multiple elements separated by a colon, e.g.:

  - 910/$a=900/$t:910/$b=900/$v:910/$c=%CDROM:801/$b=801/$b

This means:

- 900/$t from record A is copied to 910$a from record B

- 900/$v from record A is copied to 910$b from record B

- 900/$c from record B gets as contents CDROM

- 801/$b from record A is copied to 801/$b from record B

record A is for $x: the record that is currently loaded

record B is for $x: the record with which the link is established

record A is for $y: the record with which the link is established

record B is for $y: the record that is currently loaded

The source fieldcode-slash-source subfield code may be replaced by {%string} for literal strings, like in the example for $c

Note

If a matching profile is specified, also the matching key must be provided by the conversion process

When the record loader encounters such a "LNK" target field, it will try to establish a link according to the subfield contents.

During the link process no attempts will be made to produce "links based upon links", so only the specified link will be established (if possible).

Any links that could not be established will be reported.

Examples:

LNK $a 41

$b 7378290X

The link (with link type 41) is established based upon the source record number of the link record. So this record be loaded in the same run.

LNK $a 12

$c ISBN

$d 9017460023

$e 6

The link (with link type 12) is established based upon the lookup by ISBN number of the link record.

The LNK subfield

The special "LNK" subfield can now be used in conversions to establish a link with an existing authority in the database.

The structure of the "LNK" subfield is "*key / index*", where index is the authority index in which must be searched with the specified key in order to retrieve the authority to which must be linked.

For instance, "03577899/Index035" will trigger a search in index "Index035" with key "03577899".

If any other subfields must be created for the field that contains the "LNK" subfield, the "LNK" subfield must be the **first** subfield.

5. IDN

The IDN field can be used for matching on database record id. A check will be done if the number exists in the database. If it does, an update of that record is done; if the record does not exist the incoming record is rejected.

6. BBD

The BBD field can be used to add location and material type information to ^BB for use with partial indexes. The BBD field can have the following subfields:

- LOC
    - Location(s)
- MAT
    - Material type(s)

The data are merged into ^BB(Database,Record,105,*id*), where *id* is "Location" or "MaterialType".

7. HLD

The HLD field is used for holdings data. Every so called "item field" of this field corresponds with a holding element and must have one of the following predefined codes:

Code	Description	Format
BARCODE	barcode	
INSTITUTION	institution	
LOCATION	location	
SUBLOCATION	sublocation	
CALLPREFIX	call number prefix	
SHELFMARK	shelfmark	
CALLCUTTER	call number cutter	
CONTROLNO	shelving control number	
CALLSUFFIX	call number suffix	
SHELFMARKALGO	shelving algorithm	
MATERIALTYPE	material type	
MGRINSTITUTION	manager institution	
MGRLOCATION	manager location	
IMPRESSION	impression, e.g. "2nd ed."	
ITEMDATA	item data, e.g. "signed by author"	

VOLUME	volume	
YEAR	year (periodicals and multi-volume)	yyyy
TOTALLOANS	total number of loans ever	
CURYEARLOANS	total number of loans in current year	
FIRSTLOANYEAR	year of first loan	
FIRSTYEARLOANS	number of loans in first year	
YEARLOANS	total number of loans in year "yyyy"	loans/yyyy,loans/yyyy,…
LASTLOANDATE	date of last loan	yyyymmdd
LASTBORROWER	borrower code for last loan	
LASTRETURNDATE	date of last return	yyyymmdd
LASTRETURNTIME	time of last return	hhmmss
LASTRETURNLOC	institution/location of last return	institution/location
TOTALHOLDS	total number of holds ever	
ATTACHMENT	attached material (e.g. cd's, maps)	
SIGNATURE	user must sign for loan	0 (no) or 1 (yes)
ANNOTATION	annotation	
ACTUALBARCODE	actual barcode	

ENTRYDATE	data entry date	yyyymmdd
STATISTICALCODE	statistical code	
PRODCATEGORY	product category	xxx{,yyy,…}
REATTACH	flag for re-attach to other record	must be set to 1
MISC-PORDERNO	purchase order number	
MISC-ACCESSNO	accession number	
MISC-VIRTCOLL	virtual collection	
MISC-COST	cost	amount<space>currency code
MISC-PHYSCOND	physical condition	
MISC-CIRCSTAT	circulation status	
MISC-OPTSTACAT1	optional statistical category 1	
MISC-OPTSTACAT2	optional statistical category 2	
MISC-OPTSTACAT3	optional statistical category 3	
MISC-OPTSTACAT4	optional statistical category 4	
MISC-OPTSTACAT5	optional statistical category 5	
LOCCPY-COPYNO	copy number	
LOCCPY-SHSCHEME	shelving scheme	blank,0,1,2,3,4,5,6,7,8

LOCCPY-SHORDER	shelving order	blank,0,1,2
LOCCPY-SHTITLE	shelving title	
LOCCPY-COPAFC	copyright article fee code	
LOCCPY-PHYSFMT	physical format	
LOCCPY-RTGENPOL	retention, general policy	0,1,2,3,4,5,6,7,8
LOCCPY-RTSPCPOL	retention, specific policy	l,p,blank
LOCCPY-RTUNITS	number of units	blank,0,1,2,3,4,5,6,7,8,9
LOCCPY-POLUNIT	retention policy unit	m,w,y,e,i,s,blank
NOTE-PUB	public note	
NOTE-NONPUB	non-public note	
NOTE-ATTACH	attachment note	
NOTE-REPROD	reproduction note	
NOTE-OWNER	ownership note	
NOTE-VERS	copy and version identification note	
NOTE-ACT	action note	
NOTE-TERMS	terms for use and reproduction	
NOTE-CIN	return note	

INVMES-FREQ	frequency	
INVMES-FREQTYPE	frequency type	0=day,1=week,3=month,4=year
INVMES-DISPDATE	date to display the message	yyyymmdd
INVMES-MSG	inventory message	
OPT-FIELD1-20	optional field 1-20	
EH-ALL	complete electronical holding	for export only
EH-1STACCMTHD	first access method	#,0,1,2,3,4,7
EH-RELSHIP	relationship	
EH-HOSTNAME	host name	
EH-ACCESSNO	access number	
EH-INFCOMPR	information compression	
EH-PATH	path	
EH-ELECNAME	electronical name	
EH-PROCREQ	processor of the request	
EH-INSTRUCT	instruction	
EH-BPS	bits per second	
EH-PASSWORD	password	

EH-LOGON	logon	
EH-ASSIST	contact for access assistance	
EH-LOCNAME	name of location of host	
EH-OS	operating system	
EH-PORT	port	
EH-ELECFMT	electronic format	
EH-SETTINGS	settings	
EH-FILESIZE	file size	
EH-TERMEMUL	terminal emulation	
EH-URI	uniform resource indentifier	
EH-HOURS	hours access method available	
EH-RECNO	record control number	
EH-NONPUBNOTE	nonpublic note	
EH-LINKTEXT	link text	
EH-PUBNOTE	public note	
EH-ACCMTHD	access method	
EH-MATSPEC	materials specified	

EH-LINKAGE	linkage	
EH-LINKNO	field link and sequence number	
CSTATUS	Loan status code (as defined in AFO 481)	

The updates that may be performed with the items can be specified for **BARCODE** with an "Action" definition with actiontype "Allowed update actions". Choices are

- Add
- Modify
- Add or Modify (default)
- Delete

Note

The electronical holdings are exported to the intermediate format in a slightly different way; instead of having separate subfields for every electronical holdings element, all elements are grouped per electronical holding as follows:

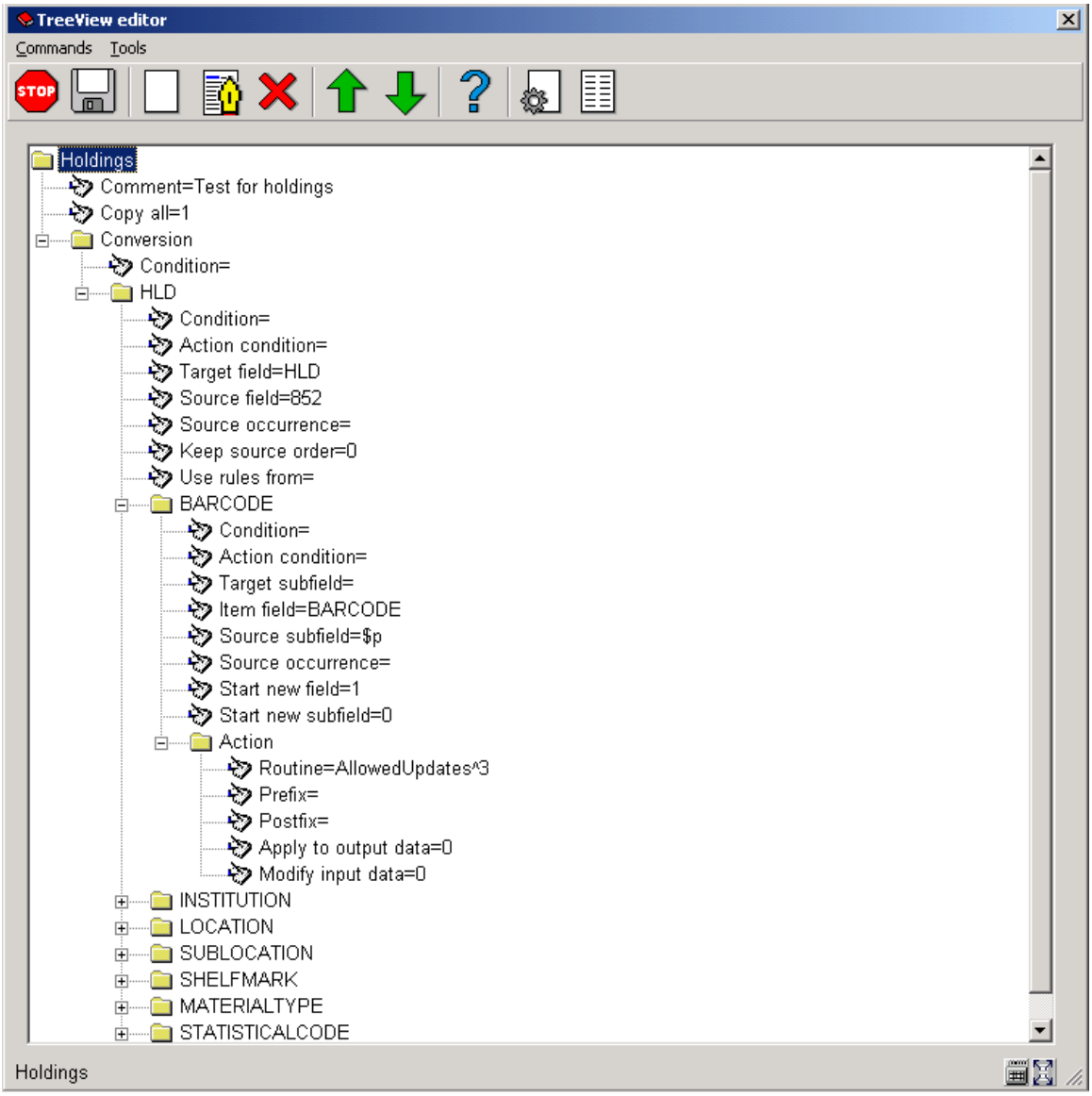
…,"HLD.xxx","EH-ALL")= *element*:*data*### *element*:*data*### *element*:*data*###…

for instance

EH-LOGON:user###EH-PASSWORD:password###EH-URI:uri###

The individual elements can be extracted from the electronical holdings buffer with the "Extract delimited data" conversion action, with as delimiters the element-id and the string ###.

Example screen of a simple holdings conversion setup:



Note that in this example there is no conversion of bibliographical data, and the "Copy all" flag has been set for the bibliographical fields.

Make sure to set the "Start new field" flag to 1 for the first subfield.

Appendix D. Editing with the TreeView control

The TreeView control is used for editing data that can be organized in a tree structure. Editing of data is done similar to editing a file name in Windows Explorer:

- Click once upon the data to enter editing mode

- Enter the (new) data

- Type <ENTER> to store the new data, or <ESC> to restore the original data

The following commands are available:

New subgroup: Create a new subgroup. This command is only valid when a group header is highlighted. A dropdown list with the possible subgroups will be displayed, or else, if there is only one subgroup, that one will be created.

Edit: Edit the current data. This has the same effect as clicking the data once.

Edit detail: Edit data that have been retrieved through a form or dropdown list. This is a fast way to change some detail in the data, and should be used with care as it may lead to unexpected results if the data formatting is not respected.

Delete: Delete the current group. This command is only valid when the group header is highlighted.

Move up: Moves the current field or group up. Note that for fields this makes no difference for the storage of the field data. Groups are however stored in the order that they are displayed.

Move down: Moves the current field or group up. Note that for fields this makes no difference for the storage of the field data. Groups are however stored in the order that they are displayed.

Save: Saves the current conversion settings.

Close: Closes the TreeView.

- **Document control - Change History**

Version	Date	Change description	Author
1.0	**July 2007**	**Creation** **Delivered as part of build 17 set**	
2.0	**June 2010**	**Possibility to load loan status code as defined in AFO 481** **part of 2.0.06 updates**	