<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>13</td>
</tr>
<tr>
<td>5426_to_unicode2</td>
<td>13</td>
</tr>
<tr>
<td>853_chrono</td>
<td>14</td>
</tr>
<tr>
<td>853_numbering</td>
<td>14</td>
</tr>
<tr>
<td>aleph_start</td>
<td>15</td>
</tr>
<tr>
<td>aleph_startup</td>
<td>19</td>
</tr>
<tr>
<td>allowed_languages</td>
<td>19</td>
</tr>
<tr>
<td>bib_format</td>
<td>20</td>
</tr>
<tr>
<td>cash_09_translate</td>
<td>21</td>
</tr>
<tr>
<td>check_circ</td>
<td>21</td>
</tr>
<tr>
<td>check_circ_override</td>
<td>22</td>
</tr>
<tr>
<td>check_doc</td>
<td>22</td>
</tr>
<tr>
<td>check_doc (error messages)</td>
<td>24</td>
</tr>
<tr>
<td>check_doc.&lt;lng&gt;</td>
<td>24</td>
</tr>
<tr>
<td>check_doc_doc</td>
<td>25</td>
</tr>
<tr>
<td>check_doc_field_006</td>
<td>27</td>
</tr>
<tr>
<td>check_doc_field_007</td>
<td>28</td>
</tr>
<tr>
<td>check_doc_field_008</td>
<td>28</td>
</tr>
<tr>
<td>check_doc_field_ldr</td>
<td>29</td>
</tr>
<tr>
<td>check_doc_line</td>
<td>30</td>
</tr>
</tbody>
</table>
tab_location_name.<lng> ................................................................. 170

tab_mapping .................................................................................. 170

tab_mapping_mng500 ...................................................................... 171

tab_match ....................................................................................... 172

tab_match_acc ............................................................................... 173

tab_match_script ......................................................................... 174

tab_merge ...................................................................................... 175

tab_merge_adv_overlay ................................................................. 176

tab_merge_overlay ....................................................................... 178

tab_mime_type ............................................................................... 180

tab_month ...................................................................................... 180

tab_move_record ........................................................................... 181

tab_oclc ........................................................................................ 183

tab_own ......................................................................................... 184

tab_photo_request ....................................................................... 185

tab_pinyin ..................................................................................... 186

tab_preferred ............................................................................... 187

tab_rlin ......................................................................................... 188

tab_service .................................................................................. 189

tab_sfx ......................................................................................... 190

tab_sip2_alert ............................................................................... 192

tab_sip2_conf ............................................................................... 193
| tab24                           | ............................................................ | 262 |
| tab25.<lng>                    | ............................................................ | 262 |
| tab27                           | ............................................................ | 263 |
| tab30                           | ............................................................ | 264 |
| tab31                           | ............................................................ | 265 |
| tab32                           | ............................................................ | 267 |
| tab33.<lng>                    | ............................................................ | 268 |
| tab34                           | ............................................................ | 269 |
| tab35                           | ............................................................ | 270 |
| tab36                           | ............................................................ | 271 |
| tab37                           | ............................................................ | 272 |
| tab37_campus_filter            | ............................................................ | 274 |
| tab37_campus_preferred         | ............................................................ | 275 |
| tab38                           | ............................................................ | 276 |
| tab39                           | ............................................................ | 277 |
| tab40.<lng>                    | ............................................................ | 278 |
| tab41                           | ............................................................ | 279 |
| tab42                           | ............................................................ | 280 |
| tab43                           | ............................................................ | 282 |
| tab44                           | ............................................................ | 284 |
| tab45.<lng>                    | ............................................................ | 285 |
| tab46.<lng>                    | ............................................................ | 286 |
General
This documentation describes the tables for the management and configuration of the ALEPH libraries. The tables in this guide are listed alphabetically. The index at the end of the guide is an aid for accessing a table by its subject or topic.

ALEPH Library Structure
An ALEPH installation consists of several libraries. The central library is the ADMinistrative library, which manages all administrative aspects of the library including items control, acquisitions, serials, ILL, and circulation. The ADM library is linked to several types of records, each of which is held in a separate library or databases:

<table>
<thead>
<tr>
<th>Type of Record</th>
<th>Format of Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIBliographic records</td>
<td>Any MARC format</td>
</tr>
<tr>
<td>HOLdings records</td>
<td>MARC21 format</td>
</tr>
<tr>
<td>AUThority records</td>
<td>MARC21/UNIMARC formats</td>
</tr>
<tr>
<td>ILL records for the bibliographic details of the ILL request</td>
<td></td>
</tr>
</tbody>
</table>

Note: One installation can have more than one ADM library and/or BIB/HOL/AUT/ILL databases.

ALEPH has the following convention for naming libraries of one installation. A numeric suffix is added to the name of the library where:
For example for the MARC21 demonstration library:

<table>
<thead>
<tr>
<th>Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>USM01</td>
<td>BIBliographic records library</td>
</tr>
<tr>
<td>USM10</td>
<td>AUThority records library</td>
</tr>
<tr>
<td>USM20</td>
<td>ILL records library</td>
</tr>
<tr>
<td>USM30</td>
<td>Course Reading library</td>
</tr>
<tr>
<td>USM50</td>
<td>ADMinistrative library</td>
</tr>
<tr>
<td>USM60</td>
<td>HOLdings records library</td>
</tr>
</tbody>
</table>

5426_to_unicode2
Location of the table: alephe/unicode directory
Purpose of the table: Translates ISO 5426 to Unicode

This table translates ISO 5426 to Unicode. ISO5426 character set is the European standard for bibliographic data, and although it is different from ANSEL (NISO Z39.47) it serves the same function. It is used by French equivalent of "Books in Print".
853_chrono

Location of the table: alephe/tab directory
Purpose of the table: Table for names of months and seasons

This table is used for supplying chronology text for months and seasons for the description field of Serials items records, when 85X/85XX fields are used for building the description.

The months should be listed from January through December, and the seasons should be listed from Spring through Winter.

The language in the ADM 008 field is matched with the language code in this table.

Structure of table:
- Col. 1 language code – 3 characters, lower case.
- Col. 2 M – month
  - S – season
- Col. 3 Text

Example of the table:

```
1 2 3
eng M Jan.
eng M Feb.
eng M Mar.
...
eng S Spring
eng S Summer
eng S Autumn
eng S Winter
fre M janv.
fre M févr.
fre M mars
fre M avril
```

853_numbering

Location of the table: alephe/tab directory
Purpose of the table: Abbreviations to be used with numbering

This table defines the ordinal suffix to add to enumeration levels a-h. The ordinal suffix will be used when the 853 field has a + sign before the enumeration caption. In this case, the enumeration and its ordinal suffix are placed before the caption.

For example:
```
$$b+^\text{quarter}$$ will create
```
1st quarter
2nd quarter, etc.

Use a ? sign in the table to denote a wild card (e.g. ?1 for 21st, 31st, 41st, etc. Note that the program uses the first row whose mask matches the number -- for the above example "11" (i.e. 11th) should be listed before ?1.

Language is taken from pos. 35-37 in the 008 of the ADM record.

Structure of table:
Col. 1  language code – 3 characters, lower case.
Col. 2  number
Col. 3  ordinal suffix

Example of the table:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>!1</td>
<td></td>
<td>th</td>
</tr>
<tr>
<td>!1</td>
<td></td>
<td>th</td>
</tr>
<tr>
<td>!1</td>
<td></td>
<td>th</td>
</tr>
<tr>
<td>!1</td>
<td></td>
<td>th</td>
</tr>
<tr>
<td>eng</td>
<td>11</td>
<td>th</td>
</tr>
<tr>
<td>eng</td>
<td>12</td>
<td>th</td>
</tr>
<tr>
<td>eng</td>
<td>13</td>
<td>th</td>
</tr>
<tr>
<td>eng</td>
<td>?1</td>
<td>st</td>
</tr>
<tr>
<td>eng</td>
<td>?2</td>
<td>nd</td>
</tr>
<tr>
<td>eng</td>
<td>?3</td>
<td>rd</td>
</tr>
<tr>
<td>eng</td>
<td>?</td>
<td>th</td>
</tr>
<tr>
<td>fre</td>
<td>1</td>
<td>er</td>
</tr>
<tr>
<td>fre</td>
<td>?</td>
<td>e</td>
</tr>
</tbody>
</table>

**aleph_start**

Location of the table: alephe directory

Purpose of the table: Node environment definitions

**General note:**
Note that there are some definitions that offer more than one “setenv” option. The non-active option does not have to be commented out – the system will always use the last line defined for that option.

In order for changes to aleph_start to take effect, you must exit aleph, re-log-in, and then restart the daemons and servers. (For batch jobs submitted through Services this means that the batch_que must be restarted.) Running aleph_shutdown and aleph_startup restarts all daemons and servers.

Following is an explanation of some of the parameters in aleph_start. The parameters are listed alphabetically ignoring ‘setenv’:
setenv ACC_AUT_STARTUP_LIBS "usm01 usm50"
For use with AUTHORITY library/ies. It links the bibliographic database headings to
authority records. This line parallels activation of UTIL E/8 “Start update BIB ACC
from AUT”.

setenv ADJACENCY
• ADJACENCY_TYPE 0 - does not build adjacent words
• ADJACENCY_TYPE 2 - builds adjacent words

If set to 0, the variable setenv set_prox_limit should be defined in www_server.conf.

Note that setting "2" improves performance for adjacency searching, but
approximately 4x storage space per document is required.

setenv alephe_synch_root $alephe_synch_dev/alephe
Used for update synchronization of alephe tables and error files

setenv aleph_utf /tmp
Defines the path to the utf files.

setenv bor_id_concat_char
This parameter is linked to the WEB OPAC and patron retrieval by concatenation of
ID and verification.

It is possible to retrieve a patron by using the ID (as Z308-KEY-DATA)
and the password/verification (Z308-VERIFICATION).

The data is loaded to the system such that Z308-KEY-DATA comprises of
• a patron name (e.g. "DANNY"),
• a concatenation character (e.g. a hyphen "-")
• and the verification (e.g. "X"):
  o Z308-KEY-DATA: DANNY-X
  o Z308-VERIFICATION : X

The concatenation character is defined in this variable.

The purpose of this development is to make it possible to use a single ID (but with
different verifications) for different tasks in the WEB OPAC.

Column 8 in tab_bor_id.<lng> should be set to "N" for codes (Z308-KEY-TYPE)
used for this purpose, to prevent patrons from changing their verification.

Note that if the variable is not set, or it is set to spaces, then no concatenation of ID
and verification will be made in the retrieval process.

setenv correct_852_subfields "2hijklm"
The link between the item and the holdings record is created by entering the holdings
record system number in the item record. When the link is created the system will
override location (sub-library, collection and call number and call number type) information in the item using the location information from the holdings record.

The following information will be overridden:

- Item sub-library by $$b of 852
- Item collection by $$c of 852
- Item call number type by 1st indicator of 852
- Item call number by sub-fields from 852 as defined in the above variable.

Note too, that the location field in expand_doc_bib_psts and expand_doc_bib_locXX are created in accordance with the definitions in correct_852_sub-field.

```
setenv date_style_s 24
setenv date_style_f 16
setenv time_style_s 04
setenv time_style_f 04
```

For a full explanation of the date and time styles refer to pc_server_defaults.

```
setenv due_date_format 1
```

This setenv controls what displays when there is just a single "Due date" column. It works in combination with switch RECALL-METHOD in tab100.

- **1** - Display a single "effective due date", based on RECALL-METHOD in tab100.
- **2** - Display the recall-due-date, if there is one (even if it's later); otherwise, display the regular due date.
- **3** - Always display only the z36_due_date; (for recall methods 2 and 3 this would mean that the recall-due-date will never display).*
- **4** - Display both: the regular due date and (if there is one) the recall-due-date.

```
setenv fine_factor_fix 000
```

It is possible to define a multiplication factor for the fine rate set in tab16, column 11. This is useful for countries whose currencies are written in the thousands (e.g., 10,000 lira).

```
setenv fine_rounding Y:
```

Round fine up or down to get rid of the decimal points. Rounds up when the sum after the decimal point is more than 0.50 and rounds down when the sum is lower than 0.50.

```
setenv ill_total_limit_period 0814
```

Defines the option to calculate the total ILL limit of a patron according to a period of time. The variable has format MMDD. The calculation is done from the current date to this variable date (the assumption is that the variable date is earlier than the current date).

All ILL requests that are created between these dates are counted and compared to the total patron's limit (Z303-ILL-TOTAL-LIMIT).

In order to ignore this calculation, either delete the variable or enter 0000.

```
setenv ill_mode EXTERNAL
```
setenv ill_mode                INTERNAL
  • EXTERNAL - uses the Ex Libris Web based ISO-ILL product
  • INTERNAL - uses the Ex Libris GUI ALEPH ILL module.

If not defined the system will work as if INTERNAL.

setenv include_all_aut
Y= all accesses which connected to authority record will be displayed
in the browse list of WEB OPAC. Even if there are no records connected
to this access.
This applies only to codes using Z01 and not to those which are using Z0102.
Z0102 continues to use the XYZ mechanism.

setenv local_currency           USD
Definitions with regard to the library’s local currency. If defined here:
  • The currency table is never consulted. The ratio for the defined currency is
    always 1.00
  • This currency will be the default currency for allocation (if not currency
    filled in the order form.)
  • Will always be the currency for transfer transactions.
  • There is an error message when trying to add/replace/delete a ratio for the
    currency.

In order to change the local currency as defined in this parameter:
  • In the GUI-ADMIN module, define currency XXX, and set its ratio to
    1,000
  • Define XXX as the local currency in ./alephe/aleph_start:
    setenv local_currency   USD
  • Run the command: source alephe_root/aleph_start
  • Restart PC server (UTIL W/3/3)

setenv MESSAGE_STARTUP_LIBS  " $z105_library"
For messaging between libraries for mutual update (e.g. update of Authority record
triggers re-linking BIB heading). This line parallels activation of UTIL E/11 “Start
messaging update”.

setenv OCLC_SERVER_STARTUP   Y
This controls the startup of the OCLC server

setenv pw_library         USR00
setenv usr_library        USR00
setenv z105_library       USR00
Every installation has a default "library" USR00. The purpose of this library is as
follows:
  • for non-ILS applications (such as Union Catalog, DigiTool) that
    nevertheless require USERS (patrons and librarians), there is no need for a
    complete ADM library; there is only need for records related to patrons (id,
    address, profile, sdi) and passwords.
• Z105 messages on the same server are written in and handled by the USR00 library. ue_11 is activated only in this library, which sends all update triggers to the correct destination.
• for applications that have multiple ADM libraries -- passwords are always shared across libraries, and USR00 can be used for storing them (rather than choosing one of the ADM libraries).

```plaintext
setenv REQUEST_STARTUP_LIBS "usm50"
Links up to tab37 or tab38 for managing retrieval and printing of hold or photocopy requests. This line parallels activation of UTIL E/6 “Start request handling” for the library.

setenv RLIN_STARTUP_LIBS "usm01"
Defines the library that will start the RLIN loader.

setenv SIP2_STARTUP_LIBS "usm50"
Defines the library that will start the Self Service Server.

Setenv_user_list_aleph_local
This variable defines whether "ALEPH" in Z305 will be taken into account when showing the Local user list.

setenv WWW_HOST ram19
This definition is needed by the www_server in cases where the hostname is not the external name. If this variable is not defined then the hostname is used as the back link in all web pages.

setenv Z39_SERVER_STARTUP Y
setenv Z39_GATE_STARTUP Y
This controls the startup of the Z39 gate and server.

For more detail of other elements of the table refer to the System Management Guide.

**aleph_startup**

Location of the table: alephe directory

Purpose of the table: Startup definitions

For more detail please refer to the System Management Guide

**allowed_languages**

Location of the table: alephe/tab directory

Purpose of the table: Language code definitions
This table lists the language codes recognized by the system. If not listed here, the system will not recognize the language code and will return an error message (e.g. when entering data in the vendor language field in the online GUI).

Note that in addition to the definitions listed in the table, the code and text MUST be defined in pc_tab_expand as well (section LANG).

Structure of the table:

Cols. 1-20
Language codes

Example of the table:

!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!

ACE ACH ADA AFA AFH AFR AJM AKA AKK ALB ALE ALG AMH ANG APA ARA ARC
ARM ARN ARP ART ARW ASM ATH AWA AVE AYM AZE
BAD BAT BAK BAL BAM BAN BAQ BAS BAT BEJ BEL BEM BEN BER BHO BIK BIN
BLA BRA BRE
BUG BUL BUR

**bib_format**

Location of the table: tab directory of the ADM library

Purpose of the table: Formats for bibliographic display in templates

This table defines the creation of the bibliographic format for administrative printouts. The system turns to the edit_paragraph.<lng> table for the display of the bibliographic information in the administrative printouts.

Note that a default line should be defined:

```
default 300
```

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>template name</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>document format, as defined in the BIB library's edit_paragraph.&lt;lng&gt;.</td>
</tr>
</tbody>
</table>

Example of the table:

```
1  2
! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!
default 300
acq-additional-claim 301
acq-arrival-slip 301
acq-claim-m-report 300
acq-claim-o-report 300
```
cash_09_translate

Location of the table: tab directory of the ADM library

Purpose of the table: Sublibrary translation to Campuses

This is a table that translates sub-libraries to a department code, for report purposes. The report used is cash-09 report.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Sublibrary;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Campus Code</td>
</tr>
</tbody>
</table>

Example of the table:

```
! 1 2
!!!!--!!!!
UELEC 3020
```

check_circ

Location of the table: alephe/error_<lng> directory

Purpose of the table: Error messages linked to circ tables

Error messages linked to tab_check_circ, check_circ_override and tab_block_circ

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Error code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>ALPHA</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Error message</td>
</tr>
</tbody>
</table>

Example:

```
!!!!--!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
0098 L Successful renew (override)
0099 L Successful renew
```
check_circ_override

Location of the table: tab directory of the ADM library
Purpose of the table: Minimum user level for override

This table defines the minimum password level that is required in order to be able to override a circulation transaction that has been trapped by a circulation check procedure.
A minimum password level is defined for each check procedure. The check procedure is identified by using the number of the error message that displays when the check procedure traps the transaction.

Error codes are system defined, and can be found in alephe/error_<lng>/check_circ.
Structure of the table:
col. 1 Error code from alephe/error_<lng> check_circ.
col. 2 Minimum patron level for override;
col. 3 Description of the error (non functional)

Example of the table:

<table>
<thead>
<tr>
<th>Error code</th>
<th>Minimum patron level for override</th>
<th>Description of the error</th>
</tr>
</thead>
<tbody>
<tr>
<td>00101</td>
<td>00</td>
<td>check_circ_1_a: Delinquency - $1 $2</td>
</tr>
<tr>
<td>00102</td>
<td>00</td>
<td>check_circ_1_a: Delinquency - $1 $2</td>
</tr>
<tr>
<td>00103</td>
<td>00</td>
<td>check_circ_1_a: Delinquency - $1 $2</td>
</tr>
<tr>
<td>00111</td>
<td>00</td>
<td>check_circ_1_b: Local Delinquency - $1 $2</td>
</tr>
<tr>
<td>00112</td>
<td>00</td>
<td>check_circ_1_b: Local Delinquency - $1 $2</td>
</tr>
<tr>
<td>00113</td>
<td>00</td>
<td>check_circ_1_b: Local Delinquency - $1 $2</td>
</tr>
<tr>
<td>00115</td>
<td>00</td>
<td>check_circ_1_b: Status not valid for this sub-library</td>
</tr>
</tbody>
</table>

check_doc

Location of the table: tab directory of the library
Purpose of the table: Check routines

The check_doc table is used to define the check programs that are used in the system.
Up to 100 program names may be assigned to library's check_doc function.

**Column 1** contains the check type that defines when the check program is performed.

Following are the reserved check types:

- **CATALOG-INSERT**: performed when the cataloging record is saved, updated or when the Check Record option is selected from the Cataloging module.
- **CATALOG-DELETE**: performed when the Delete Record from Server option is selected from the Cataloging module.
- **BATCH-DELETE**: performed when the Delete Bibliographic Records (p-manage-33) batch process is run.
- **NAV-MAP-DELETE**: performed when the Total Delete option is selected from the Record Manager of the Cataloging module to delete a bibliographic record together with all associated records.
- **P-MANAGE-11**: performed when the Check Database Tables (p-manage-11) batch process is run.
- **Z39-INSERT**: performed when a record is inserted via Z39.50 ES Update.
- **Z39-REPLACE**: performed when a record is replaced via Z39.50 ES Update.
- **Z39-DELETE**: performed when a record is deleted via Z39.50 ES Update.

**Column 2** contains the check program(s) that should be performed for the specific check type defined in column 1.

**Column 3** - Program arguments. Certain check_doc programs require additional information such as table names. These additional parameters are defined in column 3. The documentation for each check_doc program indicates whether it takes parameters, and if so, how they should be formatted. Note that if a check_doc program does not use parameters, it will ignore the contents of column 3.

Structure of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 1</td>
<td>col. 2</td>
<td>col. 3</td>
</tr>
<tr>
<td>Check type</td>
<td>Check program</td>
<td>Program arguments</td>
</tr>
</tbody>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_new_acc</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_new_acc_aut</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_unique_index</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_tag_text</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_line</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_line_contents</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_lkr</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-INSERT</td>
<td>check_doc_doc</td>
<td>!</td>
</tr>
</tbody>
</table>
Note:
Some checks are not related to the document itself but to another document.
For example check_doc_unique_index. The message window has a button to be able to retrieve the related record.

This feature is implemented on the following routines:

- check_doc_unique_index
- check_doc_delete_lkr
- check_doc_locate
- check_doc_aut_duplicate (AUT library)

Note that as a result the logic of check_doc_locate has changed slightly, and will now show up to 5 related records.

**check_doc (error messages)**

*Location of the table: alephe/error_<lng>*

*Purpose of the table: Validation messages (system driven) for check_doc programs*

The error messages defined in this table are system driven and are between the range of 0001-4999.

*Structure of the table:*

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Error message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>ALPHA</td>
</tr>
<tr>
<td>Col. 3</td>
<td>text</td>
</tr>
</tbody>
</table>

*Example of the table:*

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0002</td>
<td>L Document has $3 item(s) attached to ADM record $1 in library $2</td>
</tr>
<tr>
<td>0003</td>
<td>L ADM record has $1 item(s) attached.</td>
</tr>
<tr>
<td>0004</td>
<td>L Document has $3 order(s) attached to ADM record $1 in library $2</td>
</tr>
<tr>
<td>0005</td>
<td>L ADM record has $1 order(s) attached.</td>
</tr>
<tr>
<td>0006</td>
<td>L Document has $1 subscription(s) attached to ADM record $3 in library $2</td>
</tr>
<tr>
<td>0007</td>
<td>L ADM record has $1 subscription(s) attached.</td>
</tr>
</tbody>
</table>

**check_doc.<lng>**

*Location of the table: tab directory of the library*

*Purpose of the table: Validation messages (table driven) for check_doc programs*

This table provides for validation messages for the check doc programs. The error messages defined in this table are table dependent and should be between the range of 5000-9999 only – e.g. checks in program check_doc_doc and check_doc_mandatory.
In the following example (in check_doc_doc) the message number defined for a particular check is 6003. This matches up with the message line 6003 in check_doc.lng:

check_doc_doc

| D | BK | 6003 | 2450# | Y | 1### | N |

check_doc.lng>

6003 L Use 245 1# when there is a main entry (1XX)

Structure of the table:
- Col. 1 Error message
- Col. 2 ALPHA
- Col. 3 text

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!--!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5001 L A record cannot have more than 1 main entry (1XX).
5002 L Required 245 field is either missing or duplicated.
5003 L Required 260 field is either missing or duplicated.
5007 L Required 310 field is either missing or duplicated.
5008 L Required 008 field is either missing or duplicated.
5120 L Required 256 field is either missing or duplicated.
5121 L Tag 538 is required for Computer Files.

check_doc_doc

Location of the table: tab directory of the library

Purpose of the table: Checks between tags

Subtable OC

Occurrences - giving up to 5 field codes, with implied OR.

Structure of the table:
- Col. 1 OC
- Col. 2 Record format as registered in the FMT field in the document record. XX should be entered if check valid for all formats.
- Col. 3 error number – between 5000 and 7000 - as listed in the check_doc.lng table (located in the tab directory of the library)
- Col. 4 Minimum number of occurrences.
  00 indicates that the field is not mandatory.
  Any number from 01 indicates that the field must be present.
- Col. 5 Maximum number of occurrences.
  99 indicates unlimited number of occurrences.
Cols. 6 -10
Fields for which number of occurrences is being defined. Up to 5 field codes with OR implied.

Examples:
• For BK format, a record cannot have more than one main entry:
  OC BK 1001 00 01 100## 110## 130##
• For BK format, a record must have a title statement and the field cannot be repeated:
  OC BK 1002 01 01 245##

**Subtable D**
Structure of the table:
Col.1  D
Col. 2 Record format as registered in the FMT field in the document record. XX should be entered if check valid for all formats.
Col. 3 error number - 5000 – 8999 - as listed in the check_doc.lng table (located in the tab directory of the library)
Col. 4 field code for first part of condition (e.g. 100##). Note that wild card may be used.
Col. 5 defines whether the check relates to the field being present or not
  • Y=present,
  • N=not present
Col. 6 field code for second condition of the check. Use # for marking wild card (e.g. 245##). Leave blank of there is no second condition.
Col. 7 defines whether the check relates to the field being present or not
  • Y=present,
  • N=not present

Example:
• If 2450# is present then no main entry (1XX) should be present:
  D  BK 7003 2450#  Y 1####  N

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>!--!!--!!--!!--!!--!!--!!--!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC XX 5001 00 01 100## 110## 111## 130##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC XX 5002 01 01 245##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC BK 5003 01 01 260##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC SE 5007 01 01 310##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC XX 5008 01 01 008##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC CF 5120 01 01 256##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC CF 5121 01 99 538##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
check_doc_field_006

Location of the table: tab directory of the BIB library

Purpose of the table: Checks on MARC21 tag 006

Structure of the table:

Col.1 Format of document to check:. Use XX for any format
Col. 2 The field position used as a matching point for the character specified in column 3
Col. 3 Match character. The field character at the offset given in column 2 must match this value for the check to apply. A space here means that the check is always done.
Col. 4 Start of position range to check
Col. 5 End of position range to check
Col. 6 Type of check to apply:
   • 1 = check for valid values
   • 2 = check for obsolete values
   • 3 = run another check program
Col. 7 Check values. Value depends on the check type:
   • 1 = list of values, separated by commas, that are valid for this position range. If the position range is only one character, the commas may be omitted.
   • 2 = list of values, separated by commas, that may be present but are obsolete. If the position range is only one character, the commas may be omitted.
   • 3 = name of check program to run

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX 000-017 3 check_fixed_field_length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX 000 1 acdefgijklmoprst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX 000 a 001 1 ^abcdefgijklmop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX 000 a 002 1 ^abcdefgijklmop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX 000 a 003 1 ^abcdefgijklmop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX 000 a 004 1 ^abcdefgijklmop</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX 000 a 001-004 3 check_val_left_just</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 check_val_alpha_order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
check_doc_field_007

Location of the table: tab directory of the BIB library

Purpose of the table: Checks on MARC21 tag 007

Structure of the table:

Col.1  Format of document to check:. Use XX for any format
Col. 2  The field position used as a matching point for the character
        specified in column 3
Col. 3  Match character. The field character at the offset given in column
        2 must match this value for the check to apply. A space here
        means that the check is always done.
Col. 4  Start of position range to check
Col. 5  End of position range to check
Col. 6  Type of check to apply:
        • 1 = check for valid values
        • 2 = check for obsolete values
        • 3 = run another check program
Col. 7  Check values. Value depends on the check type:
        • 1 = list of values, separated by commas, that are valid for
          this position range. If the position range is only one
          character, the commas may be omitted.
        • 2 = list of values, separated by commas, that may be present
          but are obsolete. If the position range is only one character,
          the commas may be omitted.
        • 3 = name of check program to run

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
</table>
| ! | ! | ! | ! | ! | ! | ! >
XX 000 1 acdfghkmoqrstvz
XX 000 a 000-007 3 check_fixed_field_length
XX 000 a 001 1 dgjkqrswzy
XX 000 a 001 2 abcehijmnopvtwx

check_doc_field_008

Location of the table: tab directory of the library

Purpose of the table: Checks on MARC21 tag 008

Structure of the table:
Col. 1 Format of document to check:. Use XX for any format
Col. 2 The field position used as a matching point for the character specified in column 3
Col. 3 Match character. The field character at the offset given in column 2 must match this value for the check to apply. A space here means that the check is always done.
Col. 4 Start of position range to check
Col. 5 End of position range to check
Col. 6 Type of check to apply:
- 1 = check for valid values
- 2 = check for obsolete values
- 3 = run another check program
Col. 7 Check values. Value depends on the check type:
- 1 = list of values, separated by commas, that are valid for this position range. If the position range is only one character, the commas may be omitted.
- 2 = list of values, separated by commas, that may be present but are obsolete. If the position range is only one character, the commas may be omitted.
- 3 = name of check program to run

Example of the table:

<table>
<thead>
<tr>
<th>Col.1</th>
<th>Col. 2</th>
<th>Col. 3</th>
<th>Col. 4</th>
<th>Col. 5</th>
<th>Col. 6</th>
<th>Col. 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>000-039</td>
<td>3</td>
<td>check_fixed_field_length</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006</td>
<td>1</td>
<td>bcdeikmpqrsstu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 b</td>
<td>007-014</td>
<td>3 check_val_blank</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 c</td>
<td>007-010</td>
<td>3 check_val_date_4_or_u</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 c</td>
<td>011-014</td>
<td>3 check_val_all_9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 d</td>
<td>007-010</td>
<td>3 check_val_date_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 d</td>
<td>011-014</td>
<td>3 check_val_date_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 e</td>
<td>007-014</td>
<td>3 check_val_date_8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 i</td>
<td>007-010</td>
<td>3 check_val_date_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 i</td>
<td>011-014</td>
<td>3 check_val_date_4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>006 k</td>
<td>007-010</td>
<td>3 check_val_date_4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**check_doc_field_ldr**

Location of the table: tab directory of the library

Purpose of the table: Checks on the MARC ldr

Structure of the table:

Col. 1 Format of document to check:. Use XX for any format
Col. 2  The field position used as a matching point for the character specified in column 3

Col. 3  Match character. The field character at the offset given in column 2 must match this value for the check to apply. A space here means that the check is always done.

Col. 4  Start of position range to check

Col. 5  End of position range to check

Col. 6  Type of check to apply:
    - 1 = check for valid values
    - 2 = check for obsolete values
    - 3 = run another check program

Col. 7  Check values. Value depends on the check type:
    - 1 = list of values, separated by commas, that are valid for this position range. If the position range is only one character, the commas may be omitted.
    - 2 = list of values, separated by commas, that may be present but are obsolete. If the position range is only one character, the commas may be omitted.
    - 3 = name of check program to run

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>000-023</td>
<td>3</td>
<td>check_fixed_field_length</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XX</td>
<td>005</td>
<td>1</td>
<td>acdnp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BK</td>
<td>006</td>
<td>1</td>
<td>at</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 bn</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CF</td>
<td>006</td>
<td>1</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP</td>
<td>006</td>
<td>1</td>
<td>ef</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MU</td>
<td>006</td>
<td>1</td>
<td>cdij</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**check_doc_line**

Location of the table: tab directory of the library

Purpose of the table: Checks on tags

The **AL** section of this table enables you to define the following checks:
- Valid indicators and/or sub-field codes for the tag.
- Presence of mandatory sub-fields.
- Non-repeatability of non-repeatable sub-fields.
The **D** section of this table enables you to determine the rules for checking dependencies among sub-fields of a single field.

Structure of the table:
- **Col. 1** Type of check - **AL** or **D**
- **col. 2** Record format as in the FMT field. Use XX for all formats.

**Format of **AL**(allowed) table:**
- **col. 4** Tag
- **col. 5**
  1) - (hyphen) for indicator
  2) Sub-field code
- **col. 6**
  1) Indicator - possible values for 1st indicator
  2) Sub-field - 0=non-mandatory; 1=mandatory
- **col. 7**
  1) Indicator - possible values for 2nd indicator
  2) Sub-field - 1=not repeatable; 2-9=sub-field can be repeated up to the number of times entered here; "-"=unlimited occurrences.

Note: The indicator portion (for all formats) must be listed before the sub-field portion, for each field.

Example of the **AL** table:

```
! 2 - 4 5 6 7
!!-!!-!!!!-!!!---!-!-!
AL XX 010 -
AL XX 010 a 0 1
AL XX 010 b 0 -
AL XX 010 z 0 -
AL XX 010 8 0 -
...
AL XX 050 - 0
AL XX 050 - 0 0
AL XX 050 - 1 4
AL XX 050 a 1 -
AL XX 050 b 0 1
AL XX 050 3 0 1
AL XX 050 8 0 -
...
```

**Format of **D**(dependency between sub-fields) table**
- **col. 3** Error number. of message to be displayed as found in the relevant table in alephe/error_lng
- **col. 4** Tag
- **col. 5** First sub-field code
- **col. 6** Specific content of sub-field
- **col. 7** Type of dependency
  Y sub-field present
check_doc_line_contents

Location of the table: tab directory of the library

Purpose of the table: Checks on contents of a tag

This table checks the contents of defined fields.

Structure of the table:
- col. 1 format (Use ## for all).
- col. 2 tag and indicators
- col. 3 sub-field code (If blank field is taken as is).
- col. 4 name of check program, values are:
  - isbn - verifies that the ISBN entered in the field is a valid ISBN.
  - issn - verifies that the ISSN entered in the field is a valid ISSN.
  - length - verifies that the length of a numeric string matches the values defined in column 5.
  - number_length - verifies that the number_length of a numeric string matches the values defined in column 5.
  - range - verifies that the numeric string entered in the field matches the range defined in column 5.
- col. 5 value(s) to check. Valid options are:
  - range <from> <to>
  - For example, for sub-field $c$ of MARC 21 field 260, enter reasonable values for the range of the year (e.g. 1750-2003):
    - ## 260 c range 1750 2003
  - length <length>
For example, for sub-field $c$ of MARC 21 field 260, the length is "4" for the year:

```
## 260   c length               4
```

Example of the table:

```
!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!
## 020## a isbn
## 022## a issn
## 7#### x issn
## 260## c range               1850 2002
## 022## a length               9
```

**Note:** number-length. This is the same as length, but takes into account only the first numerical string. Example:
c.1996:xx is translated to 1996 and then counted.

**check_doc_mandatory**

Location of the table: tab directory of the library

Purpose of the table: Checks on forbidden errors in a document record

This table can be used to define whether cataloging check routines should activate a trigger or be defined as forbidden.

For example, if the check_doc_doc table is used to define that a record must have a 245 field (e.g. OC XX 5002 01 01 245##) then you can set error message 5002 to activate a trigger or be defined as forbidden.

In addition, it is possible to define (through the check type column - col.1) that the error message activates a trigger or is forbidden only in particular instances for example, when records are saved – check type CATALOG-INSERT).

Note that column 3 defines whether the error activates a trigger or is forbidden:

- **T** - Error writes a cataloging trigger record, but allows update of the database. Triggers can be later retrieved through the Cataloging module or in batch mode.
- **M** - Forbidden error, does not allow database update. For example:
  - if the record is missing the 245 field,
  - if the relevant error is set to M
  - and the check type is CATALOG-INSERT

Error codes from 5001 to 7000 are defined by the user in check_doc.<lng>.
Error codes from 0001 to 5000 and from 7001 to 9999 are system defined, and can be found in alephe/error_<lng>/check_doc

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Check type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The check type defines when the check program is performed.</td>
</tr>
</tbody>
</table>
Valid values are:

**CATALOG-INSERT**: performed when the cataloging record is saved, updated or when the Check Record option is selected from the Cataloging module.

**CATALOG-DELETE**: performed when the Delete Record from Server option is selected from the Cataloging module.

**BATCH-DELETE**: performed when the Delete Bibliographic Records (p-manage-33) batch process is run.

**NAV-MAP-DELETE**: performed when the Total Delete option is selected from the Record Manager of the Cataloging module to delete a bibliographic record together with all associated records.

**P-MANAGE-25**: performed when the P-MANAGE-25 check option is selected when running the FIX and Check Catalog Records (p-manage-25) batch process.

Note that if this column is left blank, then the error code defined in column 3, applies for all check types.

**P-MANAGE-11**: performed when the Check Database Tables (p-manage-11) batch process is run.

**Z39-INSERT**: performed when a record is inserted via Z39.50 ES Update.

**Z39-REPLACE**: performed when a record is replaced via Z39.50 ES Update.

**Z39-DELETE**: performed when a record is deleted via Z39.50 ES Update.

<table>
<thead>
<tr>
<th>Col. 2</th>
<th>Identifying number of the check program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 3</td>
<td>T - Errors that writes a cataloging trigger record, but allows database update. Triggers can be retrieved in the Cataloging client or in batch mode.</td>
</tr>
<tr>
<td>Col. 4</td>
<td>M - Error which does not allow database update.</td>
</tr>
</tbody>
</table>

Note: Defining error 9999 as T allows an application to override the system limit of number of errors preventing a record update (set at 40). If 9999 is not defined in check_doc_manatory it defaults to M(andatory).

Examples of system defined codes:
- 0101 – non unique index entry
- 0110 - new access entry

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td></td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>CATALOG-DELETE 0011 M ADM record $1 in library $2 points to current document with link type $3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATALOG-DELETE 0012 M HOL record $1 in library $2 points to current document with link type $3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
check_doc_new_acc

Location of the table: tab directory of the library

Purpose of the table: Fields that should be ignored when checking new headings in the headings list

This table defines the fields that are indexed but should be **IGNORED** for purposes of the check messages regarding new acc headings.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Field Code of the fields that should be ignored while checking for unique headings in the Heading List (ACC index). Use the hash (#) as a placeholder for undefined tags and/or indicators.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="check_doc_new_acc_table.png" alt="Table" /></td>
<td></td>
</tr>
</tbody>
</table>

check_doc_new_acc_aut

Location of the table: tab directory of the **BIB** library

Purpose of the table: Fields that should be ignored when checking new headings in the headings list together with check in the AUT library

This table defines the fields that should be **IGNORED** when checking for unique Headings index (ACC) entries, combined with a check in the relevant AUT library as defined in tab_aut.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Field Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="check_doc_new_acc_aut_table.png" alt="Table" /></td>
<td></td>
</tr>
</tbody>
</table>
check_doc_tag_text

Location of the table: tab directory of the library

Purpose of the table: Check of pre-defined texts for sub-fields

The check routine allows for validation of pre-defined texts for fields (as defined in tag_text)

Structure of the table:
- Col. 1 Tag + indicators
- Col. 2 ALPHA
- Col. 3 Sub-field
- Col. 4 Value or description

Note: Use # in cols. 1-3 in order to indicate any character

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>LKR</td>
<td>L</td>
<td>a</td>
<td>UP</td>
</tr>
<tr>
<td>LKR</td>
<td>L</td>
<td>a</td>
<td>ITM</td>
</tr>
<tr>
<td>LKR</td>
<td>L</td>
<td>a</td>
<td>ANA</td>
</tr>
<tr>
<td>LKR</td>
<td>L</td>
<td>a</td>
<td>PAR</td>
</tr>
<tr>
<td>655#</td>
<td>#</td>
<td>L</td>
<td>a</td>
</tr>
<tr>
<td>655#</td>
<td>#</td>
<td>L</td>
<td>a</td>
</tr>
<tr>
<td>655#</td>
<td>#</td>
<td>L</td>
<td>a</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STA#</td>
<td>L</td>
<td>a</td>
<td>CIRC-CREATED</td>
</tr>
<tr>
<td>STA#</td>
<td>L</td>
<td>a</td>
<td>PROVISIONAL</td>
</tr>
<tr>
<td>STA#</td>
<td>L</td>
<td>a</td>
<td>SUPPRESSED</td>
</tr>
<tr>
<td>STA#</td>
<td>L</td>
<td>a</td>
<td>DELETED</td>
</tr>
</tbody>
</table>

check_doc_unique_index

Location of the table: tab directory of the library

Purpose of the table: Fields that should be ignored when checking for duplication
Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Field Code</th>
</tr>
</thead>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>Col. 1 Field Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>010##</td>
</tr>
<tr>
<td>024##</td>
</tr>
<tr>
<td>028##</td>
</tr>
<tr>
<td>037##</td>
</tr>
<tr>
<td>086##</td>
</tr>
</tbody>
</table>

**codes.<lng>**

**Location of the table:** pc_tab/catalog directory of the library

**Purpose of the table:** Valid tags and aliases for the Cataloging Client

This table defines the valid tags and aliases for the Cataloging Client. In the on-line cataloging module the list can be activated using the F5 function key. When cataloging, the cataloging template can be set to display only the MARC tag, or the MARC tag and a library defined alias (e.g. 245 - Main title).

The option for work with aliases is set in:

DisplayTagInfo (=Y, or =N) in the [Editor] section of the PC’s catalog.ini file.

If option not set, defaults to N.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Y or N</td>
</tr>
<tr>
<td></td>
<td>• Y - tag can be accessed from the ‘new field (F5)’ option in the edit menu on the client</td>
</tr>
<tr>
<td></td>
<td>• N - will not be accessible from the ‘new field (F5)’ option. However, if the tag is displayed in the template, or record, it will display with its alias, and with its description on the status bar.</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Y or N</td>
</tr>
<tr>
<td></td>
<td>• Y - tag can be edited only by form</td>
</tr>
<tr>
<td></td>
<td>• N - tag can be edited using the editor (via the template) or a form</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Y or N</td>
</tr>
<tr>
<td></td>
<td>• Y - tag cannot have sub-fields</td>
</tr>
<tr>
<td></td>
<td>• N - tag has sub-fields</td>
</tr>
<tr>
<td>Col. 5</td>
<td>ALPHA of name (alias)</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Name (alias) of tag. This information displays on the template (if DisplayTagInfo in the catalog.ini file is set to Y)</td>
</tr>
<tr>
<td>Col. 7</td>
<td>ALPHA of tag description</td>
</tr>
<tr>
<td>Col. 8</td>
<td>Tag description. This information displays on the status bar and</td>
</tr>
</tbody>
</table>
Example of the table:

<p>| | | | | | | | | | | | | | | | | | | |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |</p>
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDR</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>L</td>
<td>Leader</td>
<td>L</td>
<td>Leader (LDR)</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>L</td>
<td>Control No.</td>
<td>L</td>
<td>Control Number</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>L</td>
<td>Control No. ID</td>
<td>L</td>
<td>Control Number Identifier</td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>L</td>
<td>Date and Time</td>
<td>L</td>
<td>Date and Time of Latest Transaction</td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>L</td>
<td>Add. Fixed Data</td>
<td>L</td>
<td>Fixed Length Data/Additional Characteristics</td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>L</td>
<td>Phys. Descrip.</td>
<td>L</td>
<td>Physical Description Fixed Field</td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>L</td>
<td>Fixed Data</td>
<td>L</td>
<td>Fixed Length Data Elements</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Corporate Name</td>
<td>L</td>
<td>Main Entry - Corporate Name</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Meeting Name</td>
<td>L</td>
<td>Main Entry - Meeting Name</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Main Uni Title</td>
<td>L</td>
<td>Main Entry - Uniform Title</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Abbrev. Title</td>
<td>L</td>
<td>Abbreviated Title</td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Acron. Title</td>
<td>L</td>
<td>Acronym or Shortened Title [OBSOLETE]</td>
<td></td>
</tr>
<tr>
<td>212</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Var. Acc. Title</td>
<td>L</td>
<td>Variant Access Title [OBSOLETE]</td>
<td></td>
</tr>
<tr>
<td>214</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Augmented Title</td>
<td>L</td>
<td>Augmented Title [OBSOLETE]</td>
<td></td>
</tr>
<tr>
<td>222</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>L</td>
<td>Key Title</td>
<td>L</td>
<td>Key Title</td>
<td></td>
</tr>
</tbody>
</table>

**drm_routines**

**Location of the table:** tab directory of the **BIB** library

**Purpose of the table:** Check routines for user authorization and payment related to digital rights

This table check routines for user authorization and payment related to digital rights.

It defines the routine to use to determine whether the patron has the right to view an electronic object. The view routines are set in the "AUT" lines. The table also defines the routine to use to set the cost of the view. The payment routines are set in the "PAY" lines, and are not yet active.

The routines available for viewing permission are

- **bor_permission:** currently available routine is "1", which checks the "Link Permission" in the patron's profile (Z61-856-PERMISSION).
- **course_enrollment:** currently available routine is "3", which checks whether the patron is listed under the course in the Z107 table.

**Structure of the table:**

- **Col. 1** Code type
  - AUT - check authorization
  - PAY - commit payment (patron and vendor)
- **Col. 2** Procedure name
Col. 3  Check Procedure
Col. 4  Procedure parameters

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUT bor_permission</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUT bor_status</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUT course_enrollment</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAY bor_permission</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAY bor_status</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAY course_enrollment</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### edi_in_attr

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** EDI incoming messages attributes

This table is for defining special attributes of incoming EDI messages as needed for specific vendors.

- The first field defines the Vendor code
- The second field contain up to 100 switches each 1 character in length
- The third field identifies the label from tab_character_conversion_line that determines the program and the tables needed for the character conversion.

**Structure of the table:**
- Col. 1  Vendor code
- Col. 2  Switches
- Col. 3  Character conversion identifier

**Example of the table:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWETS</td>
<td>UTF_TO_8859_8</td>
</tr>
</tbody>
</table>

### edi_out_attr

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** EDI outgoing messages attributes
This table is for defining special attributes of outgoing EDI messages as needed for specific vendors.

- The first field defines the Vendor code
- The second field contains up to 100 switches each 1 character in length
- The third field identifies the label from tab_character_conversion_line that determines the program and the tables needed for the character conversion.

Description of the different switches

**EDI-OUT-REMOTE-FILE-NAME**
- 0 (default) = ALEPH’s file name.
- 1 = Blackwell format of remote file name to be sent through ftp. The format is: EPInn..DDDMYY.R. DDMYY is the date and nn is the sequence number of the file sent during the same day, and all the rest of the characters are constant.
- 2 = EBSCO format. "CLAIMS.FIL"
- 3 = Ingram format. 8.3 format. 8 numeric characters (taken from the clock) and 3 constant characters - epo for ORDERS.

Additional values to this switch (2-9) may be added easily. Each value corresponds to a different program.

**EDI-OUT-FTP-MODE**
- A (default) = ASCII mode will be used for sending the outgoing messages
- B = Binary mode will be used for sending the outgoing messages

**EDI-OUT-ORDER-NUMBER-1**
- 0 (default) = Z68-ORDER-NUMBER-1 is not added to the EDI ORDERS message as a second order number.
- 1 = Z68-ORDER-NUMBER-1 is added to the EDI ORDERS message as a second order number.

**EDI-OUT-ORDER-NUMBER-2**
- 0 (default) = Z68-ORDER-NUMBER-2 is not added to the EDI ORDERS message as a third order number.
- 1 = Z68-ORDER-NUMBER-2 is added to the EDI ORDERS message as a third order number.

Structure of the table:
- Col. 1 Vendor code
- Col. 2 Switches
- Col. 3 Character conversion identifier

Example of the table:

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Switches</th>
<th>Character conversion identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLACKWELL</td>
<td>1A11</td>
<td>UTF_TO_8859_8</td>
</tr>
<tr>
<td>SWETS</td>
<td>0A</td>
<td></td>
</tr>
</tbody>
</table>
**edit_doc.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Document format definition for GUI/WWW display

Special note: This table can be duplicated with any base extension as defined in cols. 4, 5 and 6 of tab_base.<lng>. For example edit_doc.serials for the logical base "serials".

The edit_doc table defines a concatenation of a number of paragraphs It is used in conjunction with edit_field and edit_paragraph.

Structure of the table:

Col. 1  document format number. Note that format 012 is used by the system for display of bibliographic information on the OPAC list of items display

Col. 2  prefix

Col. 3  New line indicator:

## - indicates that data following should start on new line

Col. 4  New line indicator

## - data following will start on a new line preceded by blank line

Col. 5  paragraph identifier as defined in edit_paragraph

Example of the table:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>037 System number^</td>
<td>##</td>
<td>##</td>
<td>001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>037</td>
<td></td>
<td></td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>037</td>
<td></td>
<td></td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>037</td>
<td></td>
<td></td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>037</td>
<td></td>
<td></td>
<td>140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>037</td>
<td></td>
<td></td>
<td>004</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038 System number^</td>
<td>##</td>
<td>001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>038</td>
<td></td>
<td></td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038</td>
<td></td>
<td></td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038</td>
<td></td>
<td></td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038</td>
<td></td>
<td></td>
<td>140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>038</td>
<td></td>
<td></td>
<td>004</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**edit_doc_777**

Location of the table: tab directory of the library

Purpose of the table: Record conversion format for export to reference manager such as Endnote
This table defines the format 777 for the Save/Mail option in OPAC. The format converts the bibliographic records into "ISI ResearchSoft Tagged Output Format" – the format used to import records into reference managers, for e.g. Endnote. The conversion is driven by the table edit_doc_777. This function uses the form www-set-mail-777-00 (in the directory xxx01/root/form_eng) for output.

Structure of the table:

Col. 1  Tags from ISI ResearchSoft Tagged Output Format. The required tags TY (beginning of record) and ER (end of record) are filled in automatically by the system

Col. 2  Field tag + indicators

Col. 3  1) Subfield(s) of the field (blank indicates entire field).
        2) Minus (-) sign followed by subfield(s) to be stripped

Col. 4  Starting position: Defines the position from which to take data from a fixed field. For example, if column 1 is YR, the year might be taken from a fixed field. In this case, you will define the position in the fixed field from which to commence taking 4 positions, counting from base 01. If the fixed field has a subfield code, add 3 to the starting position in order to take it into account.

        e.g. 0008 to define the 8th position of the 008 field in MARC21, 0013 to define the 9th position of the 100 field in UNIMARC.

Example of the table:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>!</td>
<td>!-</td>
<td>!-</td>
<td>!-</td>
</tr>
<tr>
<td>ID</td>
<td>SID</td>
<td>bc</td>
<td></td>
</tr>
<tr>
<td>AU</td>
<td>100##</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>AU</td>
<td>110##</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>AU</td>
<td>700##</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>AU</td>
<td>710##</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>AU</td>
<td>711##</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>TI</td>
<td>245##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td>246##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td>240##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>440##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>490##</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PY</td>
<td>008##</td>
<td>0008</td>
<td></td>
</tr>
</tbody>
</table>

edit_doc_999.<lng>

Location of the table: tab directory of the library

Purpose of the table: Full document format
**Special note:** This table can be duplicated with any base extension as defined in cols. 4, 5 and 6 of tab_base.<lng>. For example edit_doc_999.eng.serials for the logical base "serials".

This table defines the **linked** display format for documents for viewing in the GUI and WEB OPAC. Display can include "expanded" fields that might be added from HOLdings, items or ACC (Z01) fields.

In addition, there are definitions that determine how links to items display (ITM...), and whether links to other records will display.

**NOTE:** the links are BROWSE and FIND links to Authority and Word lists as defined in tab11_aut and tab11_word in of the tab directory of the library.

Note the feature for external links: the link (with or without the copyright flag) passes through a program that checks the patron permission to view the link. Dependent on profile definitions a patron will/will not be able to see these links (as defined in Z61-856-PERMISSION).

Note the following definitions – in addition to field tag - for col. 1

- **UP** - display of the Up link in linked records
- **PAR** - display of Parallel records
- **DN** - display of Down link in linked records
- **ANU** - (analytical up) for MAB format
- **AND** - (analytical down) for MAB format.
- **ANU** and **AND** are used together with the update_z103_mab_ana program that can be added to tab_z103.
- **ICON** - For interaction with Syndetics.
- **ITM** - hypertext link to holdings information of display of Sub-library (GUI SEARCH)
- **ITMG** - hypertext link to Global-Holdings (all items) (GUI SEARCH)
- **ITM1** - global; i.e. displays one ITM line in doc; leads to display of:
  - bib record display
  - information from HOL records (if set to display)
  - all items in one list (all years/vols/libraries)
- **ITM2** by year; i.e. intended for serials, displays ITM for each year (based on Z30-YEAR); leads to display of:
  - bib record display
  - item records for the particular year.
- **ITM3** by sub-library; i.e. displays ITM line for each sub-library (taking items and HOL records together); leads to display of:
  - bib record display
  - information from HOL records (if set to display) of the sub-library
  - item records of the particular sub-library
- **ITM4** link to remote library items, with no circulation information (ACC central database)
- **ITM5** - by sub-library (taken from default sub-library in patron profile). If patron is signed-in, uses default sub-library from personal patron profile. If
signed-in patron does not have profile, or patron is not signed-in, takes IP address of the station (if there are patron records with "IP" as user-id, and patron personal profiles have been defined). If no personal patron profile can be matched, the personal patron profile of user-id ALEPH is used. Leads to display of:

- bib record display
- information from HOL records (if set to display) of the sub-library
- item records of the particular sub-library
- item records of the last year of issue item records

Note -- this is a way to control access to databases by IP address - by creating a PROFILE for denied bases for 'ALEPH' patrons, for IP users, and for signed-in users.

- **ITM6** - link for formatting remote items data. Used by the Ex Libris Gate (Z39.50 and ALEPH), where link to remote holdings could be through "jump to" (ALEPH link) or OPAC record syntax (Z39.50). Therefore, relevant only within EXTnn libraries, and for installations using the Ex Libris Gate.
- **ITMH** - link to holdings and items, each sublibrary + collection listed on a separate line. Note that the holdings and items display is NOT sensitive to collection, and displays the data at the sublibrary level. Up to 200 sublibraries can be displayed. **Note:** Will only display if there is a holding (HOL) record
- **ITML** - link to remote item information for non ALEPH local systems. The link based on the LOW tag which contains the sub library code of the local library.
- **LOC** - display of item information built using `expand_doc_bib_loc_usm` and `expand_doc_bib_loc_disp`. Links to items in the same way as ITM3 - i.e. tag text links to library information, and location links to list of items display. **NOTE** that in order for this to work correctly, col. 10 must have an L definition.
- **PST** item information, built using `expand_doc_bib_loc_[n]...` and `expand_doc_sort_loc...` Links the same as ITM3; i.e. tag text links to library information, and location links to list of items display.
- **DIS** - display line (could be to define a blank line)
- **ERR** - if there is a pointer to a non-existing BIB record, line 1000 from www_f_heading is displayed in this line. Therefore, this code **must** be present in the table.

Note that it is possible to define more than one ITMx line; e.g. a library might want to define ITM2 and ITM3.

Order of Sub-Libraries:
- The first sub-library to display will be the default sub-library, set through the profile record, (Z61-BOR-SEARCH-SUB-LIBRARY).
- The order of the rest sub-libraries will be alphabetical. This option concern the display codes ITM3 and ITM4.
- as set in the table tab_sub_lib_sort
Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Format – display can be sensitive to bibliographic format. Use # for all formats:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE SYS</td>
<td>D LSys. no. Serial Y</td>
</tr>
<tr>
<td>BK SYS</td>
<td>D LSys. no. Book Y</td>
</tr>
<tr>
<td>AU SYS</td>
<td>D LSys. no. Aut Y</td>
</tr>
<tr>
<td>## SYS</td>
<td>D LSys. no. Y E</td>
</tr>
</tbody>
</table>

Col. 2  Field tag - use wild card # as necessary for indicators

Col. 3  Sub-field to filter on

Col. 4  Contents to filter for (use 3+4 for filter on language)

(e.g. if there is "2 usm50" in these columns, only records with usm50 in subfield 2 will have this field indexed). If the contents is prefixed by a hyphen, the content is considered negative (e.g. if there is "2 -usm50" in these columns, only records that do not have usm50 in subfield 2 will have this field indexed).

**NOTE!!** Filter text must be entered in lower case (irrespective of case in the record itself). Use * to indicate truncated text, use # to indicate single wild card for text match.

Col. 5  Display sub-field

Col. 6  Edit_field indicator

Col. 7  ALPHA

Col. 8  Heading

Col. 9  Display new line. (Y or blank)

In the WEB OPAC multiple entries of the same field can be displayed in a run-on mode, in effect forming a single block of data. However, the size of this block of data is limited to 2000 characters. Long, repeated fields might be therefore be cut off. In order to avoid this, repeated fields can be divided into separate entries - each on a separate line.

Col. 10 Display Link Type:

**For WEB OPAC:**

- **S** - Simple-link : Wraps URL in free text with <a href> tags
- **Z** - Link to Services (as defined in tab_service)
- **A** - External Link
- **L** - Links from LOC/PST line to items display
- **M** - Electronic resource link
  - in effect only if the field contains subfield "u"
  - uses col.3 of tab_buf_z403 instead of cols.3, 4 and 6 of edit_doc_999 for filter and link definitions

**For GUI SEARCH:**

- **Z** - Find + Browse Link
- **S** - Browse Link
- **F** - Find link
Q - Links (BIB to BIB and AUT to AUT links)
E - External Link, uses the 856 line in edit_field.<lng> for field formatting.

Note the difference between S and other link-types like M or L. M / L create a link BACK TO THE SERVER with the appropriate information, while "S" type merely wraps the URL with html tags so that no copy-rights warning is shown or patron authorization is checked.

Col. 11 Display language. No longer used.

Col. 12 E for display of END-GROUP. This defines the end of a group of alternative field codes. The first occurrence of the first field of the group will display, other fields will be ignored.

For fields with only a single line, an 'E' should be registered at the end of the line. If there is more than one line per field, the 'E' should be registered ONLY on the last line (end group). In this case, DO NOT put an 'E' for any of the other lines.

Col. 13 The ACC (Z01) file that should be accessed when "headings" are requested in those fields where defined sub-fields are indexed on different ACC lists (e.g. 260 $$a is indexed in the PLAce list, 260 $$b is indexed in the PUBlisher list.). In this case, sub-field elements of the tag that are not indexed should be marked with XXX. Used only in GUI SEARCH. For WEB OPAC, this functionality is defined in col. 3 of tab_service.

Example:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
|## 260## | a | D | LPlace | Y | Z | E PLA
|## 260## | b | D | LPublisher | Y | Z | E PUB
|## 260## | c | D | LYear | Y | E | E XXX

Col. 14 W, G or blank:

W - display in the WEB OPAC only
G - display in GUI SEARCH only
Blank - display in both WEB and OPAC

Col. 15 Brief format. "00" is used except in special cases

Col. 16 Limit of linked records that will display. If there are more than the given number, system displays a single line that creates a set and displays a Brief List instead. Default is 99

Example of the table:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>
edit_doc_999_aut_<aut_lib>.lng

Location of the table: tab directory of the BIB library

Purpose of the table: display of AUT information on cross references

In the Web OPAC, clicking on a reference displays the linked Authority Record. The display of the record is defined in the edit_doc_999_aut_<aut_lib>.lng table.

In MARC 21, sub-field $w, 4th position (that is, $wxxxx4) - of both the 4XX and 5XX - contains a code that enables the generation or suppression of a cross-reference from 4XX or 5XX fields. If the fourth position of sub-field $w contains a blank or contains an ‘n’ (or the sub-field does not exist), then the cross-reference is generated and the fields are shown in all relevant places: browse list, expanded authority record, under the heading in the brief record, and so on. If the fourth position of sub-field $w contains something different, then the cross-reference is suppressed.

To support this, it is necessary to use the filter columns in this table for the display and in the tab20 table for the enrichment definitions. The following is an example of the lines for this table:

| ## 4#### w !!!n | D LSeen from | Y E |
| ## 4#### w !!! * | D LSeen from | Y E |
| ## 4#### w - | D LSeen from | Y E |
| ## 5#### w g!!n | D LBroader term | Y X E |
| ## 5#### w g!! * | D LBroader term | Y X E |
| ## 5#### w h!!n | D LNarrower term | Y X E |
| ## 5#### w h!! * | D LNarrower term | Y X E |
| ## 5#### w h | D LNarrower term | Y X E |
| ## 5#### w g | D LBroader term | Y X E |
| ## 5#### w - | D LSee also | Y E |

Structure of the table: (For full details of the structure of the table see edit_doc_999.lng)
Col. 3 Sub-field to filter on
Col. 4 Contents to filter for
Col. 5 Display sub-field
Col. 6 Edit_field indicator as defined in the authority library
Col. 7 ALPHA
Col. 8 Heading
Col. 9 Display new line (not active – define as Y)
Col. 10 Display Link Type:
  This column is relevant only for record fields that constitute a heading field.
  **Blank** - The field always displays. If the field does NOT exist as a heading in the database, the field simply displays. If the field DOES exist as a heading in the database, the field displays with FIND and BROWSE link.
  **X** - The field displays only if the field exists as a heading in the database.
Col. 11 Display language - no longer used.
Col. 12 E for display of END-GROUP. If there is more than one line per field, put E ONLY for the last line (end group). In this case, DO NOT put an E for any of the other lines.
Col. 13 Not used
Col. 14 Not used

Example of the table:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Field Type</th>
<th>Display</th>
<th>Line</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>## SYS</td>
<td>D LSys. no.</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 1####</td>
<td>D LHeading</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 260##</td>
<td>D LSub. CSR</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 360##</td>
<td>D LSub. CSAR</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 4#### w !!!n</td>
<td>D LSeen from</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 4#### w !!! *</td>
<td>D LSeen from</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 4#### w -</td>
<td>D LSeen from</td>
<td>Y</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 5#### w g</td>
<td>D LBroader term</td>
<td>Y X</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>## 5#### w g!!n</td>
<td>D LBroader term</td>
<td>Y X</td>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>

**edit_field.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Editing of fields and headings for OPAC display and print

This table is used to define the display and print options of a tag and of entries in the Browse Headings lists. It is used in conjunction with edit_paragraph and edit_doc. If a tag is not defined in this table, it will be displayed as is. Therefore, tags entered in the database **without** punctuation, should be defined here in order to add punctuation.
**Line type Identifier:**

1 - defines field code (col.3), line id (col.4), editing (col.5), subfields to strip (col.6), field prefix (col.9), and field suffix (col.10)

2 - defines subfield codes, prefixes and suffixes.

The ID of the line (col.4) in this edit_field table serves as a link to the edit_field ID defined in the

- edit_paragraph.<lng> (col.3),
- edit_doc_999.<lng> (col.6), and
- www_tab_short.<lng> (col.8) tables

in the library's tab directory.

In addition, there are set codes, as follows:

- **H** is used for ACC (headings) display in the WEB OPAC
- **C** is used for ACC (headings) display in GUI clients
- **S** is used for pc_tab_short display in GUI
- **L** is used for location (using expand_doc_bib_loc_usm and expand_doc_bib_loc_disp) with link to display of library info and display of items list
- **1** in HOL is used for 852 display in the holding window
- **2** is used for 852 in items (copies) display
- **B** is used for brief records (Z0101) display

The following are conventions in the USM01 tables:

- **D** is used for "name tags" format in GUI and WWW (edit_doc_999) –
- **3** is used for the paragraphs that make up the catalog card format (edit_doc.eng - format 037)
- **P** is used for the paragraphs that make up the citation format (edit_doc.eng - format 040)

# is a wild card throughout the table for alpha, tags, indicators, subfields and edit_field ID.

In order to cover all instances not specifically listed in the table, the last 2 lines of this table should be:

```
1 # ###### \\
2 # A ^
```

This table can be up to 1000 lines long.

**Structure of the table:**

- **Col. 1** line type identifier – line 1 or line 2:
- **Col. 2** Not used. Should have #
- **Col. 3** tag+indicator. Note that Z0101 is used for brief records display
- **Col. 4** ID of edit_field line.
- **Col. 5** Filter:
  - A or blank filters for sub-fields, but does not change their
order
- **B** filters for sub-fields and sets their order

Col. 6  - (minus) followed by sub-field codes to strip (e.g. -w)
Col. 7  sub-field code, or # to indicate "all sub-fields". Up to 20 sub-fields can be defined
Col. 8  A to indicate all occurrences of sub-field(s) within the tag
Col. 9  sub-field or field prefix
Col. 10 sub-field or field suffix.

## May be entered in order to start each repeated field on a new line.

Example of the table:

<table>
<thead>
<tr>
<th>! 2 3 4 5 6 7 8 9 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>!-!-!!!!!!-!-!-!!!!!!-!-!-!!!!!!-!-!-!!!!!!-!-!-!!!!!!</td>
</tr>
<tr>
<td>1 # SUB## H</td>
</tr>
<tr>
<td>2 a A ^</td>
</tr>
<tr>
<td>2 b A ^</td>
</tr>
<tr>
<td>2 c A ^</td>
</tr>
<tr>
<td>2 d A ^</td>
</tr>
<tr>
<td>2 e A ^</td>
</tr>
<tr>
<td>2 f A ^</td>
</tr>
<tr>
<td>2 g A ^</td>
</tr>
<tr>
<td>2 k A ^</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>1 # AUT## # -69</td>
</tr>
<tr>
<td>2 # A ^</td>
</tr>
<tr>
<td>1 # TIT## # -69</td>
</tr>
<tr>
<td>2 v ^v.^</td>
</tr>
<tr>
<td>2 # A ^</td>
</tr>
<tr>
<td>1 # 020## D</td>
</tr>
<tr>
<td>2 # A ^</td>
</tr>
<tr>
<td>1 # 020## 3</td>
</tr>
<tr>
<td>2 a ( )</td>
</tr>
</tbody>
</table>

**Note**: If the **856** entry appears without subfields in `edit_field.<lng>` (i.e. only the following line appears:

1 # 856## D

the display will be according to the following algorithm:

- Use subfield z, or if missing
- use subfield u, or if missing
- compose by using subfields a, p, d and f as done for URL creation

**edit_paragraph.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Paragraphs for displaying bibliographic information
The edit_paragraph table is used for formatting bibliographic record information for display and printing. It defines a concatenation of a number of fields, including "fields" that are included through expand_doc procedures. It is used in conjunction with edit_field and edit_doc in the BIB library, and the bib_format table in the ADM library.

Text can be added to the displayed record fields using col.4 (prefix) and col.5 (suffix). In addition, if you enter DISP in place of a field tag in col.2, you then use cols. 4 and 5 to enter a textual phrase. The DISP "tag" can be repeated as many times as required.

Within a paragraph, a "new line" can be forced between fields by using ## in cols. 4 or 5 (prefix or suffix). When edit_doc is used for formatting bibliographic record information, one format group can include any number of paragraphs. For edit_doc, it is preferable to define a separate paragraph for each new line.

Structure of the table:

| Col.1 | paragraph identifier |
| Col.2 | tag + indicators; wild card may be used |
| DISP may be used in order to display information (as defined in the prefix), without linkage to a tag |
| Col.3 | edit_field.<lng> identifier (as defined in Col. 4 of edit_field table.) |
| Col.4 | prefix |
| Col.5 | suffix |
| Col.6 | A - all occurrences of the tag |
| blank - first occurrence only |

Example of the table:

```
!1 2 3 4 5 6
!!!-!!!!!!!-!-!!!!!!!!!!!!-!!!!!!!!!!-!
000 DISP Missing
000 DISP Paragraph
!* system number
001 SYS## D [ ]
!* Author+Title
002 1#### D .^ A
002 245## D
!* Title
003 245## D
!* Note 5##
004 DISP ##
004 5#### D ^^^ ## A
```

expand_doc_bib_z30

Location of the table: tab directory of the library
Purpose of the table: Item fields expanded into the bibliographic record.

This table defines which fields from the item record (Z30) are expanded into the bibliographic record. The table is used together with the expand_doc_bib_z30 program.

The program creates a new virtual field - Z30-1 (for copy items) or Z30-2 (for issue items) - that contains the item's information.

Structure of the table:

Col. 1 Type of Item:
- Y - Issue items
- N - Copy items
- # - All types of items

Col. 2 Item Field Name. Note that active-library may also be added.

Col. 3 Sub-field into which the item field is expanded

Col. 4 Space, zero or blank
- Space - If field is empty the sub-field is not added to the expanded field.
- Zero - If the field has zeroes the sub-field is not added to the expanded field (used for date fields).
- Blank - Sub-field is always taken

Col. 5 Translation Flag
Defines if the code of the item field is to be translated to a name (e.g. sub-library code to sub-library name). Note that this is available for:
- active-library
- z30-sub-library
- z30-collection
- z30-item-status
- z30-item-process-status.

Values are:
- Y - Translate
- N - Do not translate
- X - Not applicable

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!-!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># active-library</td>
<td>1 space</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td># active-library</td>
<td>L space</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-material</td>
<td>m space</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-sub-library</td>
<td>1 space</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-sub-library</td>
<td>A space</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-collection</td>
<td>2 space</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-collection</td>
<td>B space</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-call-no</td>
<td>3 space</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-call-no-2</td>
<td>9 space</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td># z30-inventory-number</td>
<td>6 space</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
fix_doc.<lng>

Location of the table: pc_tab/catalog directory of the library

Purpose of the table: Fix_doc definitions

Defines the fix_doc routines that are included in the Cataloging module under the Edit menu/Fix record and the Edit menu/Derive record options.

Structure of the table:
Col. 1 Procedure identifier (as defined in Col. 1 in tab_fix)
col. 2 Derive or Fix:
   Defines if a new record is going to be created when performing a fix routine, or if the current record is going to be fixed.
   - Y=Open as new document
   - N=fix current document.
   - C = Conditional Open as a new record. Only active when the record has a system number (i.e. is not NEW).
col. 3 ALPHA – should always be L
col. 4 Text that displays in the window when user invokes ‘fix record’ or the Derive record option from the edit menu in the cataloging client.

Note that in the on-line Cataloging module:
- lines with "N" will appear in the Edit Menu under Fix record
- lines with "Y" will appear in the Edit Menu under Derive new record.

Derive new record options:
Create an analytic entry from an existing document.
The fix creates a template with an empty 245 field; copies the 260, 300, 050 008 and LDR fields from the parent record; creates an LKR record with ANA link and $$n=245 of parent record.

The record is created for MARC21 records as follows:
- LDR: with position 8 as b and not the default which is a.
- 008: default as a book. Note: it would be correct to update the year in this field from the year in the LKR. The language should be the same as the 008 of the parent record.
- 050: If the fields exist in the parent record they are copied into the derived record.
- 080: If the fields exist in the parent record they are copied into the derived record.
- 245: will be opened but will contain no text.
- 260: will be taken from the parent record
- 300: $$c will be taken from the parent record
- LKR: points to parent record with sub-fields as follows:
  - $$a ANA
Create an authority record in the AUT library using the same field 100, and creating a 670 built out of 245 $a$ and 260 $c$.

Example:

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>INS N L Perform fixes as executed when document is updated (INS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04-01 N L Convert UNIMARC Records to USMARC Records</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008 N L Update 008 field from 260 field</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANA C L Create a new Analytic record</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUT Y L Create a new record in the Authority DB, based on 1XX, 7XX, 6XX field</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INS2 N L INS2 procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST N L Testing various fixes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OCLC N L OCLC fix routines</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**form_description.**<lng>

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** Defining a subject for a form sent by e-mail

**Structure of the table:**
- Col. 1 Form name
- Col. 2 Form format. If left blank the entry will apply for all formats.
- Col. 3 Description

**Example of the table:**

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>opac-print</td>
<td>An Opac printout sent to you</td>
<td></td>
</tr>
</tbody>
</table>

**form_print_method**

**Location of the table:** tab directory of the library

**Purpose of the table:** Print method for different printouts

This table defines the print methods for XML based printouts.
The default method is XML_XSL. Therefore, there is no need to define printouts that use XML_XSL in this table.

If you want to use an external tool (not XML_XSL), it should be defined in Col. 3 in the following way:
EXECUTE <COMMAND>
This assumes that there is an executable that accepts XML as input and produces the requested output.

If you want to use a server-side external tool (not XML_XSL), it should be defined in Col. 3 in the following way:

\texttt{SERVER-EXECUTE <DIRECTORY>}

This assumes that you will have the XML file under /print in <DIRECTORY> directory. The execution is not under ALEPH's responsibility.

Col. 4 determines whether to translate this XML.
Col. 5 determines whether to notify the user with a message about the XML file created. The translate option can also be used for the XML_XSL method.

For example:
To use \texttt{iexplore.exe}, assign it a symbolic name, for example \texttt{IE}.
Insert in Col.3: EXECUTE IE

In addition, the following line should be added to the al500/alephcom/tab/alephcom.ini file on the client:

\begin{verbatim}
[PrintExecute]
IE=C:\Program Files\Plus!\Microsoft Internet\iexplore.exe
\end{verbatim}

The XML will then be handled by Iexplorer.

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Form name</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Form format</td>
</tr>
<tr>
<td></td>
<td>If left blank the entry will apply for all formats</td>
</tr>
<tr>
<td>col. 3</td>
<td>Print method</td>
</tr>
<tr>
<td></td>
<td>EXECUTE &lt;COMMAND&gt; - Use external tool based on alephcom.ini definition for &lt;COMMAND&gt;</td>
</tr>
<tr>
<td></td>
<td>SERVER-EXECUTE &lt;DIRECTORY&gt; - Create the XML file on the server under /print/&lt;DIRECTORY&gt;</td>
</tr>
<tr>
<td>col. 4</td>
<td>translate:</td>
</tr>
<tr>
<td></td>
<td>Y=XML file will be translated</td>
</tr>
<tr>
<td></td>
<td>N=XML file will not be translated</td>
</tr>
<tr>
<td>col. 5</td>
<td>user notification</td>
</tr>
<tr>
<td></td>
<td>Y=User will be notified about the new file created</td>
</tr>
<tr>
<td></td>
<td>N=User will not be notified about the new file created</td>
</tr>
</tbody>
</table>

Example of the table:
form_sub_library_address

Location of the table: tab directory of the ADM library

Purpose of the table: Address types for forms

This table defines address types for Acquisition, ILL & Circulation forms. The table is used to link between the print form and the particular address for those forms that include sub_library address (taken from the alephe/tab_sub_library_address table).

The address type in column 2 is matched against the address type in column 2 of the alephe/tab_sub_library_address table. If no match is found, address type 1 acts as default.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Form template name</th>
<th>Col. 2</th>
<th>Sub-library address type</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>transfer-slip</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>serial-rout-list</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>serial-item-label</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>serial-claim-letter-01</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>return-receipt</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>photo-request-wait</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>photo-request-slip</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example of the table:

formats.<lng>

Location of the table: pc_tab/catalog directory of the library

Purpose of the table: List of formats for Cataloging

This table defines the codes (2 characters) of the valid formats which can be entered in the FMT field of the document record.

Structure of the table:
Col. 1 Code
Col. 2 Character set
Col. 3 Format description text

Example of the table:

- BK L Books
- CF L Computer file
- MP L Maps
- MU L Music
- SE L Serials
- VM L Visual materials
- MX L Mixed materials

**generic_fix**

Location of the table: tab/import directory of the BIB library

Purpose of the table: Processing script for fix_doc_do_file_08 and p-file-08

This is a *sample* table of a processing script for fix_doc_do_file_08 and p-file-08.

The fix_doc_do_file_08 program is a generic fix program that modifies cataloging records based on a supplied processing script. Many standard fix programs are provided by ALEPH, but there are times when a library would like to perform a customized fix on a record. This can be done by the fix_doc_do_file_08 program.

The processing script **must** be located in the tab/import directory of the library. The table name is user-defined. You can create multiple tables to define different fix procedures. The script is in the format of a normal ALEPH table with 9 columns.

The generic_fix table in the USM01 $data_tab/import directory is an example of a processing script. Specifications for the conversion script can be found in the “How to Set Up a Script for the Correction of Records How-to document.”

Structure of the table:

- **col. 1** Iteration. The operations are performed in order of iteration. Operations in iteration 1 are processed before those in iteration 2. Within iteration 1, the operations are processed in the order listed in the script.
- **col. 2** Document record field code. # can be used to indicate truncation (e.g., 245## for 2451, 2452, 24501).
- **col. 3** Record's format code (FMT field). # can be used as a wildcard. If left blank, no filtering for record format is performed.
- **col. 4** First position filter. If the position filter is not blank, then the operation is performed only if the first position in the field matches.
- **col. 5** Position range start. Used for operations on fixed fields to specify
the position range. Note that field positions are counted from zero.

col. 6 Position range end. Used for operations on fixed fields to specify the position range. Note that field positions are counted started from zero.

col. 7 Occurrence filter. Contains a five-digit number, or

- FIRST,
- LAST,
- NOT-F (not first), or
- NOT-L (not last).

col. 8 Operation code. Following are the valid operation codes:

- ADD-FIELD
- CHANGE-FIELD
- CHANGE-FIRST-IND
- CHANGE-FIRST-IND-MATCH
- CHANGE-SECOND-IND
- CHANGE-SECOND-IND-MATCH
- CHANGE-SUBFIELD
- COND-LOAD-VAL-POS
- COPY-FIELD
- COPY-SYSTEM-NUMBER
- DELETE-FIELD
- DELETE-FIELD-COND
- DELETE-FIXED-COND
- DELETE-SUBFIELD
- DELETE-SUBFIELD-DELMITER
- EDIT-SUBFIELD-HYPHEN
- FIXED-CHANGE-VAL
- FIXED-CHANGE-VAL-RANGE
- FIXED-FIELD-EXTEND
- FIXED-RANGE-OP
- REPLACE-STRING
- SORT-FIELDS

col. 9 Operation parameters

Example of the table:

```
! 2 3 4 5 6 7 8 9
!-!!!!-!!-!!-!!!-!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!…!!!
1 001 COPY-SYSTEM-NUMBER 035 ,L,a
1 035## REPLACE-STRING ocm, (OCoLC)
1 LDR ADD-FIELD OWN ,L,$$LIN
2 852## CHANGE-FIELD 949
3 949## DELETE-SUBFIELD c
3 949## REPLACE-STRING $$i,^h
3 949## CHANGE-SUBFIELD h c
3 949## CHANGE-SUBFIELD b h
```
job_list

Location of the table: alephe/tab directory

Purpose of the table: Job daemon scheduling of programs and procedures

Note that this list can define either programs or procedures.
- If programs, col. 5 lists the program commands.
- If procedures, col. 5 lists the library code, and cols. 6 and 7 the name parameters of the procedure.

Column definition for programs
Col. 1 day
Col. 2 hour
Col. 3 queue (y/n)
Col. 4 log name OR target name (see following)
Col. 5 program to run OR library code (see following)

Column definitions for procedures
Cols. 1, 2, 3 as above
Col. 4 target
Col. 5 library
Col. 6 procedure name
Col. 7 parameters

Formatting of parameters/log file name

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%D</td>
<td>day (00-06) [Sunday=00 Monday=01 ... Saturday=06]</td>
</tr>
<tr>
<td>%H</td>
<td>hour (00-23)</td>
</tr>
<tr>
<td>%M</td>
<td>minutes (00-59)</td>
</tr>
</tbody>
</table>

xxxx.print_%D_%H_%M -> if run at 09:00 on Sunday xxxx.print_00_09_00

%DATE[+/-][D/W/M]nnn - create date relative to the current date
+ - after current date - - before current date
D - days W - weeks M - month

xxxx.%DATE+D007 -> set parameter to 7 days from now
yyyy.%DATE+M001 -> set parameter to 1 month from now

Note that macros, or templates can be used based on definitions in job_list.conf:

W1 22:00:00 Y USM50 p_cir_01 USM50
02 D1 Y USM50 p_cir_01 USM50

After editing this file restart jobd (UTIL E/15/9).
Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>W2</td>
<td>23:00:00</td>
<td>N</td>
<td>/aleph/daily_backup/cron_backup_to_tape &gt;&gt; /aleph/daily_backup/backup_log.dat</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>19:30:00</td>
<td>N</td>
<td>disk_size_0</td>
<td>df -k</td>
</tr>
<tr>
<td>00</td>
<td>23:00:00</td>
<td>Y</td>
<td>USM01</td>
<td>util_a_10_b</td>
</tr>
<tr>
<td>00</td>
<td>23:00:00</td>
<td>Y</td>
<td>USM01</td>
<td>util_a_11_b</td>
</tr>
</tbody>
</table>

`job_list.conf`

Location of the table: alephe/tab directory

Purpose of the table: Job_list definitions

This table for the definition of templates or macros for daily or weekly slots that can then be defined in the job_list.

Structure of the table:
Weekly procedure slots:
- Col. 1 Code
- Col. 2 W(EEKLY)
- Col. 3 Flags (Sun - Sat) Y/N

Daily procedure slots:
- Col. 1 Code
- Col. 2 D(aily)
- Col. 3 Start time
- Col. 4 End time
- Col. 5 Interval

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>W</td>
<td>YYYYYN</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>D</td>
<td>09:00 17:00 01:00</td>
<td></td>
</tr>
</tbody>
</table>
**ldap.conf**

Location of the table: tab directory of the **USR** library

Purpose of the table: LDAP configuration file

The LDAP authentification script is activated using check_bor_id_02. In other words, only if the verification type is "02" (z308-verification-type) then the LDAP script becomes relevant.

Required setup:
1) defining the script location in aleph_start:  setenv aleph_authen_dir $alephm_source/check
2) defining LDAP server name/port/search-parameters - construct a configuration file under the name $usr00_dev/usr00/tab/ldap.conf.

The general workflow is as follows:
- the patron's z308 must exist in the database with verification type 02. The verification field itself is irrelevant as we compare the password with LDAP, not with the local verification so this field can be empty or equal the z308-key-data.
- Once a patron attempts to login, the system identifies the verification is an LDAP one (seeing "02" in z308-verification-type) and activates the LDAP script (from the location defined in aleph_start as "aleph_authen_dir").
- The script returns "Y" or "N" and from there on, the patron is authenticated as in regular authentication.

Example of the table:
```plaintext
[general]
  host_name = exchange01
  port      = 389

  search_base = o=ExLibris
  search_filter = (uid=USERNAME)

[xml setting]
  xml_root_node = bor_authentication
```

Note that the reserved word "USERNAME" will be replaced with the patron login-name before the search takes place.

**library_relation**

Location of the table: alephe/tab directory

Purpose of the table: Relation between libraries by library type
This table defines links between libraries. The system requires that the relationship between the BIB/ADM/HOL and ILL libraries be defined. Relation to AUT library is not required except for the following cases:

- ERR, which is relevant only for applications that use p-manage-31
- AUT, which is relevant only for fix_doc_new_aut_n programs.

**Relation types:**

**ADM**

BIB library /ADM library (libraries)

Defines which ADM libraries are related to the BIB library in the first column. Only one line can be used to define this relationship. It is possible to list up to 60 ADM libraries. However, Ex Libris does not recommend linking more than 20 ADM libraries to one BIB library, if the ADM libraries are using shared bibliographic records.

**HOL**

BIB/ADM libraries / HOL library (libraries)

Defines the HOL library that is related to the BIB library or the ADM library listed in the first column.

All HOL records are actually linked only to BIB records. An item record can be linked to a HOL record through Z30-HOL-DOC-NUMBER-X.

For sites that have multiple ADM libraries, the setup can be a single HOL library, irrespective of the number of ADM libraries related to the BIB library, or multiple HOL libraries, with one HOL library for each ADM library.

An advantage of multiple HOL libraries is simpler authorization control for the HOL records. Both setups function equally well in ALEPH. Note that the setup is different for single-HOL-multi-ADM, and multi-HOL-multi-ADM. An explanation follows:

If the setup is a single HOL library, there is no need to define ADM-HOL relation. The relation occurs from ADM to BIB to HOL, and from HOL to BIB to ADM. A sample setup is:

- ADM USM01 USM50 USM51
- HOL USM01 USM60
- BIB USM50 USM01
- BIB USM51 USM01
- BIB USM60 USM01

If the setup is a HOL library for each ADM (in which case each ADM MUST have its own HOL), all relations must be defined, for ADM-HOL, ADM-BIB, HOL-BIB, BIB-HOL, HOL-ADM. A sample setup is:

- ADM USM60 USM50
- ADM USM61 USM51
- HOL USM01 USM60 USM61
- HOL USM50 USM60
- HOL USM51 USM61
- BIB USM50 USM01
NOTE: if there are multiple HOL libraries that are linked to items in a single ADM library, the setup is the same as single HOL, and there is no need to define a HOL-ADM and ADM-HOL relations.

**BIB**

HOL/ADM to BIB

defines which BIB libraries are related to the HOL library or the ADM library in the first column.

**ILL**

ADM library / ILL library (libraries)

The second column defines the database library where bibliographic data for ILL requests is stored.

**ERR**

AUT library / AUT library

For p-manage-31 (load authority records) defines the library on which the "unacceptable" records are written.

**AUT**

BIB library/AUT library

For fix_doc_new_aut_n (e.g. fix_doc_new_aut_2) defines the default authority library for which the new derived authority record is created. Note that this entry is not mandatory. If this relation type is not defined, the authority library can be defined through the tab_fix table. If no authority library is defined in the library_relations table and in the tab_fix table, the system uses the default of XXX10 as the authority library.

**PID**

BIB library/BIB library

The PID library relation links an indexing library to an actual library for the purpose of parallel indexing. The link enables running index jobs in the indexing library in parallel to normal workflow of the actual library. The actual library is not locked during this period.

**BLK**

AUT library / AUT library

This setting is relevant only if the BIB library sets alternate AUT libraries for authority control of a single browse list, e.g. tab_aut setup is:

\[
\text{SUB 2 USM10 USM12}
\]

The above setting infers that the USM10 AUTHority record is preferred over the USM12 record.

The BLK line in this table is used by the system to define the actual preference. The outcome is that when the link in a BIB heading is changed from the unpreferred AUT
library to the preferred AUT library, the unpreferred AUT record's 008/14-16 is changed to "bbb", effectively de-activating the record for future use.

**PAS**

This relation sets the list of libraries that appears in the Access Rights window when creating privileges for an operator. Each ADM library defines the libraries its users can modify. Only the libraries in the relation will be shown in the permissions list (GUI and WEB).

For example:

```
PAS USM50 USM01 USM10 USM60
```

If you have connected to the GUI with an operator whose Z66-USER-LIBRARY is USM50, this is the list of libraries that will be displayed in the Access Rights window.

**Note** that up to 200 libraries can be defined per relation type. In addition, it is not permitted to break a relation type for a particular library into separate lines.

**Structure of the table:**
- Col. 1  Relation type
- Col. 2  First library
- Col. 3-6 Second to fifth libraries

**Example of the table:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
</tr>
<tr>
<td>ADM</td>
<td>USM01 USM50 USM51 USM52 USM53 USM54 USM55 UXP50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADM</td>
<td>USM20 USM50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADM</td>
<td>USM21 USM51</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOL</td>
<td>USM01 USM60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOL</td>
<td>USM30 USM60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIB</td>
<td>USM50 USM01 USM20 USM30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIB</td>
<td>USM51 USM01 USM21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note** that if changes are made to the table, the GUI server must be killed and reactivated in order to see the changes reflected in the online modules.

**license**

**Location of the table:** alephe/tab directory

**Purpose of the table:** License parameters

Parameters pertaining to the institution’s **AALEPH 500** license are defined here, as per the parameters issued by **Ex Libris**.
marc_country_codes

Location of the table: alephe/tab directory

Purpose of the table: Country codes that may be used in MARC records

This table contains the list of country codes that may be used in MARC records in various places. The table may be used to validate input entered into fixed fields, e.g., the 008 field.

Structure of the table:
- Col. 1 - 15
  - Country code – each column consists of up to three lower case characters

Example of the table:

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>!</td>
<td>aa</td>
<td>abc</td>
<td>ae</td>
<td>af</td>
<td>aq</td>
<td>ai</td>
<td>aj</td>
<td>aku</td>
<td>alu</td>
<td>am</td>
<td>an</td>
<td>ao</td>
<td>aq</td>
<td>aru</td>
<td>as</td>
</tr>
<tr>
<td>at</td>
<td>au</td>
<td>aw</td>
<td>ay</td>
<td>azu</td>
<td>ba</td>
<td>bb</td>
<td>bcc</td>
<td>bd</td>
<td>be</td>
<td>bf</td>
<td>bg</td>
<td>bh</td>
<td>bi</td>
<td>bl</td>
<td></td>
</tr>
<tr>
<td>bm</td>
<td>bn</td>
<td>bo</td>
<td>bp</td>
<td>br</td>
<td>bs</td>
<td>bt</td>
<td>bu</td>
<td>bv</td>
<td>bw</td>
<td>bx</td>
<td>cau</td>
<td>cb</td>
<td>cc</td>
<td>cd</td>
<td></td>
</tr>
<tr>
<td>ce</td>
<td>cf</td>
<td>cg</td>
<td>ch</td>
<td>ci</td>
<td>cj</td>
<td>ck</td>
<td>cl</td>
<td>cm</td>
<td>cou</td>
<td>cq</td>
<td>cr</td>
<td>ctu</td>
<td>cu</td>
<td>cv</td>
<td></td>
</tr>
<tr>
<td>cw</td>
<td>cx</td>
<td>cy</td>
<td>dcu</td>
<td>deu</td>
<td>dk</td>
<td>dm</td>
<td>dq</td>
<td>dr</td>
<td>ea</td>
<td>ec</td>
<td>eg</td>
<td>enk</td>
<td>er</td>
<td>es</td>
<td></td>
</tr>
<tr>
<td>et</td>
<td>fa</td>
<td>fg</td>
<td>fi</td>
<td>fj</td>
<td>fk</td>
<td>flu</td>
<td>fm</td>
<td>fp</td>
<td>fr</td>
<td>fs</td>
<td>ft</td>
<td>gau</td>
<td>gb</td>
<td>gd</td>
<td></td>
</tr>
</tbody>
</table>

marc_exp.dat

Location of the table: pc_tab/catalog directory of the library

Purpose of the table: Sub-fields for Open Form and List of fields (F5)

This table sets the sub-fields that will display when open form (ctrl-F) and when selecting a field from the List of fields (F5) is invoked in the Cataloging client. The table should include only the fields for which a form screen has NOT been created. (Form screens are defined and located in the directory pc_tab/catalog of the library). Note that not ALL the sub-fields need be defined - only those most in use.

Structure of the table:
- Col. 1 Tag
- Col. 2 Indicators. Use wild cards (#) to indicate any indicator
- Col. 3 Record format as registered in the FMT field in the document record. Use XX for all formats.
- Col. 4 sub-fields

Example of the table:
marc_language_codes

Location of the table: alephe/tab directory

Purpose of the table: Language codes that may be used in MARC records

This table contains the list of language codes that may be used in MARC records in various places. The table may be used to validate input entered into fixed fields, e.g., the 008 field.

Structure of the table:

Col. 1 - 15
language code – each column consists of up to three lower case characters

Example of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>ace</td>
<td>ach</td>
<td>ada</td>
<td>ady</td>
<td>afa</td>
<td>afh</td>
<td>afr</td>
<td>aka</td>
<td>akk</td>
<td>alb</td>
<td>ale</td>
<td>alg</td>
<td>amc</td>
<td>amh</td>
<td></td>
</tr>
<tr>
<td>angi</td>
<td>qua</td>
<td>ara</td>
<td>arc</td>
<td>arg</td>
<td>arm</td>
<td>arn</td>
<td>arg</td>
<td>art</td>
<td>arw</td>
<td>asm</td>
<td>ast</td>
<td>ath</td>
<td>ava</td>
<td>ave</td>
</tr>
<tr>
<td>awa</td>
<td>aym</td>
<td>azi</td>
<td>baq</td>
<td>bal</td>
<td>bai</td>
<td>bak</td>
<td>bal</td>
<td>bam</td>
<td>ban</td>
<td>baq</td>
<td>bas</td>
<td>bat</td>
<td>bej</td>
<td></td>
</tr>
<tr>
<td>bel</td>
<td>bem</td>
<td>ben</td>
<td>ber</td>
<td>bho</td>
<td>bik</td>
<td>bin</td>
<td>bis</td>
<td>bia</td>
<td>bnt</td>
<td>bra</td>
<td>bre</td>
<td>btk</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

marc8_xxx_to_unicode

Location of the table: alephe/unicode directory

Purpose of the table: Character conversion tables

This is a generic description related to a number of tables that convert characters from the MARC8 character set to Unicode. The following tables have been defined:

- marc8_ara_to_unicode
- marc8_eacc_to_unicode
- marc8_ext_ara_to_unicode
- marc8_ext_rus_to_unicode
When converting from UTF to MARC8 the programs will take the first occurrence in the second column that matches the Unicode input character. It is possible to rearrange the tables so that the line with the value desired for export precedes other lines for the same Unicode character. Marc8 values don't have to be sorted.

Structure of the table:

- Col. 1  Hexadecimal value of the Unicode character
- Col. 2  Hexadecimal value of the MARC-8 character
- Col. 3  Comment from the source table

Example of the marc8_greek_to_unicode table:

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>037E</td>
<td>3F</td>
<td>GREEK QUESTION MARK</td>
</tr>
<tr>
<td>0391</td>
<td>41</td>
<td>GREEK CAPITAL LETTER ALPHA</td>
</tr>
<tr>
<td>0392</td>
<td>42</td>
<td>GREEK CAPITAL LETTER BETA</td>
</tr>
<tr>
<td>0393</td>
<td>44</td>
<td>GREEK CAPITAL LETTER GAMMA</td>
</tr>
<tr>
<td>0394</td>
<td>45</td>
<td>GREEK CAPITAL LETTER DELTA</td>
</tr>
<tr>
<td>0395</td>
<td>46</td>
<td>GREEK CAPITAL LETTER EPSILON</td>
</tr>
<tr>
<td>03DA</td>
<td>47</td>
<td>GREEK LETTER STIGMA</td>
</tr>
<tr>
<td>03DC</td>
<td>48</td>
<td>GREEK LETTER DIGAMMA</td>
</tr>
<tr>
<td>0396</td>
<td>49</td>
<td>GREEK CAPITAL LETTER ZETA</td>
</tr>
<tr>
<td>0397</td>
<td>4A</td>
<td>GREEK CAPITAL LETTER ETA</td>
</tr>
<tr>
<td>0398</td>
<td>4B</td>
<td>GREEK CAPITAL LETTER THETA</td>
</tr>
<tr>
<td>0399</td>
<td>4C</td>
<td>GREEK CAPITAL LETTER IOTA</td>
</tr>
<tr>
<td>039A</td>
<td>4D</td>
<td>GREEK CAPITAL LETTER KAPPA</td>
</tr>
<tr>
<td>039B</td>
<td>4E</td>
<td>GREEK CAPITAL LETTER LAMDA</td>
</tr>
</tbody>
</table>

**path_convert**

Location of the table: tab directory of the library

Purpose of the table: Re-direction of directories for sharing between libraries

The path_convert table serves to re-direct directories and/or files and configuration tables from one path to another. This allows sharing setup tables between libraries.

The left column lists the path where the system expects to find the directory or file, and the right column lists the path that should be used instead.

Structure of the table:

- Col. 1  Actual path
- Col. 2  Redirect path

Example of the table:
Note that path_convert under UTIL Y/6 has an important sub-menu option:

4. Check path convert of library file name

This enables the system to check path convert for files in a specific library or under Salephe_root.

**pc_server_defaults**

Location of the table: alephe_root directory

Purpose of the table: Default values for GUI applications

This file defines various default values for working with the GUI environment:

**Note:** If a change is made in this file, the server needs to be killed (UTIL W/2/4) and reactivated (UTIL W/3/3) in order to be able to see the changes online.

General note:
Note that there are some definitions that offer more than one “setenv” option. The non-active option does not have to be commented out – the system will always use the last line defined for that option.

**Explanation of some of the elements in the file:**

```plaintext
setenv PC_NUM_SERVERS 5
This determines how many backend server are running

setenv acq_user_z71_sort_routine 00
setenv acq_user_z71_sort_order A
These parameters define the sort routine and the sort order for the Acquisitions Logger in the online client

Sort routines:
- 00-by the open date & hour of the transaction (the most common used sort)
- 01-by sequence no. of the log (mostly for conversions)
- 02-by sequence no. of the log, the logs with the Z71-ACTION-DATE will always be sorted on to by Z71-ACTION-DATE.

Sorting order:
- A – ascending
- D- descending
```
setenv borname_style
Setenv borname_style_u 02
setenv borname_style_s 01
setenv borname_style_f 03
Defines the username style for:
- U – user input
- S – Screen display
- F – Form
The numbers denote format
- 01 - name, title
- 02 - name
- 03 - title name
- default if not defined – name (02)

setenv bor_list_address 02,03
Display of patrons in CIRC – defines which address lines will display.
- 02,03 will display patron name together with the second and third lines of the address (up to five lines can be defined – 02,03,04,05.) The first line is the patron name, copied automatically from the name field.
- 00 - name only with no address elements.

setenv call_no_support N
This line defines whether the system is using call numbers instead of barcodes for item identification for Circulation. Libraries that use this setup must have a unique Z30_CALL_NO value for each item, and must have an index on this field.

setenv circ_cataloger_request_date 001
Defines the number of days after the creation of a quick catalog record in CIRC that a hold request for patron CATALOGER will be created.

setenv circ_user_z36_sort_routine 01
setenv circ_user_z36_sort_order A
Defines the sorting of the patron’s loan list. Two routines are available:
- 00 - "standard" sort using the due-date
- 01 - sort using sub-library, item-status, collection, due-date
Sort order can be
- (A)scending or
- (D)escending
If the standard sort is sufficient then these lines can be commented out.

setenv circ_user_z37_sort_routine 01
setenv circ_user_z37_sort_order A
Defines the sort of the patron’s hold list. Three routines are available:
- 00 - "standard" sort using priority, request date, open date & hour
- 01 - sort using sublib, item-status, collection, open date & hour
- 02 - sort using sublib, item-status, collection, status, open date & hour
Sort order can be
• (A)scending or
• (D)escending
If the standard sort is sufficient then these lines can be commented out.

```bash
setenv circ_user_z38_sort_routine       01
setenv circ_user_z38_sort_order         A
```
As above, for photocopy requests

```bash
setenv create_statistics             Y
```
The z34 records can be used for statistical analysis of server use. If set to Y the counter “last-z34-sequence” should be defined in the z52 table (UTIL G/2).

```bash
setenv date_style_s 12
setenv date_style_f 12
```
Defines the date display for:
• S – Screen
• F – Form

The first digit of the number denotes the separator style. The second digit, the date style.

default: dd/mm/yy

**First digit - separator**
0 separator '‐'
1 separator '/'
2 separator ''

**Second digit – date style**
1 dmy
2 dmy w/century
3 dmy w/century, long month
4 dmy w/century, short month
5 mdy
6 mdy w/century
7 mdy w/century long month
8 mdy w/century short month

For example
date style 14 will display
25/Dec/2003
date style 25 will display
12 25 2003

**First digit – no separator**
no separator (not 0,1,2)

**Second digit – date style**
1 ddmmyy
For example
date style 34 will display
25122000

setenv default_lock_period 300
Locked acquisition, item, ILL and circulation records are automatically unlocked after the period defined in this section.
The period is defined in seconds. By default (if not otherwise defined), the variable has been set to lock records for 300 seconds

setenv default_recall_type "01"
Defines the default recall type for the Create Hold Request in the Circulation interface.
If a value is not found, the program defaults to "03".

setenv doc_lock_period 3600
Defines the period after which the documents which are locked by the cataloguing lock function will be automatically unlocked. The period is defined in seconds.

setenv due_date_format 1
This setenv controls what displays when there is just a single "Due date" column. It works in combination with switch RECALL-METHOD in tab100.

- 1 - Display a single "effective due date", based on RECALL-METHOD in tab100.
- 2 - Display the recall-due-date, if there is one (even if it’s later); otherwise, display the regular due date.
- 3 - Always display only the z36_due_date; (for recall methods 2 and 3 this would mean that the recall-due-date will never display).*
- 4 - Display both: the regular due date and (if there is one) the recall-due-date.

* Though this is an option, it is not recommended.
NOTE: value "1" is recommended. If this line is set in aleph_start (and it must be set there for batch jobs), there is no need to set this line in www_server.conf or pc_server_defaults

setenv expand_tree_style 2
How to expand catalog navigation tree and overview trees:
- 1 - Don’t Expand, only the specific record node will be visible
- 2 - Expand only record nodes (ADM,BIB and HOL)
- 3 - Expand all tree nodes (ADM,BIB,HOL,Z30,Z68,Z16 ...)

external binding settings
setenv ext_bind_process_status SB
setenv ext_perform_loan Y
setenv ext_default_return_hour 2359

setenv ext_bind_ref_no_1 LCNum,010,a
setenv ext_bind_ref_no_2 ISBN,020,a
setenv ext_bind_ref_no_3 ISSN,022,a

Up to 3 additional fields are supported for including data from BIB for external binding. Each has 3 sections:
- Text description of the tag,
- the BIB tag,
- subfield

setenv file_lock_period 1800
This variable defines the lock period of files (tables) edited through alephadm. It must be at least two minutes longer than the warning time out period defined in alephadm.ini (variable ‘WarningTimeOut’). The period is defined in seconds. The lock period for a record is defined either in aleph_start or in pc_server_defaults. By default the system will use the values listed above.

setenv hold_pickup_from_all_items Y
This variable should be defined both in pc_server defaults, and in www_server.conf It defines whether to collect the possible pickup locations also from all like copies. Only locations which exist in the requested item and all like copies will be shown in the list. Otherwise it will create the list only from the requested items.

In both methods the possible pickup locations will be taken from tab37.

Creating the list is relevant in the following instances:
- Hold request screen in the GUI CIRC
- Hold request screen in the WEB OPAC
- Checking the correction of record Z37 before writing
- check_hold_request_e - Checking hold request routine.

If not defined defaults to “N.”

setenv ill_expected_return_date 028
This line is used for defaulting the due date when an item is loaned from the ILL service to an ILL partner.

setenv ill_return_for_user 003
The due date on an ILL item will be the “expected date of return” minus the number defined in this field.
Note in relation to this: The vendor form has two delay periods, one for arrival, and one for return. If the delay for return is 0, the delay will be taken from this variable.

setenv item_hol_tree_style 3
Item-HOL connection using the field Z30-HOL-DOC-NUMBER - How to show it in catalog navigation tree and overview trees:
• 1 - Don’t show the connection between item and HOL record
• 2 - Show item under its ADM record and under the linked HOL
• 3 - Show item only under its linked HOL record and show HOL libraries before ADM libraries

setenv item_label_restore Y
When an item label is printed using the print label facility in the items module, item status, sub-library, collection and location are automatically restored to original values, as kept in the Z30 field of the ADM record.

setenv item_schedule_preview_period 00:30
Sets a preview period in order to see requests in the following time slot for short loan/advanced booking items. If an item is not currently requested and is available, the item will be loaned for the remainder of the current slot and for the next slot as well.

Note: that this environment is defined directly in $aleph_proc/sc_server for short term loans using Self Check machines.

setenv item_schedule_release_period_60 00:30
setenv item_schedule_release_period_61 00:45
setenv item_schedule_release_period_all 01:00
It is possible to define the release period for advanced booking items. The definition can be item status specific (e.g. ‘60’, ‘61’ in the above examples), or can have the definition ‘all’ to signify all other item statuses that are not explicitly defined.

setenv max_unit_price_diff_percent 1
In the Acquisitions module, when a new invoice is created, or an existing one is modified, and the actual unit price exceeds the estimated unit price by more than a defined percent, an alert message displays.
The maximum percent difference is determined by setting the value of the max_unit_price_diff_percent variable.
Note that the alert message is displayed only ONCE, and it does NOT prevent the invoice data from being saved in the database. That is, even if the alert message is discarded, and the "Line Item" window is closed via the "Cancel" button, the invoice record (Z75) is updated.

setenv modify_item_arrival Y
Defines whether the item specific check-in screen displays for serial issue check-in. If set to N, only the general check-in screen displays.

setenv overdue_delinquency_period 00
The number of overdue days after which the patron has global delinquency code 70 registered in the delinquency field of the global patron record. This happens automatically, therefore set to 00, comment out, or remove line if not required.

setenv pc_cat_hol_item_support
N - It is not possible to highlight the holdings library node in the record manager section of the GUI and choose "Record manager > Load Create record". When the
parameter is "N" it will always give an "ADM" link even when the record is created from HOL library.

Y - When the cataloger highlights the holdings library node in the record manager section of the GUI and chooses "Record manager > Load Create record", a HOL type record is created.

```
setenv pc_sort_field       "01   D02   A"
setenv pc_sort_field_aut   "02   A03   A"
setenv pc_sort_field_sub   "03   D02   A"
setenv pc_sort_field_shl   "DISPLAY"
```

Sort options – see www_server.conf defaults for more detail
Note, however, that it is possible to write special value "DISPLAY" for a specific access code. This value means that the sort will be as it was displayed in the browse list, and not based on parameters defined in tab_sort.

```
setenv PC_SERVER_TIMEOUT
If no action within the defined time (SS), the client’s link to the server times-out
```

```
setenv pc_transactions_log
• "Y" - a file pc_ser_<port> is created in $LOGDIR and transactions are written to this file.
• "N" - the file pc_ser_<port> is not created in $LOGDIR.
It is recommended in production versions to set this parameter to "N", to improve performance
```

```
setenv pc_tree_view_max_branch 10
setenv pc_filter_tree_view_max_branch 99
Defines the limit after which the display of the navigation tree in the on-line clients is truncated with an appropriate message. The range defined must be between 1 and 99. Trees are limited to maximum 800 lines.
```

```
setenv pc_z37_date_to                  M006
This line defines the default value for "to date" (date of last interest) for the Create Hold Request function in the Circulation interface. The format is <x><nnn>, where x can be D (days), M (months) or Y (years).
```

```
setenv remote_z30_lib_01        XXX50
setenv remote_z30_lib_02        YYY50
Defines the environment variables “remote_z30_lib_nn (01,02...)” for every ADM library that needs to be checked when the local ADM library fails to find an item (by barcode).
```

```
setenv security_path_p "$TMPDIR;$FILE_TMPDIR"
setenv security_path_g $TMPDIR;$FILE_TMPDIR…
Security path definitions for upload/download of files:
security_path_p – upload - put (used especially for alephadm)
security_path_g – download - get (used especially for printouts/temporary files).
```

```
setenv serial_bind_period       50
```
serial_bind_period - min no. days to bind non-arrived issues. This relates to the 'Print Completed Volumes' option to be found the under Utilities menu in the on-line Items module.
This parameter defines the maximum number of days to wait for an item after its issue date (Z30-issue-date), before considering it 'unarrived' for the purpose of this report.

```bash
setenv set_default_z110_create_order D
```
This defines ascending or descending sort of system numbers for sets which are more than the defined sort limit

```bash
setenv set_prox_limit 100
```
Defines the limit of number of records for proximity searching

```bash
setenv time_style
setenv time_style_s 04
setenv time_style_f 04
```
Time display for:
- S – Screen
- F – Form
Numbers denotes format
default: hhmm
01 hh:mm
02 hh-mm
03 hh mm
04 hh:mm AM/PM
05 hh-mm AM/PM
06 hh mm AM/PM
07 hhmm AM/PM
Date and Time formatting can be set in ALEPH for screen display and for forms. The "setting" for the user input for the GUI clients can be done on the client itself by activating the F9 key.

**Note:**
```bash
setenv pc_sort_limit
setenv set_result_set_limit
```
are no longer in use. The set is limited to 1000 results and all the results are sorted. If the query found more than 1000 records, only the first 1000 will be shown.

**pc_tab_acq_fast_cat.<lng>**

Location of the table: tab directory of the BIB and ADM libraries

Purpose of the table: Fields for cataloging in Acquisitions

This table defines fields for the quick cataloging option in the Acquisitions GUI module.
Quick cataloguing is functional in the library defined in the acq.ini file on the client: [FastCatDlg]
Library=<library code – e.g. USM01>

In the BIB library, the table should contain the bibliographic fields.

In the ADM library, the table should contain the administrative fields that will be included in the ADM record.

Note: The total number of lines for the BIB and ADM tables must not exceed 10

Note: In the ADM library, if no fields are necessary, the table should be left empty but it MUST be present.

Structure of the table:

Col. 1 indicates whether the field is
Y = the field is optional.
N = the field is mandatory
H – the field is hidden, but its data is written in the record. The field and the text in col. 4 are always written on the record;

Col. 2 field code.
Col. 3 sub-field.
Col. 4 Field name or text of hidden field

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
</tr>
<tr>
<td>Y</td>
<td>100</td>
<td>a</td>
<td>Author</td>
</tr>
<tr>
<td>N</td>
<td>245</td>
<td>a</td>
<td>Title (mandatory)</td>
</tr>
<tr>
<td>Y</td>
<td>260</td>
<td>a</td>
<td>Place</td>
</tr>
<tr>
<td>Y</td>
<td>260</td>
<td>b</td>
<td>Publisher</td>
</tr>
<tr>
<td>Y</td>
<td>260</td>
<td>c</td>
<td>Date</td>
</tr>
<tr>
<td>Y</td>
<td>020</td>
<td>a</td>
<td>ISBN</td>
</tr>
<tr>
<td>Y</td>
<td>245</td>
<td>b</td>
<td>Additional name</td>
</tr>
<tr>
<td>Y</td>
<td>245</td>
<td>d</td>
<td>Dates</td>
</tr>
<tr>
<td>Y</td>
<td>245</td>
<td>c</td>
<td>Optional name</td>
</tr>
</tbody>
</table>

pc_tab_cat_conv

Location of the table: tab directory of the BIB library

Purpose of the table: Remote conversion programs

The standard conversion programs, performed when the Import Records function of the Cataloging module is used, are usually based on a program that is executed from the client. In addition it is also possible to convert a file of selected records by sending it to the server - where the conversion is performed - instead of converting the records at the level of the PC.

This table is used to define the remote conversion programs
For remote conversions, note that the conversion line in under the [ConvertFile] section of the catalog.ini file must be defined as explained below.

ConvertN=TextALPHA,Text,ProgramPath,[Parameter1,Parameter2,ParameterN]

For server conversions:
- ProgramPath: Must be REMOTE
- Parameter1 : Must be a routine that matches column 1 of this table.

Note that following routines can be used for the conversion of CDMARC and MAB2-Diskettenformat records:
- MAB pc_cat_conv_mab_d 850_TO_UTF,MAB
- CDMARC pc_cat_conv_cdmarc 8859_1_TO_UTF

Structure of the table:
- Col. 1 Routine name. This is the identifier of the conversion that is being performed (free-text). It matches Parameter1 in the conversion lines for remote conversions in the catalog.ini file
- Col. 2 Program name - Enter the conversion program that should be performed for the specific conversion routine defined in column 1
- Col. 3 Program arguments - Certain conversion routines require additional information, such as character conversion routines. This column is used to define additional parameters for conversion programs.

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAB</td>
<td>pc_cat_conv_mab_d</td>
<td>850_TO_UTF,MAB</td>
</tr>
<tr>
<td>!CDMARC</td>
<td>pc_cat_conv_cdmarc</td>
<td>8859_1_TO_UTF</td>
</tr>
<tr>
<td>CDMARC</td>
<td>pc_cat_conv_cdmarc</td>
<td></td>
</tr>
<tr>
<td>MARC</td>
<td>pc_cat_conv_marc</td>
<td></td>
</tr>
<tr>
<td>MARC_FRNCH</td>
<td>pc_cat_conv_marc</td>
<td>DIA_TO_UNICODE2</td>
</tr>
<tr>
<td>SEQ</td>
<td>pc_cat_conv_aleph_seq</td>
<td></td>
</tr>
<tr>
<td>SEQ_8859_1</td>
<td>pc_cat_conv_aleph_seq</td>
<td>8859_1_TO_UTF</td>
</tr>
<tr>
<td>SEQ300</td>
<td>pc_cat_conv_aleph300_seq</td>
<td>ALEPH300_TO_UTF</td>
</tr>
</tbody>
</table>

pc_tab_circ_fast_cat.<lng>

Location of the table: tab directory of the BIB and ADM libraries

Purpose of the table: Fields for cataloging in Circulation

This table defines fields for the quick cataloging option in the Circulation GUI module.
Quick cataloguing is functional in the library defined in the circ.ini file on the client: [FastCatalog]
Library=<library code – e.g. USM01>
The tables is should appear on both the ADM and the BIB libraries.

In the BIB library, the table should contain the bibliographic fields.

In the ADM library, the table should contain the administrative fields that will be included in the ADM record.

Note: The total number of lines for the BIB and ADM tables must not exceed 10
Note: In the ADM library, if no fields are necessary, the table should be left empty but it MUST be present.

Structure of the table:
  Col. 1 indicates whether the field is
  Y = the field is optional.
  N = the field is mandatory
  H – the field is hidden, but its data is written in the record. The field and the text in col. 4 are always written on the record;
  Col. 2 field code.
  Col. 3 sub-field.
  Col. 4 Field name or text of hidden field

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>-</td>
<td>-</td>
<td>!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
</tr>
<tr>
<td>Y</td>
<td>100</td>
<td>a</td>
<td>Author</td>
</tr>
<tr>
<td>N</td>
<td>245</td>
<td>a</td>
<td>Title (mandatory)</td>
</tr>
<tr>
<td>Y</td>
<td>260</td>
<td>a</td>
<td>Place</td>
</tr>
<tr>
<td>Y</td>
<td>260</td>
<td>c</td>
<td>Date</td>
</tr>
<tr>
<td>Y</td>
<td>020</td>
<td>c</td>
<td>ISBN</td>
</tr>
<tr>
<td>Y</td>
<td>022</td>
<td>c</td>
<td>ISSN</td>
</tr>
<tr>
<td>H</td>
<td>500</td>
<td>a</td>
<td>Provisional Circ record</td>
</tr>
</tbody>
</table>

Note: If an application does not want to “dirty” its database, non-standard fields can be used (e.g. AU, TL, etc.), which can then be indexed separately.
When the item is created (after registering the BIB information) a hold is automatically placed for a patron CATALOGER (which therefore must be defined in the library’s patron file).
In pc_server_defaults the parameter:
  setenv circ_cataloger_request_date
defines the number of days after the creation of a quick catalog record in CIRC that a hold request for user CATALOGER will be created.

pc_tab_col.<lng>

Location of the table: tab directory of the BIB library

Purpose of the table: Definition of columns in GUI screens
This table defines the columns of information that will display in List windows in the GUI clients. A unique identifier defines every display list (e.g. PC_ITEM_ITEM) and each column is assigned a column number. These two elements must **NOT** be changed.

The display order of columns is set by the order of the lines within the display list. The columns of display are defined in percentages. The percentages of the columns to be displayed should not exceed 100.

Set the percentage column to 000 for non-display of a column.

Note that SOME list window columns are **NOT** controlled by this table. They are controlled by the /alephcom/tab/\lng /tab_col table on the GUI client.

Example of the table in /alephcom/tab/\lng

<table>
<thead>
<tr>
<th>ID</th>
<th>Column header</th>
<th>Col. 1</th>
<th>Col. 2</th>
<th>Col. 3</th>
<th>Col. 4</th>
<th>Col. 5</th>
<th>Col. 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT</td>
<td>#L Library</td>
<td>01</td>
<td>100</td>
<td>01</td>
<td>C04</td>
<td>100</td>
<td>C04</td>
</tr>
<tr>
<td>LOCATE_LIST</td>
<td>#L Database</td>
<td>01</td>
<td>070</td>
<td>01</td>
<td>C04</td>
<td>070</td>
<td>C04</td>
</tr>
<tr>
<td>LOCATE_LIST</td>
<td>#L No Records</td>
<td>02</td>
<td>030</td>
<td>01</td>
<td>C04</td>
<td>030</td>
<td>C04</td>
</tr>
<tr>
<td>ACQ_BUDGET_YEAR</td>
<td>#L Year to Invoice</td>
<td>01</td>
<td>100</td>
<td>01</td>
<td>C04</td>
<td>100</td>
<td>C04</td>
</tr>
<tr>
<td>CAT_EMPTY_MESSAGE</td>
<td>#L No Information</td>
<td>01</td>
<td>100</td>
<td>01</td>
<td>C01</td>
<td>100</td>
<td>C01</td>
</tr>
<tr>
<td>CAT_CATALOGER_LIST</td>
<td>#L Cataloger</td>
<td>01</td>
<td>030</td>
<td>01</td>
<td>C04</td>
<td>030</td>
<td>C04</td>
</tr>
<tr>
<td>CAT_CATALOGER_LIST</td>
<td>#L Level</td>
<td>02</td>
<td>020</td>
<td>01</td>
<td>C04</td>
<td>020</td>
<td>C04</td>
</tr>
<tr>
<td>CAT_CATALOGER_LIST</td>
<td>#L Date</td>
<td>03</td>
<td>025</td>
<td>01</td>
<td>C04</td>
<td>025</td>
<td>C04</td>
</tr>
<tr>
<td>CAT_CATALOGER_LIST</td>
<td>#L Hour</td>
<td>04</td>
<td>025</td>
<td>01</td>
<td>C04</td>
<td>025</td>
<td>C04</td>
</tr>
</tbody>
</table>

Structure of the table:

- **Col. 1** ID of list. The first part of name usually points to the application - e.g. PC_ITEM, PC_CASH, etc.
- **Col. 2** ALPHA
- **Col. 3** column header
- **Col. 4** column number. This number is linked to a defined element in the list, and **CANNOT** be changed. For e.g. in PC_ITEM_ITEM, Collection will **ALWAYS** be 06, Barcode will **ALWAYS** be 04. In order to change the display sequence of the columns change the order of the lines.
- **Col. 5** Percentage of screen for column display. Defined columns and their percentages should total 100%.
  - Enter 000 if you do not want a column to display.
- **Col. 6** Font code. The value of the font code is set in the client's Alephcom\tab\font.ini file, in the last two characters of the "ListBox" definition. (for example, ListBox01.) The client can be forced to ignore the pc_tab_col setup, by using ListBox##. **Note:** When a bitmap is used (e.g. checkmark) the characters *it* are entered here.
- **Col. 7** Color code. The color value is defined in the client's
Alephcom\tab\alephcom.ini file, in the [TextListBox] section.

**Note:** When a bitmap is used (e.g. checkmark) the characters *it* are entered here

Col. 8  Font2 – this is for font differentiation in a list. Relevant only in instances where the line has a system differentiation, e.g. arrived/non-arrived issue.

Col. 9  Color2 - this is for color differentiation in a list. Relevant only in instances where the line has a system differentiation, e.g. arrived/non-arrived issue.

Col. 10  Explanatory note, comment

**NOTE** In the on-line GUI applications a right hand click on any headings column, lists the columns that can be displayed for that window. The user can then choose to select, or deselect column options, or to take the system default. The user setup is saved on the client in the new alephcom/tab/UserTabCol.ini file.

**Note too** that you can display the "Link" field (title hard-coded) at the bottom of the callout bubble invoked by right-clicking on a line in a GUI list.

The "Link" field contains the name of the section in pc_tab_col which represents the list (for example, PC_CASH_SUMMARY), together with various record keys and other pieces of information which are used by programs to identify a line uniquely and to process its data.

A flag in \Alephcom\Tab\Alephcom.ini (under the [TextListBox] section) determines whether or not the "Link" field will be displayed at the bottom of the callout bubble or not:  
ShowLineLinkInBubble=Y
The flag's value can be easily changed online via a check box entitled "Display link field at bottom of callout bubble"; it is located in the "General Configuration" tab (ALEPH menu/Options/Customize...).

The following is an example of a "Link" field:

```
Link        [PC_ACQ_LIST] "000001650000152003049                          010
            NNNM00005"
```

In the example above, the section in pc_tab_col is enclosed in square brackets, and the rest of the data is enclosed in double quotes. In this case, the data is composed of the following parts:

```
Z68-REC-KEY:                      00000165000015
Z68-ORDER-NUMBER:                 2003049
Z68-VENDOR-CODE:                  010
Z68-INVOICE-STATUS = "D" ? (Y/N): N
Z68-ARRIVAL-STATUS = "C" ? (Y/N): N
Z68-ORDER-STATUS = "CLS" ? (Y/N): N
Z68-ORDER-TYPE:                   M
Z68-NO-UNITS:                     00005
```
Example of the table:

```
TREE_ERROR_LIST   L  01 020 it map  error message list
TREE_ERROR_LIST   L Message  02 080 01 C01 error message list
!*   Common - Item circulation History - Hold Request List
!*   PC_COM_HOLD_H_LIST   L Patron ID  01 015 04 C04 Patron ID
PC_COM_HOLD_H_LIST   L Patron Name  02 025 03 C03 Patron Name
PC_COM_HOLD_H_LIST   L Pickup  03 000 04 C04 Pickup location
PC_COM_HOLD_H_LIST   L Req Date  04 020 05 C01 Request Date
PC_COM_HOLD_H_LIST   L End Req  05 020 06 C06 End Request Date
...  
PC_ITEM_ITEM   L Call number 2  10 000 01 C01 z30-call-no-2
PC_ITEM_ITEM   L Hol no. 11 000 01 C01 z30-call-no-2
PC_ITEM_ITEM   L Hol+ 12 000 01 C01 hol+sublib+coll
+call number
PC_ITEM_ITEM   L Sublibrary code  13 000 01 C01 z30-sub-library
PC_ITEM_ITEM   L coll. code  14 000 01 C01 z30-collection
PC_ITEM_ITEM   L item stat. code  15 000 01 C01 z30-item-status
PC_ITEM_ITEM   L Enum-a  16 000 01 C01 z30-enumeration
-a
...  
```

Note:

```
PC_SERIAL_ITEM_L   L Loan 01 003 01 C01 06 C02 Out on loan?
```

If this line is defined as a display line (item information in the SERIAL client), issues out on loan will display with an asterisk (*).

Note:

```
PC_CIR_RR_LOAN_LIST   L Due Hour 10 000 03 C03 Due Hour
PC_CIR_RR_LOAN_LIST   L RR 06 005 it map
```

In some parts of the system, items might be marked with ✓. This is defined with the terms it map in the font and color columns as can be seen in the above example.

Note:

```
PC_CIR_LOAN_LINE   L Status 04 015 01 C03 Item Status
```

Allows for the display of item status text in the lower part of the loan window.
**pc_tab_exp_field.<lng>**

Location of the table: tab directory of the **ADM** library

Purpose of the table: Field values for the GUI pull down menus

The table defines the field values for the pull down menu options in the different GUI applications.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Internal command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>ALPHA</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Description (text as will display in the pull down menu) - (up to 50 characters)</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Link to values as defined in the different tables (up to 100 characters)</td>
</tr>
</tbody>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>
| !!!!!!!!!!!!!!!!!!!!----!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!--!!
| !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!> |
| SERIAL-SORT-TYPE | L Sub-Library/Item seq. | SE |
| SERIAL-SORT-TYPE | L Year/Vol./Issue/Part Ascending | SE |
| SERIAL-SORT-TYPE | L Year/Vol./Part/Issue Ascending | SE |
| SERIAL-SORT-TYPE | L HOL no./Year/Vol./Issue/Part Ascending | SE |
| ITEM-PROCESS-STATUS | L Not in process | |
| ITEM-PROCESS-STATUS | L Order initiation | OI |
| ITEM-PROCESS-STATUS | L On order | OR |
| ITEM-PROCESS-STATUS | L In process | IP |
| ITEM-PROCESS-STATUS | L Cataloging | CT |
| CIRC-LEVEL | L Beginner | 01 |
| CIRC-LEVEL | L Intermediate | 02 |
| CIRC-LEVEL | L Supervisor | 20 |

**Note:**

In the online Acquisitions General invoice form, the default invoice type and status were hard coded to "REG" in previous versions. This has been changed and will now default to the first defined entry in the relevant section of pc_tab_exp_field.<lng>
Note:
When printing from catalog it is possible to print server defined printouts using the expand list CATALOG-PRINT in pc_tab_exp_field.

Currently there are two types of configurable printouts: FORM and TABLE
It is defined in the following manner:

FORM,
• format no,
• template file,
• width of first column,
• width of second,

example: FORM,001,doc-printout-001,010,060,,,,,
The additional commas are intended for future parameters.

TABLE
• format-no,
• rep col section,
• rep col number
• head foot filename + rep col header section

Example: 002,doc_print_table_002,01,doc_print_header_002

These printouts are flexible and can be setup with any fixed text both before and after the printout of the actual fields of the document.

The expand list also contains a line with the printout code INTERNAL. This means the printout of the catalog record as was existing in previous versions.

pc_tab_field_extended.<lng>

Location of the table: tab directory of the ADM library

Purpose of the table: Sub-library sensitive field values for the GUI pull down menus

This table contains entries that were moved from pc_tab_exp_field.<lng> in order to allow for sublibrary sensitivity.

Only the fields with the following internal strings can be used with pc_tab_exp_field_extended.<lng>:
• BOR-STATUS
• BOR-TYPE
• INVENTORY-NUMBER
• ITEM-LOCATION
• ITEM-LOCATION-2
• LOCATION-GR
• LOCATION-USIG
• LOCATION-PREFIX
• LOCATION-SOURCE
• LOCATION-SUFFIX

Structure of the table:
Col. 1 Internal command
Col. 2 Sub-library
Col. 3 Text
Col. 4 Description
Col. 5 Code - Up to 100 characters

Example of the table:

<table>
<thead>
<tr>
<th>BOR-STATUS</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>BOR-STATUS 01: L Undergrad</td>
</tr>
<tr>
<td>02</td>
<td>BOR-STATUS 02: L Graduate</td>
</tr>
<tr>
<td>03</td>
<td>BOR-STATUS 03: L Academic Staff</td>
</tr>
<tr>
<td>04</td>
<td>BOR-STATUS 04: L Visitor</td>
</tr>
<tr>
<td>05</td>
<td>BOR-STATUS 05: L Other institutions</td>
</tr>
</tbody>
</table>

`pc_tab_fast_circ`

Location of the table: tab directory of the ADM library

Purpose of the table: Parameters for fast circulation
This table defines the parameters for fast circulation. It defines which system checks should be activated, and whether the trap can be overridden.
Fast circulation will be used most often by distant branches with poor communication lines, or by book buses connecting to the main library via a modem.

Loan options
ON_HOLD checks if item is held for another patron
BOR_ITEM checks patron/item limits

Return options
OVERDUE checks if item is late
RECALL checks if item has been recalled
Structure of the table:
Col. 1 Name of the program [up to 10 characters]
Col. 2 Type
   L – Loan
   R-Return
Col. 3 Check (Y/N)
Col. 4 Overridable (Y/N)

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!!!!</td>
<td>!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON_HOLD</td>
<td>L</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BOR_ITEM</td>
<td>L</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>OVERDUE</td>
<td>R</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>RECALL</td>
<td>R</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

pc_tab_find.<lng>

Location of the table: tab directory of the ADM library

Purpose of the table: Fields for searching for a record in the ILL module

This table allows the library to define by which fields the database can be keyword searched from within the ILL module.

Structure of the table:
Col. 1 Find code
Col. 2 Find Code Text

A maximum of 8 codes can be defined.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!</td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>ADM Administrative Record</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIB Bibliographic Record</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISSN ISSN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WTI Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAU Author</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WYR Year of publication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SICI SICI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note the option of searching for an item by SICI number.
When searching by the SICI number, the system looks for all BIB records that match the ISSN part of the SICI number. For every matched BIB record, it will then look for items in the ADM library that match the filter fields (Chronology and enumeration). Finally there will be a display of all the BIB records that have matched items. If there is only one BIB record, it will highlight the matched item, from the list.

**pc_tab_scan.<lng>**

Location of the table: tab directory of the ADM library

Purpose of the table: Fields for browsing for a record in the ILL module

Structure of the table:
- Col. 1: Scan code
- Col. 2: Scan Code Text

A maximum of 8 codes can be defined.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUT</td>
<td>Author</td>
<td></td>
</tr>
<tr>
<td>TIT</td>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>WAU</td>
<td>Word from Author</td>
<td></td>
</tr>
<tr>
<td>WTI</td>
<td>Word from Title</td>
<td></td>
</tr>
<tr>
<td>SYS</td>
<td>System number</td>
<td></td>
</tr>
<tr>
<td>ISSN</td>
<td>ISSN</td>
<td></td>
</tr>
<tr>
<td>SRS</td>
<td>Series</td>
<td></td>
</tr>
</tbody>
</table>

**pc_tab_sear.<lng>**

Location of the table: tab directory of the BIB and AUT libraries

Purpose of the table: Definition of GUI SEARCH search options

The table is structured as follows:
- Col.1: SC – SCAN
- FI – FIND
- FO - DISPLAY FORMAT
- SO – SORT
- PF - PRINT/SAVE formats
- FF - FIXED FIELD - For Fixed Field Search.. If no such lines are defined for a base, the "Anywhere in the record" field will appear.
LN  Record language
Col. 2  Base code as defined in col. 1 of tab_base.<lng>
Col. 3  ALPHA code
Col. 4  Heading displayed
Col. 5  According to code defined in col. 1:
   SC  - link to the ACCess code as defined in tab00.<lng> in the tab directory of the library
   FI  - link to the WORD files in as defined in tab00.<lng> in the tab directory of the library
   FO  - link to edit_doc format (must be expressed in 3 digits).
   SO  - link to field tag codes as defined in tab_sort) table in the tab directory of the library
   PF  - link to edit_doc format for print/save formats
   FF  - For the 'fixed field' search options in the Search clients

<table>
<thead>
<tr>
<th>!</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Titles</td>
<td>TIT</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Authors</td>
<td>AUT</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Authors + Titles</td>
<td>AWT</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Series</td>
<td>SRS</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Subjects (LC)</td>
<td>SUL</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Subjects (MeSH)</td>
<td>SUM</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>USM01</td>
<td>L Subjects (all)</td>
<td>SUB</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>MUSIC</td>
<td>L Titles</td>
<td>TIT</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>MUSIC</td>
<td>L Main Entry</td>
<td>AUT</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>MUSIC</td>
<td>L Performers</td>
<td>PRF</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>MUSIC</td>
<td>L Librettists</td>
<td>LBT</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>MUSIC</td>
<td>L Anywhere in the record</td>
<td>WRD</td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>MUSIC</td>
<td>L Titles</td>
<td>WTI</td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>MUSIC</td>
<td>L Main entry</td>
<td>WAU</td>
<td></td>
</tr>
</tbody>
</table>

**pc_tab_short.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Display of the 'Brief format' of a record in the Search function

This table defines the columns displayed in the brief list format in the Search function available in the GUI modules.
The bibliographic field tag is used to define the content of a column. The tag could also be a virtual field created by an expand routine. The content can be formatted using edit_field (line ID in edit_field is 'S').

Special tags LOC+, PST and PSTS can be used in order to display up to three locations in the brief display. The locations displayed are separated from each other using site-defined line 0002 in the alephe/error_<lng>/sear_brief_edit table.

- **LOC+** - displays the LOC field (created by expand_doc_bib_loc_disp and using edit_field), and a link to relevant holdings. This is limited to three locations --- otherwise displays alephe/error_lng/sear_brief_edit: line 0001
- **PSTS** - displays the PSTS field (created by expand_doc_bib_psts_disp and using edit_field), and a link to relevant holdings. This is limited to three locations --- otherwise displays alephe/error_lng/sear_brief_edit: line 0001
- **PST** - like PSTS, for the PST field created by the expand_doc_bib_loc_[n] and expand_doc_sort_x group of programs. Refer to ALEPH User Guide/ Indexing section / Expand routines, tables and Expanded fields.

Example of the sear_brief_edit table:

```
!!!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
0001 0000 L More locations...
0002 0000 L --
```

Structure of the table:
- **Col.1 Format:**
  - "00" is used except in special cases
  - At this point all brief lists use format 00 except brief lists that originated from "Too many links" which will use format 01.
- **Col.2 Column heading**
- **Col. 3 Column number**
- **Col. 4 Field code and sub-field code.**
  - Code SUM will number the documents in the list and their relative positioning in the list
- **Col. 5 1st alternative field and sub-field**
- **Col. 6-8 alternative fields and sub-fields.**
- **Col. 9 Percentage of screen display.** The total length of the line is 100%, which is divided among the different fields displayed on the line.
- **Col. 10 Font (font number as defined on the PC in the alephcom/tab/font.ini file.)** The relevant lines in font.ini start with "ListBox…”
- **Col. 11 Color (color number as defined on the PC in the alephcom/tab/alephcom.ini file.)**
- **Col. 12 For fixed fields, starting position for text extraction.**

**Note:** The count is from base 001. When determining character position, if the fixed field begins with a sub-field code (as in UNIMARC
field 100), the calculation does NOT have to take the three positions of the sub-field into account.

For example, for:

**MARC21** field 008, date-1 is 008 (listed as 07-10 in the MARC21 manual).

**UNIMARC** field 100, date-1 is 010 (listed in the UNIMARC manual as 09-12).

**Col. 13** For fixed field, number of characters for display

**Col. 14** Language code can be entered here. It will act as a filter on the field, if the field has $9 with a language code.

Example of the table:

| ! | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|
| 14|    |    | ! | - | ! | ! | ! | - | ! | ! | ! | ! | ! |

| 00 Doc no. | 1 SUM | 010 01 C08 |
| 00 Call No | 2 050## LOC## | 015 01 C01 |
| 00 Author | 3 100##a 700##a | 025 02 C02 |
| 00 Title | 4 245## 240## | 030 03 C03 |
| 00 Year | 5 008 260##c | 010 03 C03 008 004 |
| 00 $1 | 6 BASE1 | 005 01 C01 |
| 00 $2 | 7 BASE2 | 005 |

**pc_tree_c1501**

*Location of the table: alephe/error_<lng>*

*Purpose of the table: Messages related to the Navigation Windows in the online ALEPH modules*

*Structure of the table:*

  Col. 1 Error code
  Col. 2 Error position
  Col. 3 ALPHA
  Col. 4 Error message

*Example of the table:*

| 0010 0000 L Cannot find ADM record |
| 0011 0000 L Cannot find BIB record |
| 1001 0000 L $1 - $2 |
| 1002 0000 L ITEMS |
| 1003 0000 L ORDERS |
| 1004 0000 L Barcode - $1 |
| 1005 0000 L Order number - $1 |
permission.dat

Location of the table: pc_tab/catalog directory of the library

Purpose of the table: Allowed and denied tags for users in cataloging

This table defines which tags can/cannot be edited and by whom. The # sign can be used for any portion of the tag-indicator combination.

**NOTE** that each user that has cataloging authorization must also be listed here in order to be able to catalog.

A library that does NOT want this control should delete this table, in which case all users that have cataloging authorization can use all cataloging fields.

Denied tags will display in gray (or as defined in the catalog.ini variables):
- DeniedFieldColor,
- DeniedFieldBackGroundColor.
The Cataloging GUI's status bar will note that the tag is denied as well as the usual tag definition.
Actions such as editing, delete, open sub-field, open new tag etc. will not be available for denied tags.

The entries are automatically added when the cataloging tables are repacked (UTIL M/7).

Structure of the table:
- Col. 1 User name
- Col. 2 Tag code (use # for wildcards)
- Col. 3 Permission
  - Y=allowed
  - N=denied
- Col. 4 Internal. This column is used only by the system in intermediate files.

Example of the table

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>!!!!!!!!!!!-!-!-!-!-!-!-!-!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td>###</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td>245##</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JUDY50</td>
<td>###</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SHIRLY50</td>
<td>###</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
```
**rep_tab_short.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Fields for printing bib information in columnar form for a report (p-print-08)

This table defines the structure of bibliographic print-outs in tabular form for the p-print-08 "line-list" report. The bibliographic fields are used to define the content of a column. The tags can also be virtual fields created by expand routines. The content can be formatted using edit_field (line identifier is 'S').

Tags can be listed on the same line in order of preference, so that the first tag found is taken to print. If Y is added at the end of the line, all occurrences of all tags listed will print.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>format number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>ALPHA</td>
</tr>
<tr>
<td>Col. 3</td>
<td>heading</td>
</tr>
<tr>
<td>Col. 4-8</td>
<td>1st – 5th alternative field + sub-field. # can be used as a wild card.</td>
</tr>
<tr>
<td>Col. 9</td>
<td>Width of the column, expressed as a percentage of the line. Therefore, all columns together should total 100</td>
</tr>
<tr>
<td>Col. 10</td>
<td>font</td>
</tr>
<tr>
<td>Col. 11</td>
<td>color</td>
</tr>
<tr>
<td>Col. 12</td>
<td>---</td>
</tr>
<tr>
<td>Col. 13</td>
<td>Starting position for fixed fields. The count is from base 001.</td>
</tr>
</tbody>
</table>

**Note:** when determining character position, if the fixed field begins with a sub-field code (as in UNIMARC field 100), the calculation does NOT have to take the three positions of the sub-field into account.

For example:

- for MARC21 field 008, date-1 is 008 (listed as 07-10 in the MARC21 manual).
- For UNIMARC field 100, date-1 is 010 (listed in the UNIMARC manual as 09-12).

<table>
<thead>
<tr>
<th>Col. 14</th>
<th>Number of characters for fixed field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 15</td>
<td>Y for all occurrences of the field</td>
</tr>
</tbody>
</table>

**Example:**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
scancode.dat

**Location of the table:** pc_tab/catalog directory of the library

**Purpose of the table:** Table for defining which authority lists are accessed on defined cataloging tags

This table defines the selection of libraries and the headings file of each one that is accessed when the "Search Headings" function is invoked in the Cataloging module.

**Note:** The first line of this table must have the word **LOCAL** in column one, and the code of each of the libraries that can be consulted, in columns 2-11. Column 12 is irrelevant to the first line.

The second and subsequent lines define which browse index is consulted for a particular field tag+indicators+sub-field. The sub-field is required only for cases where the particular sub-field is being searched for. A hash sign (#) can be used as wildcard for indicators.

Up to 10 libraries can be defined (cols. 2-10). Col. 11 is used to define sub-fields that are **NOT** overwritten when a heading is chosen from a browse list.

**Note:** It is possible to specify a library more than once on a line. This allows for scanning different codes. For example:

![Table](http://example.com/table.png)

When selecting the Search Field Headings of Other Library option for the 100 field, the cataloger is prompted to select one of the following options:

1. Scan USM10 with PER
2. Scan USM01 with AUT
3. Scan USM01 with SUB
Note that the first library specified in the table is the local library and it is not displayed when selecting to scan the headings from other libraries. For this reason, to enable the user to select this option when using this function, the column must be repeated.

In column 1, if specific indicators are used, they must be listed before general lines, since the first match found is always taken. Example:

24510
245##

Structure of the table:

| Col. 1 | Tag+indicators+sub-field code. Sub-field is defined in the 6th position. # can be used for indicators. The 6th position may be used for defining a specific sub-field that may be searched. |
| Col. 2 | Access Code. At the top of the column is the code of the searchable library. Below that, for each field tag defined in column 1, is the code of the headings list of the selected library. |
| Col. 3-10 | Heading (ACC) File code of additional libraries defined. (Code of the library/database defined as part of the header). |
| Col. 11 | Up to 20 sub-field codes that should not be overwritten. When the heading is copied to the cataloging draft, these sub-fields in the cataloging draft will then be retained. |

Note that when an heading which is a see reference is selected, the preferred heading from the authority record will be taken. For example:

Scan in USM01, select "AT & T" (which is a 4xx heading), "American Telephone and Telegraph Company" (which is the 1xx heading of the record) is taken instead.

If the heading is not in an AUT library and there is no reference to an AUT document, the heading is taken as is via correction using tab_sub-field_punctuation.

Example of the table

<table>
<thead>
<tr>
<th>LOCAL</th>
<th>USM01</th>
<th>USM10</th>
<th>USM12</th>
<th>USM01</th>
<th>USM11</th>
</tr>
</thead>
<tbody>
<tr>
<td>050##</td>
<td>LCC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>086##</td>
<td>SUD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100##</td>
<td>AUT</td>
<td>PER</td>
<td>PER</td>
<td>AUT</td>
<td></td>
</tr>
<tr>
<td>110##</td>
<td>AUT</td>
<td>COR</td>
<td>GEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>111##</td>
<td>AUT</td>
<td>MET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130##</td>
<td>TIT</td>
<td>TIT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>245##</td>
<td>TIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>260##a</td>
<td>PLA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>260##b</td>
<td>PUB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>440##</td>
<td>SRS</td>
<td>TIT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
server_ip_allowed

Location of the table: alephe/tab directory

Purpose of the table: Access definitions to WEB and GUI

This table lists the IP addresses that are allowed to access the ALEPH WEB interface, and ALEPH GUI interface.

Structure of the table:
- Col. 1 Server type:
  - N = NCIP server
  - P - PC server
  - W - web server
  - X = X-Server (part of WWW web server services)
- Col. 2 Allowed server IP address
  - A - allowed
  - D - denied
- Col 3 IP address. * signifies wildcard

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>A</td>
<td>10.1.235.*</td>
</tr>
<tr>
<td>W</td>
<td>A</td>
<td><em>.</em>.<em>.</em></td>
</tr>
<tr>
<td>X</td>
<td>A</td>
<td><em>.</em>.<em>.</em></td>
</tr>
<tr>
<td>P</td>
<td>A</td>
<td><em>.</em>.<em>.</em></td>
</tr>
<tr>
<td>N</td>
<td>A</td>
<td><em>.</em>.<em>.</em></td>
</tr>
</tbody>
</table>

tab_85x_import

Location of the table: tab directory of the ADM library

Purpose of the table: Matching fields for p-serial-52 (import 85XX records)

The p-serial-52 batch procedure is used to add 853-855 and 853X-855X fields from other libraries to the ADM records of the identical titles in your library.

Using the table it is possible to define:
- which index or word file should be used for the match,
- whether the entire field from the input file or a specific sub-field should be used for the match,
- which sub-field in the field in the database should be used.

It is not possible to match against a headings index.
A field can be repeated in the table so that the system checks sub-fields in the input file against various sub-fields in the database. (See, for example, 022 in the example below). In such a case, the field is entered only once in the service - the program will check all the occurrences defined in the table.

Structure of the table:
- Col. 1 Field tag
- Col. 2 Sub-fields in the imported data
- Col. 3 Sub-fields in the database data
- Col. 4 Index type (Word or Index)
- Col. 5 Database index for search

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!!!!!!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>I</td>
<td>I CNO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>#</td>
<td>a</td>
<td></td>
<td>I 010</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>!020##</td>
<td></td>
<td>I 020</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>a</td>
<td></td>
<td>I ISSN</td>
<td></td>
</tr>
<tr>
<td>022</td>
<td>#</td>
<td>a</td>
<td></td>
<td>I ISSN</td>
<td>I ISSN</td>
</tr>
<tr>
<td></td>
<td>a</td>
<td>z</td>
<td></td>
<td>I ISSN</td>
<td></td>
</tr>
<tr>
<td>022</td>
<td>#</td>
<td>y</td>
<td></td>
<td>I ISSN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>a</td>
<td></td>
<td></td>
<td>I 035</td>
<td></td>
</tr>
</tbody>
</table>

**tab_98_def**

Location of the table: tab/import directory of the BIB library

Purpose of the table: Default values for p-file-98 (BNA loader)

This table provides default values for various record fields when records are added to the database using p-file-98 (the BNA – Blackwell North America - loader).

Default values can be set for: Z30-CATALOGER, Z30-SUB-LIBRARY, Z30-COLLECTION, Z30-MATERIAL, Z30-ITEM-STATUS, Z30-ITEM-PROCESS-STATUS and Z30-CALL-NO-TYPE

Structure of the table:
- Col. 1 Name of the field for which value is being defined
- Col. 2 Input value - text that is supplied in the input record from BNA
- Col. 3 Record value - text that is written in the ALEPH record field

Example of the table:
Location of the table: tab/import directory of the BIB library

Purpose of the table: Default values for p-file-99 (Marcive loader)

This table is an example of an input file for p_file_99 (Marcive loader) which can receive values from an input file. The input file will be the last parameter of the p_file_99 script. The tab_99_def example defines default values for BIB, Z30 and HOL record fields.

Default values can be set for:

- **CATALOGER-NAME** - sub-field $$a$$ of the CAT field in the BIB, ADM and HOL records
- **Z30-CATALOGER** for item records
- **SUB-LIBRARY, COLLECTION, CALL-NO-TYPE, MATERIAL-TYPE, ITEM-STATUS** and **ITEM-PROCESS-STATUS** for item record fields
- **HOL-OWN-PREFIX and HOL-OWN-SUFFIX** together create OWN field in HOL record. OWN field is not created if both are blank.
- **HOL-007** sets the first two characters of the 007 field in the HOL record.
- **HOL-008** sets the value of the 008 field in the HOL record (first six bytes should be "xxxxxx" --system puts the create date in this position)
- **HOL-LIB-SYMBOL** sets the content of sub-field $$a$$ of the HOL record 852 field.

The MARCIVE loader procedure examines the location stamp in the first sub-field of the 049 field (up to 10 characters), and uses this data to set the sub-library and collection.

Structure of the table:

- **Col. 1** Name of the field for which value is being defined
- **Col. 2** Input value - text that is supplied in the input record from Marcive
- **Col. 3** Record value - text that is written in the ALEPH record field

Example of the table:
tab_abbrev

Location of the table: tab directory of the library

Purpose of the table: Expanding abbreviations to full text

Language sensitive tables

This table is used in conjunction with the expand_doc_fix_abbreviation routine. The fix_abbreviation routine is used in order to change abbreviations into full text. In actual fact, the routine can be used to replace any text string in a bibliographic record with a different text string.

The text exchange is defined per field tag + indicators. Field tags can be grouped by using # for wild card.

For each line, you can define X, Y or N.

- **X** defines that the tag is not to be considered at all, for any abbreviation fixing. Therefore, an X line should not include text in cols. 3 and 4.
- **N** defines that the text exchange should be ignored for the particular tag, or tag grouping. An N line is necessary only if it is followed by a Y line which uses #.
- **Y** defines that the text exchange should occur.

For example:

```
2#### Y Ft. FORT
```

changes Ft. to FORT in all fields that begin with 2.

But

```
245## N Ft. FORT
2#### Y Ft. FORT
```

changes Ft. to FORT in all fields that begin with 2, except for field tag 245.

**Note:**

- If there is more than one line for an abbreviation, with different codes, the lines with "N" should appear first.
- All lines with the same abbreviation should appear one after the other.
- Characters outside the ASCII range can be expressed in \textless backslash\textgreater hexadecimal notation, e.g. \textbackslash 266D for MUSICAL FLAT SIGN.

Structure of the table:
- col. 1: Field tag + indicators;
- col. 2: Usage: X, N or Y as explained above.
- col. 3: Text to be changed
- col. 4: Expanded form

Example of the table:

<table>
<thead>
<tr>
<th>!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!-</th>
<th>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</th>
</tr>
</thead>
<tbody>
<tr>
<td>255## Y 1st</td>
<td>FIRST</td>
</tr>
<tr>
<td>043## Y u-at---</td>
<td>Australia</td>
</tr>
<tr>
<td>043## Y u-at-ne</td>
<td>Australia New South Wales</td>
</tr>
<tr>
<td>043## Y u-at-no</td>
<td>Australia Northern Territory</td>
</tr>
<tr>
<td>043## Y u-at-qn</td>
<td>Australia Queensland</td>
</tr>
<tr>
<td>043## Y u-at-sa</td>
<td>Australia South Australia</td>
</tr>
<tr>
<td>043## Y u-at-tm</td>
<td>Australia Tasmania</td>
</tr>
<tr>
<td>043## Y u-at-vi</td>
<td>Australia Victoria</td>
</tr>
</tbody>
</table>

Note the following for defining this in tab\_expand:
It is possible to define whether the tab\_abbrev definition \textbf{adds} or \textbf{replaces} a line to the document. If you want to replace the line in the document instead of adding one, you may use col. 3 in tab\_expand with "REPLACE". This is useful for display purposes (where the abbreviated term need not display). Note that ‘add’ is the default if not defined.

Example (tab\_expand):

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!-</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U39-DOC expand_doc_fix_abbreviation</td>
<td>ADD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U39-DOC expand_doc_fix_abbreviation</td>
<td>REPLACE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All lines with the same abbreviation should appear one after the other.

\textbf{tab\_acq\_index}

Location of the table: tab directory of the \textit{ADM} library

Purpose of the table: Display of bibliographic fields in the Acquisition's Order index
This table defines the bibliographic fields for the ACQ Order Index. The name of the index must also be added to the list of field values in pc_tab_exp_field.lng - ACQ_INDEX_TYPE.

In order to create the index retrospectively run 'Rebuild ILL and Acquisitions Order Index' (p-acq-04) from the Acquisitions batch service.

Note: Make sure that the Index Type (Col. 1 of table) is unique and is not used in either of the following 2 ADM library tables:

- tab_ill_index - Display of bibliographic fields in ILL Outgoing requests index
- tab_ill_in_index - Display of bibliographic fields in ILL Incoming requests index

Note that the index entry is updated only if the order record is updated - i.e. a change in the bibliographic record will not automatically update the index in the Acquisitions Client.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Index name (up to 5 characters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>bibliographic field code+sub-field code</td>
</tr>
<tr>
<td>Col. 3</td>
<td>1st alternative</td>
</tr>
<tr>
<td>Col. 4</td>
<td>2nd alternative</td>
</tr>
<tr>
<td>Col. 5</td>
<td>3rd alternative</td>
</tr>
<tr>
<td>Col. 6</td>
<td>4th alternative</td>
</tr>
</tbody>
</table>

Example:

```
! 1 2 3 4 5 6
!!!!!!!!-!!!!!!!!-!!!!!!!!-!!!!!!!!-!!!!!!!!
TIT 245##a
AUT 100## 110## 111## 130##
ISBN 020##
ISSN 022##
```

**tab_alephadm.<lng>**

Location of the table: tab directory of the library AND the alephe/tab directory

Purpose of the table: List of files for activation from the online ALEPHADM

This table lists that files and their location for activation from the ALEPHADM module. It is library sensitive, and there should be a table for each library type (BIB, ADM, etc.). In addition the table, in the alephe/tab directory defines the common node tables accessible for update via the online ALEPHADM module

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Filename</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Language extension (Y/N)</td>
</tr>
</tbody>
</table>
Example of the BIB table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc_tab/catalog</td>
<td>scancode.dat</td>
<td>N L BROWSE list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc_tab/catalog</td>
<td>permission.dat</td>
<td>N L Cataloger</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc_tab/catalog</td>
<td>tag_text.dat</td>
<td>N L Pre-defined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc_tab/catalog</td>
<td>tagonnew.dat</td>
<td>N L Default Fields</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc_tab/catalog</td>
<td>codes</td>
<td>Y L Field codes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc_tab/catalog</td>
<td>fix_doc</td>
<td>Y L Fix routines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pc_tab/catalog</td>
<td>pc_tab_find</td>
<td>Y L Search screen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example of the ADM table:

<table>
<thead>
<tr>
<th>tab</th>
<th>tab14</th>
<th>N L Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reshelving Time</td>
<td>tab15</td>
<td>Y L Item Statuses</td>
</tr>
<tr>
<td>tab</td>
<td>tab16</td>
<td>N L Due dates,</td>
</tr>
<tr>
<td>fines &amp; limits</td>
<td>tab17</td>
<td>N L Library</td>
</tr>
<tr>
<td>Opening Hours</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example of the ALEPHE table:

<table>
<thead>
<tr>
<th>tab</th>
<th>tab_base</th>
<th>Y L Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>of databases</td>
<td>allowed_languages</td>
<td>N L Allowed</td>
</tr>
<tr>
<td>languages</td>
<td>tab_days</td>
<td>N L tab_days</td>
</tr>
<tr>
<td></td>
<td>tab_month</td>
<td>N L Months</td>
</tr>
</tbody>
</table>

**tab_attr_sub_library**

Location of the table: alephe/tab directory

Purpose of the table: Sublibrary relationships based on station ID / IP address relating to loan/return, cash and reading room management
This table defines, for a particular work station, or group of work stations, the sub-libraries that come under its jurisdiction. Different type definitions (col. 2) define different options:

**NOTE:**
- For Attribute types 1, 2, 3 and 6 in column 2, you can setup up to 500 sublibraries.
- For Attribute types 4, 5, 7 and 8 in column 2, you can setup only one sublibrary/code.

Sub-libraries of the same type must be grouped together. The sub-library code must be filled to 5 characters, and listed with one space between each sub-library. In other words, if the sub-library code is only 3 characters, it will then have 2 blanks for fill, and an additional blank for separator before the next sub-library code.

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>attribute identification for ip address identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>attribute type:</td>
</tr>
<tr>
<td></td>
<td>1 - item can be loaned and returned</td>
</tr>
<tr>
<td></td>
<td>2 - item can be returned only</td>
</tr>
<tr>
<td></td>
<td>3 - home sub-libraries (for transfer)</td>
</tr>
<tr>
<td></td>
<td>4 - reading room loan handling</td>
</tr>
<tr>
<td></td>
<td>5 - cash generation sub-library</td>
</tr>
<tr>
<td></td>
<td>6 - sorting of cash transactions</td>
</tr>
<tr>
<td></td>
<td>7 - preferred sublibrary for sorting routine 06</td>
</tr>
<tr>
<td></td>
<td>8 - cash receipt counter suffix</td>
</tr>
<tr>
<td>cols. 3</td>
<td>sub-library code/counter suffix</td>
</tr>
<tr>
<td>cols. 4-7</td>
<td>sub-library codes</td>
</tr>
</tbody>
</table>

**Attribute types (col. 2)**

**Type 1 - item can be loaned and returned**
Use type 1 if you want to be able to loan and return items of the sublibrary at this station. If the sublibrary of the item is not listed as type 1 for this workstation, the item cannot be loaned or returned, and a message is displayed, when a loan or return is attempted.

**Type 2 - item can be returned only**
Use type 2 if you want to be able to return (but not loan) items of this sublibrary at this workstation. If the sublibrary of the item is not listed as type 2 for this workstation, the item cannot be returned, and a message is displayed, when a return is attempted.

**Type 3 - sublibraries of items (for transfer)**
Use type 3 to define that the items returned belong to this sublibrary. If the item's sublibrary is not listed in the type 3 group, the return is considered a "transfer" return, that is, the item belongs to a different library. A message is displayed, and a transfer slip is printed.
The item will appear as "In Transit" until it is returned to its own sublibrary. If there is a patron record that has the item's sublibrary code as its ID, the item is also loaned to the sublibrary for the transfer. Such a patron has to have a local (Z305) record. This Z305 record must have at least

- $z305\_loan\_permission$ set to "Y",
- $z305\_loan\_check$ set to "N",
- $z305\_ignore\_late\_return$ set to "Y",
- $z305\_rr\_permission$ set to "Y".
- It is also advisable to set $z305\_expiry\_date$ to the latest possible date and
- the $z305\_cash\_limit$ to the highest possible sum.

A type 3 sublibrary must also be registered as type 1 or 2 or 4 for the same Station ID/IP address for the purpose of circulation activities.

**Type 4 - reading room loan handling**

Use type 4 to define a Reading Room workstation. It is important to note that when there is a Reading Room sublibrary defined for an Station ID/IP address it means that all the loan transactions performed at this Station ID/IP address will be treated as Reading Room loans.

This means that the item's sublibrary will change to the Reading Room and will remain as such as long as the item is on loan. When the item is returned, the item's sublibrary will change back according to the item's original sublibrary (Z30_SUB_LIBRARY). When a Station ID/IP station serves a regular sublibrary (that is, not a Reading Room sublibrary), type 4 should be left empty.

**Type 5 - cash generation sublibrary**

Use this type for cash transactions that are not item specific. When a transaction of this type (for example self-registration of patrons in the Web OPAC, photocopy request, ILL processing) occurs at the Station ID/IP address, the sublibrary registered here will be written in the cash record.

You can define only one sublibrary of type 5. If no sublibrary is entered as type 5, the active ADM library is automatically taken.

**Type 6 - sorting of cash transactions**

This type filters the cash transaction records listed in the Patron Activity/Cash pane. Only the transactions of the libraries that are registered here will display. The amount in the "sum to pay" is adjusted accordingly.

**Type 7 - preferred sublibrary for items sorting routine 06**

Defines which sublibrary will be the first when sorting items list by sorting routine 06. Only one sublibrary code should be defined for this type in column 3.

This type can also be used in order to associate a workstation with a specific sublibrary. This setup is used to determine the sender sublibrary for a Transfer Slip between sublibraries.
Type 8 - cash receipt counter suffix
If defined, the cash receipt number will use the counter last-cash-rec-<suffix> instead of last-cash-receipt-no. (UTIL G/2). You can enter any code up to 5 characters in column 3. Only one code should be set up.

For example, if you enter xxxxx as type 8, the counter should be last-cash-rec-xxxxx.

If one of the parameters is missing, the system will use the last-cash-receipt-no counter.

Example of the table:

```
<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!!!!!!!!!!!!!!!!!!-!-!!!!!-!!!!!!-!!!!!!-!!-!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.1.235.85</td>
<td>URDOC</td>
<td>1</td>
<td>UARCV</td>
<td>UELEC</td>
<td>URLEC</td>
<td>UGDOC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>UEDUC</td>
<td>UHLTH</td>
<td>ULAB</td>
<td>UMUSI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ULINC</td>
<td>1</td>
<td>USCI</td>
<td>URPTN</td>
<td>USM50</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>ELEC5</td>
<td>GDOC5</td>
<td>EDUC5</td>
<td>LINC5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LINC6</td>
<td>URDOC</td>
<td>3</td>
<td>UARCV</td>
<td>UELEC</td>
<td>URLEC</td>
<td>UGDOC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>UEDUC</td>
<td>UHLTH</td>
<td>ULAB</td>
<td>UELEC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ULINC</td>
<td>3</td>
<td>USCI</td>
<td>UUPTN</td>
<td>URPTN</td>
<td>UREXU</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>LINC5</td>
<td>EDUC5</td>
<td>GDOC5</td>
<td>ELEC5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>5</td>
<td>ULINC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>7</td>
<td>UEDUC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**tab_aut**

Location of the table: tab directory of the **BIB** library

Purpose of the table: Matching of BIB library's Headings (ACC) index to AUThority records

There are several aspects to the link between BIB and AUT type libraries. One of these is the enrichment of the BIB headings table (Z01) from the AUT including cross-references from non-preferred terms and language equivalents of headings.

For the update of z01 from AUT:

- The UTIL E/8 daemon must be running.
- If a new heading is inserted in BIB the relation to AUT is set to –NEW–.
- Using tab_aut, the E-8 daemon checks all records marked as –NEW– against headings in the AUT library.
- If a parallel record is found in the linked AUT, a link to the AUT record is added to BIB heading record.
- **tab_aut** defines, for each BIB headings list, which AUT library to search for a match.
- For more on AUT-BIB links refer to UTIL E-8.

**NOTE:** In order for tab_aut to work correctly there must be a GENERAL (with code GEN) headings table in the AUT library to which all authority terms and their cross references are indexed.

Note: Maximum number of lines – 100.

**Structure of the table:**

Col. 1 headings list (ACC) code in BIB library

Col. 2 Usage code from tag 008 of the AUT library which must be a or c:

- 1 - pos. 14 (Main heading)
- 2 - pos. 15 (Subject)
- 3 - pos. 16 (Series)

If left blank there is no check on the usage codes in the authority record.

Col. 3 code of 1st AUT library

Col. 4 code of 2nd AUT library

Col. 5 code of 3rd AUT library

Col. 6 code of 4th AUT library

The **usage code** filters the match according to the values for heading use defined in the 008/14-16 field of the authority record. The AUT record is considered a match only if both the text and the usage codes match.

**Example of the table:**

```
! 1 2 3 4 5 6
!!!!!!-!-!!!!!!-!!!!!!-!!!!!!-!!!!!!
AUT TIT TIT SRS SUL SUB
1 USM10
USM10
USM10
USM10
USM12
```

**tab_base.<lng>**

Location of the table: alephe/tab directory

Purpose of the table: Database definitions

The tab_base table defines the logical and physical databases that can be accessed by the user through WEB OPAC and GUI SEARCH functions.

**Logical bases:**
Logical bases are defined by setting a FIND command that serves as a pre-filter or scope. The find command can have up to 20 operators and be up to 500 characters long.

In order to set a logical base to include everything EXCEPT a group of records, use `alldocuments` to define everything, together with `not`.

E.g. alldocuments not wsp=suppressed.

Note that "alldocuments" is actually all records that are indexed in the WRD word group.

**Limited access**

As part of the `tab_base` definitions, it is possible to define the libraries that can be entered without signing in. To prevent a non signed-in patron from entering a base, * (asterisk) should be entered in col. 2.

Example:

| ISRAEL     | * Israel | USM01 | USM01 wti=history |

**Extension code - Base-sensitive WEB screens:**

When the patron connects to a base (logical or physical) in the WEB OPAC, the system uses html pages that have an extension that is the same as the base code. In addition to the base code itself, up to 3 extensions can be defined (in col. 4-6). In this way, it is possible to set a common extension for a group of base codes. The system chooses the first match found from the 3 columns. If no matching html page is found, the system uses the page that does not have an extension.

**Extension code - Base-sensitive full record display**

The extension code mechanism described for WEB html pages is also used for tables that govern full record display:

- `edit_doc.<lng>`
- `edit_doc_999.<lng>`
- `edit_doc_999_aut_<aut_library>.<lng>`

This mechanism serves both WEB OPAC and GUI SEARCH.

**Base-sensitive browse list:**

In the browse list, the display of the counter of records related to a heading can be set to base-sensitivity for more than one logical base, using `/alephe/tab/tab_base_count`.

See `tab-base_count` for a more detailed explanation.

**Base-sensitive links to library holdings (items):**

The list of sublibraries (using the ITM3 or ITM5 lines in `edit_doc_999`, and in the drop-down list of libraries on the list of items display), can be filtered according to logical bases, using `/lib/tab/tab_item_list_order`.

**Logical base efficiency - Z0102:**

In order to make the browse functionality more efficient for small logical bases, you should utilize the Z0102 Oracle table. Usage is defined in this `tab_base` table in column 8. Note that up to 200 small bases can be defined using this mechanism.
Use UTIL H/1/10 to determine whether Z0102 should be generated for a particular base. It is recommended Z0102 be used only for bases that constitute less than 10% of the total database. It cannot be used for bases defined by "alldocuments not ...".

Structure of the table:
- Col. 1 base code
- Col. 2 * - Limit access to signed-in patrons;
- Col. 3 base name (as will display in the Web OPAC)
- Col. 4-6 WWW base extension code x 3 (i.e. up to 3 possible fallbacks).
- Col. 7 Code of the physical library
- Col. 8 Use Z0102 (limited to 200)
  - Y = Use Z0102
  - N = Do not use
- Col. 9 Find command (maximum 500 characters including spaces. Maximum 20 conditions in a FIND command)

Example of table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>USM01</td>
<td>USMARC BIB (USM01)</td>
<td>USM01 USM01</td>
<td>USM01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM10</td>
<td>USMARC AUT (USM10)</td>
<td>USM10 USM01</td>
<td>USM10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM20</td>
<td>USMARC ILL (USM20)</td>
<td>USM20 USM01</td>
<td>USM20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>SERIALS</td>
<td>Serial (USM01)</td>
<td>USM01</td>
<td>USM01 Y wfm=SE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RUSSIAN</td>
<td>Russian (USM01)</td>
<td>USM01</td>
<td>USM01 N wln=RUS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FRENCH</td>
<td>French Lang.</td>
<td>USM01</td>
<td>USM01 Y wln=fre</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>* United States</td>
<td>USM01</td>
<td>USM01 Y wrd = unit ed states and wsl = uarcv or ueduc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!*=================================</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LINC</td>
<td>Lincoln Library</td>
<td>USM01</td>
<td>USM01 N wsl=ulinc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>HEAL</td>
<td>Health Library</td>
<td>USM01</td>
<td>USM01 N wsl=uhlth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIEDUC</td>
<td>Linc/Educ</td>
<td>USM01</td>
<td>USM01 N wsl=(ulinc or ueduc)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_base_count**

Location of the table: alephe/tab directory

Purpose of the table: Logical bases (record counters) in Web and GUI browse list

The tab_base_count is used for OPAC display of browse lists, that takes logical bases into account. Using this table, up to three columns can be set, for three logical bases. Each column displays a counter of the number of relevant records.

Standard browse list display:
Browse lists of WEB OPAC and GUI SEARCH display the headings relevant to the logical base being searched, and the counter of records is calculated accordingly. For libraries that do not use this tab_base_count table, the browse list will show the counter and the heading.

Multiple base browse list display:
Browse lists of WEB OPAC and GUI SEARCH can be set up to display separate counters for the number of records related to a heading for a "primary" logical base, and up to two "secondary" bases. The primary base must be set up in tab_base to include all the records that will be included in each of the secondary bases.

The primary base is set in column 1, and the count of the number of records displays in the "Total" column in WEB OPAC, and in "Docs" column in GUI SEARCH. The secondary bases are set in cols. 2 and 3 for WEB OPAC, and in columns 4 and 5 for GUI SEARCH. The base code displays at the head of each of these columns.

Display in WEB OPAC:
In order for the base counter columns to display in the WEB OPAC, scan-acc-head and scan-acc-body html pages must be set up properly. Proper setup includes:
- add the base code or an alternate extension (defined in tab_base.<lng>) to the html file name (e.g. scan-acc-head-educ)
- include column definitions in scan-acc-head, e.g.
  <th class="text3" nowrap>Total</th>
  <th class="text3" nowrap>$0500</th>
  <th class="text3" nowrap>$0600</th>
  <th class="text3">Entry</th>
- include column definitions in scan-acc-body, e.g.
  <td class=td1 id=centered width=8%>$0200</td>
  <td class=td1 id=centered width=8%>$0700</td>
  <td class=td1 id=centered width=8%>$0800</td>
  <td class=td1 >$0400$0500$0600</td>

Display in GUI Search:
In order for the columns to display in GUI SEARCH, pc_tab_col.<lng> must include the following lines:

PC_SEAR_SCAN L $1 05 010 01 C01 Base-1 count
PC_SEAR_SCAN L $2 06 010 01 C01 Base-2 count

The browse list of headings is filtered to display the headings that are relevant to the base defined in col.1. Therefore, the OPAC headings display will make sense only if the second and third columns are subsets of the first column.

For example:
Col. 1 - entire database
Col. 2 – Ex Libris University
Col. 3 - Math/Physics library in Wazawoo University
    OR
Col. 1 - Ex Libris University
Col. 2 - Math/Physics library in Ex Libris University
Col. 3 - Humanities library in Ex Libris University

Structure of the table:
Col. 1 Base code
Col. 2-3 Additional Base in WWW browse list
Col. 4-5 Additional Base in GUI browse list

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>L</td>
<td>L</td>
<td>E</td>
<td>L</td>
</tr>
<tr>
<td>!</td>
<td>U</td>
<td>L</td>
<td>E</td>
<td>L</td>
</tr>
<tr>
<td>!</td>
<td>S</td>
<td>L</td>
<td>E</td>
<td>L</td>
</tr>
<tr>
<td>!</td>
<td>M</td>
<td>E</td>
<td>U</td>
<td>U</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>U</td>
<td>E</td>
<td>U</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td>L</td>
<td>E</td>
<td>L</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td></td>
<td>L</td>
<td>E</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td></td>
<td>E</td>
<td>L</td>
</tr>
<tr>
<td>!</td>
<td></td>
<td></td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>

**tab_block_circ**

**Location of the table: tab directory of the USR library**

**Purpose of the table: Parameters for blocking loans**

This table is used to determine the values for checks 9a and 9b in tab_check_circ. 9a is for checks at the ADM level and 9b is for checks at the sub-library level.

Values can be set for the patron at Global (ADM) level (9a) for
- maximum number of overdue loans or
- maximum number of recalled overdue loans or
- maximum fines owing.

Values can be set for the patron at sub-library level (9b) for
- maximum number of overdue loans or
- maximum number of recalled overdue loans.

The cash limit does not work at the sub-library level.

If the patron passes the threshold, the circulation transaction for which the check was defined in tab_check_circ will be blocked.

If the transaction is blocked, lines 0241, 0242, 0243, 0251 or 0252 from the file alephe/error_<lng> /check_circ display the reason for the block.

Structure of the table:
- col. 1 Patron Sub-library
- col. 2 Patron status
- col. 3 Item Sub-library
- col. 4 Maximum number of overdue loans
- col. 5 Maximum number of recalled overdue loans
col. 6 Maximum amount of fines owing. Amount can be entered as full number (e.g. 500), or with decimal point (e.g. 500.00). Leading zeroes are not required (e.g. 500, not 000000000500).

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>USM50</td>
<td>##</td>
<td>####</td>
<td>05</td>
<td>04</td>
<td>10000</td>
</tr>
<tr>
<td>UARCV</td>
<td>##</td>
<td>####</td>
<td>02</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>UELEC</td>
<td>03</td>
<td>####</td>
<td>04</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>UELEC</td>
<td>06</td>
<td>####</td>
<td>04</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>UELEC</td>
<td>##</td>
<td>####</td>
<td>02</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>UEDUC</td>
<td>##</td>
<td>####</td>
<td>03</td>
<td>02</td>
<td></td>
</tr>
</tbody>
</table>

**tab_bor_address**

*Location of the table: tab directory of the USR library*

*Purpose of the table: Patron address periods*

This table is used by libraries which prefer to define the "current patron address" by time periods in the year (such as semester 1, semester 2, summer holiday, spring break), rather than by the convention of address type

- 01 for permanent address
- 02 for mailing address, etc.

together with valid from-to dates in the address record itself.

Define time periods in this table if you want your address records to be defined accordingly. Otherwise, leave the table empty and the system will automatically work with the "01" and "02" method.

**Note that the table MUST be present (even if it is empty).**

The table defines the periods from-date - to-date, and then the patron address type that is used for the time period.
The dates in the table must be continuous. They must NOT overlap, and there must be NO gaps.
The patron address type is entered in the relevant patron address record, when the record is created or updated. Note that the dates from-to in the patron record are ignored, and the first address record with the matching address type is used by the system.

Each line defines which address-type identification is in effect for a time period.

**If the address period is defined in this table:**

When the patron updates his address record in the Web OPAC, the current record will be updated. If there is no current record, the record that suits the immediately preceding line in this table will be used (only one line up) to create a new (current)
In the same manner, when the system searches for the appropriate patron record for printed products, if there is no record with the address type expected according to the time period in the table, the system takes the record that has the same type as is registered in the previous line in the table, and duplicates it, assigning the expected address type.

If the address period is NOT defined in this table:
Updating the address in the Web OPAC (using the Update Patron Address form, filename bor-update) will NOT change the last record, it will create a new record with an increased sequence number.

When the system determines which address record to use, if there is more than one record with relevant valid from-to dates, the system chooses the record type 02 (mailing address) with the highest sequence number. If the system does not find a suitable 02 address, the system then searches for a suitable 01 (permanent) address. If no address is found, no address is printed.

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>From date</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>To date</td>
</tr>
<tr>
<td>col. 3</td>
<td>Address type</td>
</tr>
</tbody>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>19990101</th>
<th>19990631</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>19990701</td>
<td>19991231</td>
<td>22</td>
</tr>
</tbody>
</table>

**tab_bor_id.<lng>**

Location of the table: tab directory of the USR library

Purpose of the table: management of patron ID’s

This table defines all system-wide behavior relating to the Z308 (global patron’s IDs) records. This table exists only in the library defined by the environment variable usr_library (e.g. USR00) in aleph_start.

Note that it is possible to use more than one additional ID type as a key to retrieve Z303 (Web OPAC and GUI).

IDs that are NOT used as a key to retrieve Z303 in the GUI (N in column 5), can be duplicated for different patrons. Duplication for the same patron is always allowed.

The ID date can be the same but the verification must be different so there won't be two identical sets of ID and verification.

Duplication is not possible between IDs that are not used as a key to retrieve Z303 in the GUI and those that are used as key in the GUI (Y in column 5).
IDs of type 00 or 01 are always unique throughout the system.

Structure of the table:

Col. 1 ID code
- 00 - system ID. Mandatory and cannot be updated.
- 01 - barcode. Mandatory, can be updated.
- 02 - additional ID. Optional, can be updated.
- nn - additional IDs, Optional, can be added or deleted.

Col. 2 ALPHA
Col. 3 Description of type of ID
Col. 4 Not used. Enter 00
Col. 5 GUI: can be used as a key to retrieve a Z303 record in the GUI.
In this column, for ID code 00 - System ID, the value may be set to "N" if the value for ID code 01 - Barcode is set to "Y". In this case the system will only look for a patron's Barcode and not for his System ID.

Col. 6 WWW: can be used as a key to retrieve a Z303 record in the WEB OPAC. A verification is always required.

Col. 7 Verification field is optional (N) or mandatory (Y):
Y=when updating a patron record in the GUI, this ID must have a verification number.
For example, system ID must have pin code, barcode must have verification-1, etc.
ID's which are Y in column 6 must also be Y in this column (col. 7)

Col. 8 Update in WWW. this column is used in order to identify which verification (PIN) code is updated from the WEB interface. Any verification which is set to "Y" will be updated according to the code entered. Col.8 must be set to "Y" in order for the patron be able to recall a forgotten PIN code.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>00</td>
<td>L</td>
<td>System number</td>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>L</td>
<td>Barcode</td>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>L</td>
<td>Additional ID 02</td>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>L</td>
<td>Additional ID 03</td>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>L</td>
<td>Additional ID 04</td>
<td>00</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>L</td>
<td>Additional ID 05</td>
<td>00</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>L</td>
<td>Campus Access</td>
<td>00</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

**tab_buf_z403**

Location of the table: tab directory of the BIB library
Purpose of the table: List of programs for building buf_z403.

`tab_buf_z403` is used for the creation of `BUF_Z403`, which is used to govern access to electronic resources.

Each line in the table under the "BUILD" function in column 1 determines how the line from the document will be loaded into the z403 buffer.

The SHORT-VIEW function is used to define the display of fields that contain $$u in the brief display format table.

Data in `BUF_Z403` is used to govern access to electronic resources. `BUF_Z403` is used for fields which are defined as "M" (link to Electronic resource) in column 10 of `edit_doc_999.<lng>`.

Each line in this table must have a corresponding tag in `edit_doc_999.<lng>`, with link type value (Col. 10) $M$.

Column 3 in `tab_buf_z403` can be used for defining display parameters for the FULL display in WEB OPAC. In this case, this column is used instead of columns 3, 4 and 6 in `edit_doc_999`.

The use of Column 3 is as follows:

\[
\text{[display_tag, display_filters, subfield to filter on, contents to filter for, edit-field indicator,]}
\]

The parameters are separated by commas.

Examples for Column 3:

\[
856##, ,y, *, Y, \\
856##, ,y, -, U,
\]

In this example, the 856 field is the basis for display.

The edit_field formatting that will be used depends on the presence or absence of $$y in the 856 field.

The program will display the 856 field if the 856 field has subfield y with any data (*).
If it does, $Y$ is used for matching Col. 4 (ID of edit_field line) in the library's edit_field table. If the 856 field does not contain subfield y (-), the $U$ is used for matching Col. 4 in edit_field.

Structure of the table:

- col. 1  Function name
- col. 2  Program name
- col. 3  Program arguments

Example of the table:
### `tab_cat_hidden_fields`

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** Defines which fields are not displayed in the Catalog Editor

This table defines which fields are not displayed in the Catalog Editor. Since the fields included in this table are not displayed, they cannot be updated through the Cataloging module.

**Structure of the table:**

Col. 1  Field code

**Example of the table:**

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>!++++</td>
</tr>
<tr>
<td>555</td>
<td></td>
</tr>
</tbody>
</table>

### `tab_ccl_boolean_operators`

**Location of the table:** alephe/tab directory

**Purpose of the table:** Language definitions of CCL Boolean operators

This table defines which words are treated as Boolean operators in given language. For each operator a translation to the ALEPH 500 standard Boolean operator (AND, OR, NOT) is defined.

**Structure of the table:**

Col. 1  Language code
Col. 2  CCL operator in given language
Col. 3  ALEPH CCL operator
Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!-!!!!!!!!!!!!!!!-!!!!!&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>###</td>
<td>OR</td>
<td></td>
</tr>
<tr>
<td>### ~</td>
<td>NOT</td>
<td></td>
</tr>
<tr>
<td>### +</td>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>### &amp;</td>
<td>AND</td>
<td></td>
</tr>
</tbody>
</table>

DAN IKKE NOT
DAN OG AND
DAN ELLER OR

**tab_character_conversion_line**

*Location of the table:* alephe/unicode directory

*Purpose of the table:* Unicode conversion tables

This table defines the character conversion procedure and character conversion table to be used in various instances when character conversion is needed.

**The instances are:**

<table>
<thead>
<tr>
<th>Name of program</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF_TO_URL</td>
<td>translation of the url link in field 856 from UTF-8 to standard required for URL</td>
</tr>
<tr>
<td>UTF_TO_WEB_MAIL</td>
<td>translation of UTF-8 bibliographic data for MAIL and PRINT options in WEB OPAC</td>
</tr>
<tr>
<td>LOCATE</td>
<td>translation of data for the locate query.</td>
</tr>
<tr>
<td>FILING-KEY-nn</td>
<td>translation for filing purposes. This is not system-set, and there can be multiple tables. The &quot;nn&quot; must be coordinated with the char_conv line in the library's /tab/tab_filing table</td>
</tr>
<tr>
<td>MARCIVE_TO_UTF</td>
<td>translation of data imported from MARCIVE to UTF-8</td>
</tr>
<tr>
<td>RLIN_TO_UTF</td>
<td>translation of data imported from RLIN to UTF-8</td>
</tr>
<tr>
<td>YBP_TO_UTF</td>
<td>translation of data imported from to UTF-8</td>
</tr>
<tr>
<td>OCLC_TO_UTF</td>
<td>Translation from oclc imports to utf-8</td>
</tr>
<tr>
<td>UTF_TO_LOW</td>
<td>translation of the file which will be automatically created on the local PC if a LOW tag is added to a document</td>
</tr>
<tr>
<td>UTF_TO_VST</td>
<td>translation of data sending by the VST server</td>
</tr>
<tr>
<td>UTF_TO_LSSU</td>
<td>translation of data sending to a Sensomatic Selfcheck machine</td>
</tr>
<tr>
<td>UTF_TO_3M</td>
<td>translation of data sending to a 3M Selfcheck machine</td>
</tr>
<tr>
<td>ACQ_INDEX</td>
<td>translation utf to utf for Order and ILL Indexes</td>
</tr>
<tr>
<td>VENDOR_NAME_KEY</td>
<td>translation utf to utf for Vendor index</td>
</tr>
<tr>
<td>COURSE_NAME_KEY</td>
<td>translation utf to utf for Course Reading index</td>
</tr>
<tr>
<td>ADM_KEYWORD_KEY</td>
<td>translation utf to utf for keyword searching in ADM clients (budget, vendor)</td>
</tr>
<tr>
<td>BORROWER_NAME_KEY</td>
<td>translation utf to utf for index of patrons</td>
</tr>
</tbody>
</table>
The character conversion procedures are:

- `line_utf2line_sb` - translates utf data to single-byte
- `line_utf2line_utf` - translates utf data to other utf
- `line_sb2line_utf` - translates single-byte data to utf
- `line_marc8_2_line_utf` - translates marc8 to utf
- `line_mab2line_utf` - translates MAB to utf

For `line_utf2line_utf`, which allows translating a single utf character to up to 5 characters, there is a mechanism that will allow exchanging a string of up to 5 characters for another string of up to 5 characters. This is accomplished by creating an additional table that has the same name as the basic table, adding ".extended" to the table name. For example, "unicode_to_locate" can have an additional table named "unicode_to_locate.extended".

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Environment</td>
</tr>
<tr>
<td></td>
<td>PC, WWW, or any environment</td>
</tr>
<tr>
<td>Col. 3</td>
<td>ALPHA</td>
</tr>
<tr>
<td></td>
<td>For further refinement of col. 1. Identifies the ALPHA code of the field, where relevant</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Name of procedure to run</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Character conversion table</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Backslash notation. This column is relevant only for procedures</td>
</tr>
<tr>
<td></td>
<td><code>line_utf2line_sb</code></td>
</tr>
<tr>
<td></td>
<td><code>line_sb2line_utf</code></td>
</tr>
<tr>
<td></td>
<td><code>line_mab2line_utf</code></td>
</tr>
<tr>
<td></td>
<td>It indicates that a utf-8 character which cannot be displayed in the current font is displayed in the &quot;backslash-hexa&quot; format (e.g. <code>\05E9</code>). Note that translation tables can use &quot;01&quot; in order to compress the character.</td>
</tr>
</tbody>
</table>

The following are examples of some specific code page conversions to UTF and vice_versa, which might be required for cataloging fix procedures, import/export of records, etc.

- `8859_1_TO_UTF`
- `UTF_TO_8859_1`
- `8859_8_TO_UTF`
- `UTF_TO_8859_8`
- `8859_7_TO_UTF`
- `UTF_TO_8859_7`
- `8859_5_TO_UTF`
- `UTF_TO_8859_5`
- `UTF_TO_MARC8`
- `UTF_TO_MAB`
Note that MARC8_TO_UTF conversion is different from the above procedures. For this procedure, col.6 should be left blank, since the procedure itself is set to use the following tables:

- marc8_ara_to_unicode
- marc8_heb_to_Unicode
- marc8_eacc_to_unicode
- marc8_lat_to_Unicode
- marc8_greek_to_unicode
- marc8_rus_to_Unicode

In addition, some of the MARC8_TO_UTF conversion values are set in the program itself, and not in the tables.

Example of the table:

<table>
<thead>
<tr>
<th>UTF_TO_URL</th>
<th>UTF_TO_WEB_MAIL</th>
<th>UTF_TO_WEB_MAIL_ASCI</th>
<th>UTF_TO_TELNET_ASCI</th>
<th>LOCATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTF_TO_URL</td>
<td>UTF_TO_WEB_MAIL</td>
<td>UTF_TO_WEB_MAIL_ASCI</td>
<td>UTF_TO_TELNET_ASCI</td>
<td>LOCATE</td>
</tr>
<tr>
<td>UTF_TO_URL</td>
<td>UTF_TO_WEB_MAIL</td>
<td>UTF_TO_WEB_MAIL_ASCI</td>
<td>UTF_TO_TELNET_ASCI</td>
<td>LOCATE</td>
</tr>
</tbody>
</table>

**tab_check_circ**

Location of the table: tab directory of the ADM library

Purpose of the table: checks for loans and renewals

This table sets the checks that will be performed when an item is loaned or renewed.

Column 1 defines when the check is performed:

- **LOAN-GBL** - checks global patron/local patron prior to loan and when "Check" is performed on the Global Patron Information window
- **LOAN** – checks when item is being loaned to a patron. Do not use with check_circ_11_b.
- **RR-LOAN** - item is being loaned at a reading room station (type 4 in tab_attr_sub_library. Intended for checking the patron using the original sub-library of the item to find the z305 patron record. Used with check_circ_2_d.
- **RENEW** – checks when item is being renewed. applies to Renew and Renew All functionality.
- **RENEW-WWW** – checks when patron renews loan in the Web OPAC
- **RENEW-GBL** - checks on patron prior to ‘renew all’
- **EXTEND** checks for "Change Date" function
- **during Loan** in Patron's Loan List window
- **B-CIR-10** - item renewed to a patron (p-cir-10). This section is only relevant if the service cir-10 is run with "Automatic Renewal" option set to "Yes".
- **B-CIR-08** - item renewed to a patron (p-cir-08). Note that running the service cir-08 with a format that has a "Hold Status" column, will give incorrect results if the routine check_circ_6 in the B-CIR-08 section is active.

- **B-CIR-77** - Update patron records (p-cir-77), enables the batch cir-77 to search for Z-305 records containing delinquencies.

- **ROUT-LOAN** - item is being checked out to a routing member

- **X-GBL** - Checks global patron / local patron prior to renewal in X Server

- **X-RNW** - Item renewed to a patron in X Server

- **SIP-AUTH** - checks during the authentication of a patron through in SIP2 server

- **SIP-LOAN** - item is loaned/renewed through SIP2 server

- **SIP-RETURN** - item is returned through SIP2 server

- **RMT-REG** - checks regarding the duplication of a patron from a remote system.

Column 2 defines what check procedure to perform:

- **1_a** - patron (z303) delinquencies
- **1_b** - patron (z305) delinquencies
- **1_c_a** - Patron expiry date: z305_expiry_date
- **1_c_b** - Patron expiry date renew warning: z305_expiry_date. Renew warning according to parameters set in tab30
- **1_d** - overdue check. Works according to tab100 -OVERDUE-LETTER-NO which determines when an item is overdue" (i.e. due date has passed and "n" (0/1/2/3/4) notices have been sent)
- **1_e** - block check - z305_end_block_date
- **1_f** - Proxy Patron expiry date - z305expiry_date. This check works with LOAN, RENEW and RENEW_WWW
- **2_a** - check patron loan permission - z305_loan_permission
- **2_b** - check patron renew permission - z305_renew_permission
- **2_c** - check patron reading room permission - z305_rr_permission, taking the z305 that matches the reading room sub-library at which the loan is being performed sub-library (i.e. when item is loaned at a reading room station - type 4 in tab_attr_sub_library
- **2_d** - check patron loan and reading room permission using the original sub-library of the item (i.e. the sub-library of the item before it was assigned the Reading Room sub-library).
- **2_e** - If the patron tries to loan/renew/return a magnetic media (tab25.<lng>, col. 4 = Y) an error message (0039 from error_lng/sc_global) is called up
- **3** - check patron renew limit according to parameters set in tab16 (col.21)
- **4_a** - check patron loan limits according to parameters set in tab16 (col.12)
- **4_b** - check the patron's overall loan limit in a sub-library according to parameters set in tab16 - 99 line
- **4_c** - check the patron's overall total loan limits for ADM library according to parameters set in tab16) 99 line
- **5** - check cash limit checks whether the patron has exceeded the amount allowed in z305_cash_limit
- **5_b** - check patron cash limit checks whether the patron has exceeded the amount allowed in z305_cash_limit - uses only O(pen) transactions and not T(ransferred)
- **6** - renew - check item requests before renewing.
- **6_a** - loan - check item for holds before loaning
- **7_a** - check item loan permission - check whether the item can be loaned, based on settings in column 6 in tab15.<lng>
- **7_b** - check item renew permission - check whether a loan on this item can be renewed, column 7 in tab15.<lng>.
- **7_c** - check item re-borrowing limit. Checks a restriction on re-borrowing the exact item within a specific time. The specific time is defined in tab15 col. 15 (where value 99 means no limitation)
- **8_a** - check loan - lost/claimed return
- **8_b** - check loan - letter number/letter date
- **9_a** - check overdues and fines - for ADM library, checks maximum overdues, recalled overdues and fines, based on tab_block_circ
- **9_b** - check overdues – sub-library. Checks maximum overdues and recalled overdues in a sub-library based on tab_block_circ
- **10_a** - same as 9_a except that all loans are taken into account - active, lost and claimed return
- **10_b** - same as 9_b except that all loans are taken into account - active, lost and claimed return
- **11_b** - check item due date relative to patron expiration date
- **12_a** - check if there is a match between the original ILL patron and the patron who is trying to loan the ILL item.

**Structure of the table:**
- Col. 1 defines when check is performed
- Col. 2 defines check procedure to perform
- Col. 3 Table consulted by checks 9a, 9b, 10a and 10b

**Example of the table:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>!1</td>
<td>!1</td>
<td>!1</td>
</tr>
<tr>
<td>!1</td>
<td>!1</td>
<td>!1</td>
<td>!1</td>
</tr>
<tr>
<td>!1</td>
<td>!1</td>
<td>!1</td>
<td>!1</td>
</tr>
<tr>
<td>!1</td>
<td>!1</td>
<td>!1</td>
<td>!1</td>
</tr>
</tbody>
</table>

| LOAN-GBL | check_circ_1_a |
| LOAN-GBL | check_circ_1_b |
| LOAN-GBL | check_circ_1_c_a |
| LOAN-GBL | check_circ_1_c_b |
| !LOAN-GBL | check_circ_9_a |
| LOAN-GBL | check_circ_10_a |
|!<    |             |
| !RENEW-GBL | check_circ_1_a |
| !RENEW-GBL | check_circ_9_a |
| RENEW-GBL | check_circ_10_a |
|</tab> block_circ_test
**tab_checksum**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Checksum programs

This table is used for defining checksum programs. The programs can be used for creation, validation, and correction of:
- item barcode,
- patron barcode,
- budget code prefix,
- vendor/supplier code prefix
- vendor/supplier additional-code.

**Conventions:**
- validate programs must be called checksum_valid_<>.cbl
- create programs must be called checksum_create_<>.cbl
- correct programs must be called checksum_correct_<>.cbl
- validate routine names must start with CSV-
- create routine names must start with CSCR-
- correct routine names must start with CSCO-

**Available programs are:**
- checksum_create_z30_by_seq
  creating item barcode, using UTIL G/2 last-barcode-number
- checksum_create_z30_by_doc_no
  creating item barcode, using Z30-REC-KEY (ADM record number + item sequence number)
- checksum_create_codabar
  creating codabar compatible item barcode
- checksum_create_z30_icl
  site-specific procedure for the Danish libraries, which uses barcode-sequence counter in UTIL G/2)
- checksum_valid_item_bar_uea and checksum_valid_bor_id_uea
  site-specific procedures for an application in the UK, for validation of item and patron barcodes
- checksum_valid_bar_newcast and checksum_valid_bar_newcast
  site-specific procedures for an application in the UK, for validation of item and patron barcodes
- checksum_valid_z30_bar_nlc
  site-specific procedure for an application in China for validation of item barcodes
- checksum_valid_vendor_abn
  performs validation of the ABN (Australian Business Number) on Vendor Additional Code (Z70-ADDITIONAL-VENDOR-CODE). If data is entered in the Vendor Additional Code field, it must match the ABN check.
- checksum_valid_code_prefix
  performs a check of the vendor code prefix and/or the budget code prefix. It works in conjunction with the prefixes defined in the tab_code_prefix table.
The instances for which the programs can be used are:

- **CSCR-Z30-BARCODE**: create item barcode
- **CSCR-Z30-BARCODE-ISS**: create issue-item barcode for items generated in prediction patterns
- **CSCR-CODA**: create item barcode
- **CSCO-VENDOR-ADD-CODE**: correct vendor ABN
- **CSV-Z30-BARCODE**: validate item barcode
- **CSV-VENDOR-ADD-CODE**: validate vendor ABN
- **CSV-LOAN-ITEM-BARCODE**: validate item barcode when loan is performed
- **CSV-LOAN-BOR-ID**: validate patron id when loan is performed
- **CSV-VENDOR-CODE**: validate vendor code prefix (using tab_code_prefix)
- **CSV-BUDGET-NUMBER**: validate budget code prefix (using tab_code_prefix)

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Routine name</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Program name</td>
</tr>
</tbody>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSV-VENDOR-ADD-CODE</td>
<td>checksum_valid_vendor_abn</td>
</tr>
<tr>
<td>CSV-VENDOR-ADD-CODE</td>
<td>checksum_correct_vendor_abn</td>
</tr>
<tr>
<td>CSV-LOAN-ITEM-BARCODE</td>
<td>checksum_valid_item_bar_uea</td>
</tr>
<tr>
<td>CSV-LOAN-ITEM-BARCODE</td>
<td>checksum_valid_bar_newcast</td>
</tr>
<tr>
<td>CSV-LOAN-BOR-ID</td>
<td>checksum_valid_bor_id_uea</td>
</tr>
<tr>
<td>CSV-LOAN-BOR-ID</td>
<td>checksum_valid_bar_newcast</td>
</tr>
<tr>
<td>CSV-VENDOR-CODE</td>
<td>checksum_valid_code_prefix</td>
</tr>
<tr>
<td>CSV-BUDGET-NUMBER</td>
<td>checksum_valid_code_prefix</td>
</tr>
<tr>
<td>CSV-Z30-BARCODE</td>
<td>checksum_valid_z30_bar_nlc</td>
</tr>
<tr>
<td>CSCR-Z30-BARCODE</td>
<td>checksum_create_z30_by_seq</td>
</tr>
</tbody>
</table>

**tab_code_prefix**

Location of the table: tab directory of the ADM library

Purpose of the table: Prefixes for budget, vendor and ILL supplier codes

This table builds a list of prefixes for budget code, vendor code and ILL supplier code.
The list is checked when tab_checksum is set to perform a check of vendor/supplier and budget code:
CSV-VENDOR-CODE,
CSV-BUDGET-CODE.

Structure of the table:
Example of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Language</th>
<th>col. 2</th>
<th>Day numeric code (00=Sun, 01=Mon, 02=Tue, 03=Wed, 04=Thu, 05=Fri, 06=Sat)</th>
<th>col. 3</th>
<th>Day abbreviation</th>
<th>col. 4</th>
<th>Day name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENG</td>
<td>Sun</td>
<td>00</td>
<td>Sun</td>
<td>Sun</td>
<td>Sun</td>
<td>Wed</td>
<td>Wed</td>
</tr>
<tr>
<td>ENG</td>
<td>Mon</td>
<td>01</td>
<td>Monday</td>
<td>Mon</td>
<td>Mon</td>
<td>Mon</td>
<td>Mon</td>
</tr>
<tr>
<td>ENG</td>
<td>Tue</td>
<td>02</td>
<td>Tuesday</td>
<td>Tue</td>
<td>Tue</td>
<td>Tue</td>
<td>Tue</td>
</tr>
<tr>
<td>ENG</td>
<td>Wed</td>
<td>03</td>
<td>Wednesday</td>
<td>Wed</td>
<td>Wed</td>
<td>Wed</td>
<td>Wed</td>
</tr>
</tbody>
</table>

### tab_days

**Location of the table:** alephe/tab directory

**Purpose of the table:** Day codes and abbreviations

This table allows for the definition of the days of the week in different languages. The table is used in the WEB OPAC for the short loan reservations.

**Structure of the table:**

- col. 1 Language
- col. 2 Day numeric code (00=Sun, 01=Mon, 02=Tue, 03=Wed, 04=Thu, 05=Fri, 06=Sat)
- col. 3 Day abbreviation
- col. 4 Day name

Example of the table:

```
!1 2 3 4
!!!-!!-!!!-!!!!!!!!!
ENG 00 Sun Sunday
ENG 01 Mon Monday
ENG 02 Tue Tuesday
ENG 03 Wed Wednesday
```

### tab_dbs.<lng>

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Configuration for Advanced Library Statistics (DBS)

This table is used to define the parameters that are required for the statistical evaluation according DBS (UTIL S/3). These statistics are used for sites in Germany.

**Structure of the table:**

- col. 1 Parameter
- col. 2 Value
col. 3  Description

Example of the table:

| ! | 1 | 2 | 3 |
|---------------------------------|
| !!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!! |
| REPORT-BEGIN 20000410           |
| REPORT-END 20011212             |
| CHECK-SL-CODE UEDUC Erziehung   |
| CHECK-SL-CODE UHLTH Gesundheitsw. |
| CHECK-SL-CODE ULAW Recht       |
| CHECK-EXT-BOR 01                |
| SUSPEND-PROCESS-STALL          |
| SUSPEND-Z30-MATERIAL ISSUE      |

tab_delinq.<lng>

Location of the table: tab directory of the ADM library

Purpose of the table: Block codes and restrictions

This table is used for two purposes:

- to define the effect that a block code on the patron's Global and Local records has on patron circulation privileges. In order to block a specific privilege, enter N in the appropriate column. Otherwise leave it blank.
- to set the default text description for a block code. When a block code is set in the GUI interface, if the text description field is left blank, the description in this table is automatically applied. Note that although the description field can hold up to 200 characters, the LOAN window displays only the first 40 characters, and only 100 characters are accepted from this table.

Note: Loan and Renewal in col. 4 appear as “not implemented” as these privileges are always blocked whenever there is a delinquency on the patron record.

Structure of the table:

Col. 1  Delinquency Code
Col. 2  Record type
  - 1=Patron (Global) (Z303)
  - 2=Patron (Local) (Z305)
Col. 3  Sub-library. Relevant only if col. 2 is type 2 (record type – patron).
  # for wild card can be used.
Col. 4  Patron permission
  - "N" to set to "no privilege",
  - Blank to set to privilege
  - a = Not Implemented (Z305-LOAN-PERMISSION)
• \( b = Z305\)-PHOTO-PERMISSION
• \( c = Z305\)-OVER-PERMISSION
• \( d = Z305\)-MULTI-HOLD
• \( e = Z305\)-LOAN-CHECK
• \( f = Z305\)-HOLD-PERMISSION
• \( g = \text{Not Implemented (}Z305\)-RENEW-PERMISSION\)
• \( h = Z305\)-RR-PERMISSION

Col. 5  ALPHA
Col. 6  Description (displays on patron card in WEB OPAC).

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>abcdefgh</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>1</td>
<td>L Patron has NO global blocks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>1</td>
<td>L Patron consistently returns books late</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>1</td>
<td>L Disruptive behavior in library</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>1</td>
<td>L Personal item left in library - contact CIRC desk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>1</td>
<td>L Patron to contact academic adviser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>2</td>
<td>L Patron has NO local blocks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2</td>
<td>L Patron is blocked from placing a hold request.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>2</td>
<td>L Patron is not allowed to place hold or photocopy request.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_events.<lng>**

**Location of the table:** tab directory of the library

**Purpose of the table:** Events definitions

This table defines which transactions will write a record on the Z35 events Oracle table. The transactions relate to:

• GUI/Web
• Circulation (loans, returns, holds, photocopy requests)
• GUI/Web OPAC (searching).

The event type identifier is written in Z35-EVENT-TYPE of the Z35 table.

**Structure of table:**

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Event Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Activate - Y/N</td>
</tr>
<tr>
<td></td>
<td>• Y = activate event type</td>
</tr>
<tr>
<td></td>
<td>• N = do not activate</td>
</tr>
<tr>
<td>col. 3</td>
<td>Event description.</td>
</tr>
</tbody>
</table>

Example of the ADM table:
Example of the BIB table:

```
!1 2 3
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
31 Y Creating New ILL Request

50 Y Simple Loan
51 Y Reading Room Loan
52 Y Offline Loan
53 Y Advance Booking Loan
54 Y Routing List Loan
55 Y ILL Loan
56 Y Transfer Loan
57 Y Remote Storage Loan
```

### `tab_exp_own.<lng>`

**Location of the table:** tab directory of the **ADM library**

**Purpose of the table:** defines the possible values for the OWN field

This table defines the possible values for the OWN field. It lists the possible values for the OWN field given to a staff user in the fields: Z66-USER-OWN-CREATE and Z66-USER-OWN-CHECK.

In the staff user form in the GUI, these 2 fields can be expanded. The values are taken from the table of the user library (Z66-USER-LIBRARY).

E.g. If modifying/adding a user of USM50, the expand of these fields will show the list of library USM50. The values in these fields will be checked against this table. If there is an incorrect value an error message will popup.

If it is a supervisor user (Z66-USER-LIBRARY="ADMIN"), then the expand list will be empty and no checks will be done for the fields. This is because ‘supervisor’ users are working on the management of the staff users and not working with the catalog, so these fields aren’t relevant for them.

**Col. 1** OWN description.

**Col. 2** OWN values which the user can have

Example of the table:

```
!1 2
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
10 Y Search Command
11 Y Search command beyond limit
12 Y Search command with zero hits
```
**tab_expand**

Location of the table: tab directory of the library

Purpose of the table: List of expand programs

This table allows for the defining of *expand* programs for merging linked records (from the same and different libraries) for display and indexing.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Col. 2</th>
<th>Col. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expand Menu</td>
<td>Expand procedure</td>
<td>Program arguments</td>
</tr>
</tbody>
</table>

**Note:** Column 3 is used for passing arguments to the expand-routines. For example, the `expand_doc_sort_field` gets as an argument from the 3rd column, the field+sub-field to sort:

| U39-DOC      | expand_doc_sort_field | CAT##,h                     |

This expand program will sort the CAT fields according to $$h (the hour).

In addition to listing the expand programs that could be active in a library, the table defines *where* in the library the expand will be active.

Expand Procedures:

For the full list of expand programs refer to ALEPH User Guide/ Indexing section / Expand routines, tables and Expanded fields.

However, note that

- `expand_doc_adm_bib`
- `expand_doc_adm_hol`
- `expand_doc_bib_hol`
- `expand_doc_hol_bib`

support the following options:

1) Parameters in column 3 of tab_expand can be either:
   a. a list of up to 10 comma-separated fields to include or exclude (to exclude, precede the list with a single minus (-) sign). For `expand_doc_adm_bib` and `expand_doc_hol_bib` it is a list of BIB fields; for `expand_doc_adm_hol` and `expand_doc_bib_hol` it is a list of HOL fields.
or

b. a merge-type to be used to merge BIB into ADM or HOL (for expand_doc_adm_bib and expand_doc_hol_bib), or merge HOL into ADM or BIB (for expand_doc_adm_hol and expand_doc_bib_hol).

2) BEFORE record2 is added or merged into record1 (for any expand_doc_lib1_lib2), it is itself expanded by using a special section in the format "LIB1-LIB2" in tab_expand of LIB2. For example, in the case of expand_doc_hol_bib, if the section "HOL-BIB" is defined in tab_expand of the BIB library, the BIB record is first expanded by using this section, and only then its fields are added to the expanded HOL record or merged into it (according to the content of column 3 of tab_expand for expand_doc_hol_bib, defined in the HOL library).

Note that if "MERGE-TYPE=" is used in column 3 of tab_expand for expand_doc_adm_hol or expand_doc_bib_hol, it is possible that more than one HOL record will be merged into the ADM or BIB record, e.g. for expand_doc_bib_hol, if more than one HOL record is linked to the BIB record, the first HOL record will be merged into the BIB record; then, the second HOL record will be merged into the already merged BIB record, etc.

In addition to the expand program, the table also defines the context wherein the expand will be operative:

<table>
<thead>
<tr>
<th>Code</th>
<th>Context</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>p_manage_02_b</td>
<td>create headings (service) internal program</td>
</tr>
<tr>
<td></td>
<td>update_acc</td>
<td></td>
</tr>
<tr>
<td>ACC-BRIEF</td>
<td>p_manage_35</td>
<td>create Z0101 brief records</td>
</tr>
<tr>
<td>BUF-950</td>
<td>www_f_build_950</td>
<td>Usage – SFX gateway</td>
</tr>
<tr>
<td>CREATE-Z13</td>
<td>p_manage_07_a</td>
<td>create short bib record (service) internal program</td>
</tr>
<tr>
<td></td>
<td>update_short_doc</td>
<td></td>
</tr>
<tr>
<td>E-DOC-&lt;format no.&gt;</td>
<td>edit_doc_xxx</td>
<td>Specific format record display</td>
</tr>
<tr>
<td>GUI-ACCREF</td>
<td>pc_sear_c1004</td>
<td>display of AUT record from BIB heading</td>
</tr>
<tr>
<td>GUI-BRIEF</td>
<td>pc_sear_c1005</td>
<td>Brief display/GUI SEARCH</td>
</tr>
<tr>
<td>GUI-DOC-D</td>
<td>pc_sear_c1001</td>
<td>full display GUI SEARCH</td>
</tr>
<tr>
<td>GUI-DOC-P</td>
<td>pc_sear_c1013</td>
<td>full print from GUI SEARCH</td>
</tr>
<tr>
<td>HOLDING</td>
<td>pc_sear_c1024</td>
<td>display of item list</td>
</tr>
<tr>
<td></td>
<td>pc_sear_c1025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>get_bib_line</td>
<td></td>
</tr>
<tr>
<td>HOL-LOC</td>
<td>www_f_edit_hol</td>
<td>for HOL library only. Use with expand_doc_hol_loc_1_a</td>
</tr>
<tr>
<td>INDEX</td>
<td>p_manage_05_a</td>
<td>create indexes (service) internal program</td>
</tr>
<tr>
<td></td>
<td>update_ind</td>
<td></td>
</tr>
<tr>
<td>PRINT-CAT</td>
<td>p_print_04_a</td>
<td>print catalog (service)</td>
</tr>
<tr>
<td>PRINT-COL</td>
<td>p_print_08_a</td>
<td>columnar printout (service)</td>
</tr>
<tr>
<td>PRINT-CUST</td>
<td>p_print_01_a</td>
<td>custom format (service)</td>
</tr>
<tr>
<td>PRINT-REC</td>
<td>p_print_03_a</td>
<td>download records (service)</td>
</tr>
<tr>
<td>Code</td>
<td>Context</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>RET</td>
<td>p_ret_01</td>
<td>- retrieval (service interface)</td>
</tr>
<tr>
<td></td>
<td>p_ret_21</td>
<td>- sort (service interface)</td>
</tr>
<tr>
<td>SORT-DOC</td>
<td>p_manage_27_a</td>
<td>create sort keys (service)</td>
</tr>
<tr>
<td></td>
<td>update_sort_doc</td>
<td>internal program</td>
</tr>
<tr>
<td>TAB33-DOC</td>
<td>pc_com_c0123</td>
<td>Navigation Window (GUI)</td>
</tr>
<tr>
<td></td>
<td>pc_com_tab33_heading</td>
<td>Format a doc according to tab33</td>
</tr>
<tr>
<td>U39-DOC</td>
<td>u39-doc</td>
<td>record display (UTIL F/4)</td>
</tr>
<tr>
<td>UE-08</td>
<td>ue_08</td>
<td>expands doc in the AUT library for ue-08 procedures</td>
</tr>
<tr>
<td>UPDATE-Z103</td>
<td>p_manage_12_a</td>
<td>create links (service)</td>
</tr>
<tr>
<td>WEB-ACCREF</td>
<td>www_f_accref</td>
<td>display of AUT record from BIB heading (defined in AUT library)</td>
</tr>
<tr>
<td>WEB-BRIEF</td>
<td>www_f_short</td>
<td>brief display/Web OPAC</td>
</tr>
<tr>
<td></td>
<td>www_f_booking</td>
<td>[not yet implemented]</td>
</tr>
<tr>
<td>WEB-FULL</td>
<td>www_f_edit_doc</td>
<td>full display/Web OPAC</td>
</tr>
<tr>
<td>WEB-FULL-1</td>
<td>www_f_edit_doc</td>
<td>full display/Web OPAC format 01</td>
</tr>
<tr>
<td>WEB-MAIL</td>
<td>www_f_basket_mail</td>
<td>full print (mail) from Web/OPAC</td>
</tr>
<tr>
<td></td>
<td>www_f_full_mail</td>
<td></td>
</tr>
<tr>
<td></td>
<td>www_f_short_mail_print</td>
<td></td>
</tr>
<tr>
<td>WEB-SAVE</td>
<td>www_f_save_set_print</td>
<td>full print (save) from Web/OPAC</td>
</tr>
<tr>
<td>WEB-SCNIND</td>
<td>www_f_scan_ind</td>
<td>display titles in scan list for a direct index</td>
</tr>
<tr>
<td>WORD</td>
<td>p_manage_01_a</td>
<td>create word indexing (Service)</td>
</tr>
<tr>
<td></td>
<td>ue_01_w</td>
<td>internal program</td>
</tr>
<tr>
<td>Z39_HOL</td>
<td>z39_server</td>
<td>(present service, holdings schema)</td>
</tr>
<tr>
<td>Z39_SERVER</td>
<td>z39_server</td>
<td>(present service)</td>
</tr>
</tbody>
</table>

Example of the table:

```
U39-DOC expand_doc_join
!U39-DOC expand_doc_bib_hol_usm
!U39-DOC expand_doc_type tab_type_config

WORD expand_doc_bib_loc_usm
WORD expand_doc_bib_accref
WORD expand_doc_type tab_type_config.eng
WORD expand_doc_fix_abbreviation

!ACC-BRIEF expand_doc_type tab_type_config
ACC-BRIEF expand_doc_split_sub1
```

**Note:**
The expand procedure `expand_doc_join` now incorporates the former procedures `expand_doc_join` and `expand_doc_join_all`. If you use the Parameter TYPE=ALL in column 3, the functionality equals the former functionality of `expand_doc_join_all`. Leaving column 3 blank, will provide the functionality of the former procedure `expand_doc_join`.

© Ex Libris Ltd., 2004
Updated: 3/21/2004
Page 127 of 336
Example of the table with the functionality of expand_doc_join_all:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
| !!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!>

U39-DOC expand_doc_join TYPE=ALL

tab_expand_duplicate_field

Location of the table: tab directory of the library

Purpose of the table: Field duplication

The tab_expand_duplicate_field table is required for expand_doc_duplicate_field, which duplicates a field assigning a new field tag + indicators.

This expand procedure and table was added in order to overcome the problem created when using expand_doc_split, which does not retain the source field. When _split is based on a field created by _duplicate_field, the source field is retained.

Structure of the table:
- Col. 1 Input Field tag + indicators
- Col. 2 Output Field tag + indicators

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!-!!!!!!-!!!!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1### A99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4### A99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>260## IMP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

tab_expand_external

Location of the table: tab directory of the library

Purpose of the table: Splits external fields

This table defines definitions for expand_doc_split_external, that splits external fields (856,505 etc) when finding multiple occurrences of a designated subfield.

The split is done according to the subfield specified in col.2.

Each new field includes a single occurrence of the subfield specified, and all the other subfields. New lines created by this process have the original tag, and the original field is suppressed.
If the field does not contain multiple occurrences of the subfield, the field remains as is.

For example:

856## U

Original data:
8564  L $$4$$uhttp://authorities.loc.gov$$uhttp://dublincore.org
Output:
8564  L $$4$$uhttp://authorities.loc.gov
8564  L $$4$$uhttp://dublincore.org

505## U

Original data
5054  L $$a505 FORMATTED CONTENTS NOTE
     $$uhttp://lcweb.loc.gov/marc/bibliographic/ecbdnot2.html#mrcb555
     $$uhttp://www.loc.gov/standards/mets
Output:
5054  L $$a505 FORMATTED CONTENTS NOTE
     $$uhttp://lcweb.loc.gov/marc/bibliographic/ecbdnot2.html#mrcb555
5054  L $$a505 FORMATTED CONTENTS NOTE
     $$uhttp://www.loc.gov/standards/mets

Structure of the table:
Col. 1  Input Field tag + indicators
Col. 2  Subfield to identify

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>856## u</td>
<td>505## u</td>
<td>514## u</td>
</tr>
</tbody>
</table>

**tab_expand_extract**

Location of the table: tab directory of the library

Purpose of the table: Extraction of sub-fields for indexing

The table defines extraction of sub-fields (in a virtual manner) for indexing. This allows for separate indexing of every occurrence of a sub-field.

Structure of the table:
Col. 1  tag – wild card may be used
Col. 2  sub field
Col. 3  new field that will be created and that will be used for indexing (using tab11_acc)
Col. 4  Number of subfield occurrences for which the new virtual field is created. For example, it is possible to define that only the first occurrence of subfield $y$ in the 650 field should be used for the creation of the new field. Default is 9.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>010##</td>
<td>a</td>
<td>a010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>010##</td>
<td>z</td>
<td>z010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>020##</td>
<td>a</td>
<td>a020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>020##</td>
<td>z</td>
<td>z020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>022##</td>
<td>a</td>
<td>z022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>022##</td>
<td>y</td>
<td>y022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>022##</td>
<td>z</td>
<td>z022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>035##</td>
<td>a</td>
<td>a035</td>
<td></td>
<td></td>
</tr>
<tr>
<td>035##</td>
<td>z</td>
<td>z035</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_expand_join**

Location of the table: tab directory of the library

Purpose of the table: Creation of a virtual field from two or more fields

The system can add virtual fields to a bibliographic record, through the expand_doc_join procedure (as defined in tab_expand). The procedure uses the tab_expand_join table to create the virtual fields, by joining up to 5 fields together in a single new field.

Note that for non MAB formats, the expand_doc_join will create a virtual field only if at least two of the fields being joined are present. To always create a new virtual field, use expand_doc_join_simple and its accompanying table, tab_expand_join_simple.

For MAB installations, the new virtual field will **always** be created even if only one of the fields being joined is present.

If there are multiple occurrences of the fields, joining is done in pairs if expand_doc_join is used. For example:

- $a1$ $a2$ $b1$ $b2$ $b3$ will create $a1+b1$ and $a2+b2$.
- expand_doc_join (option TYPE=ALL) joins every occurrence of the fields. For example:
  - $a1$ $a2$ $b1$ $b2$ $b3$ will create $a1 + b1$, $a1 + b2$, $a1 + b3$, $a2 + b1$, $a2 + b2$ and $a2 + b3$.

Expand_doc_join is useful for indexing, while expand_join_simple is useful for creating virtual fields for display - for example a field which is title plus edition statement [245 and 250].

Note that expand_doc_join strips non-filing initial text (using the non-filing indicator of the field(s) being joined) when the virtual field is created.
Columns 3/4, 6/7, etc. are used to define which sub-field to take and to which sub-field to assign it. If the columns are left blank, the entire field is taken, and all sub-fields are retained as is.

Differences between tab_expand_join and tab_expand_join_simple:

<table>
<thead>
<tr>
<th>TAB_EXPAND_JOIN</th>
<th>TAB_EXPAND_JOIN_SIMPLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtual field will be created only if all defined fields are present</td>
<td>new virtual field is always created</td>
</tr>
<tr>
<td>up to five fields can be defined</td>
<td>two fields are defined</td>
</tr>
<tr>
<td>useful for indexing</td>
<td>useful for display purposes</td>
</tr>
<tr>
<td>strips non-filing initial text (using the non-filing indicator of the field(s) being joined) when the virtual field is created</td>
<td>does not strip initial non-filing text from fields before joining them</td>
</tr>
</tbody>
</table>

Structure of the table:

Col. 1 tag + indicator. **Note:** in order to use the new field it must be defined in the relevant tables - tab01.<lng>, tab11_acc, and edit_doc_999.<lng>

Col. 2 first tag for building

Col. 3 sub-fields to take for match

If this column is empty, all subfields are taken. If you want a selection of subfields, enter the subfield codes, or enter ",-" followed by the codes of the subfields that should not be taken. If the field does not have a subfield code (e.g. 001), this column should be left empty.

Col. 4 new sub-field code. If this column is left empty, the original subfield codes are retained.

There are 3 ways to use this column:

- Take few (or one) sub-fields from the text and substitute them with exact number of new sub-fields. e.g. take abc and change to xyz
- Take all text (or few sub-fields) and change them to one new sub-field. e.g. take abcd and change to x
- Take all BUT few sub-fields (- sign) and change them to one sub-field. e.g. take -abcd and change to x

Cols. 5-16 second-fifth tags and sub-fields for building (as cols. 3-4)

Example of the table:
tab_expand_join_simple

Location of the table: tab directory of the library

Purpose of the table: Joining fields for indexing

The system can add virtual fields to a bibliographic record, through the expand_doc_join_simple procedure. The procedure uses this table to create the virtual fields, by joining multiple occurrences of two fields in a single new field.

Note that unlike expand_doc_join, which creates a virtual field only if all the fields being joined are present, expand_doc_join_simple always creates a new virtual field. For MAB installations expand_doc_join or expand_doc_join_all can be used instead.

Expand_join_simple is useful for creating virtual fields for display - for example a field which is title plus edition statement [245 and 250], whereas expand_doc_join is useful for indexing.

Note that expand_doc_join_simple does not strip initial non-filing text from fields before joining them, only expand_doc_join does this.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>tag + indicator</th>
<th>Note: in order to use the new field it must be defined in the relevant tables - tab01.&lt;lng&gt;, tab11_acc, and edit_doc_999.&lt;lng&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>first tag for building</td>
<td></td>
</tr>
<tr>
<td>Col. 3</td>
<td>first tag occurrence (nn for index, AA for All)</td>
<td></td>
</tr>
<tr>
<td>Col. 4</td>
<td>sub-field/s to take for match</td>
<td></td>
</tr>
<tr>
<td>Col. 5</td>
<td>new sub-field code</td>
<td></td>
</tr>
<tr>
<td>Col. 6</td>
<td>second tag for building</td>
<td></td>
</tr>
<tr>
<td>Col. 7</td>
<td>second tag occurrence</td>
<td></td>
</tr>
<tr>
<td>Col. 8</td>
<td>sub-field/s to take for match</td>
<td></td>
</tr>
<tr>
<td>Col. 9</td>
<td>new sub-field code</td>
<td></td>
</tr>
</tbody>
</table>

Example of the table:

| TST01 100## ab 1 100## cd 2 245## -b t |
| ATT01 100## a 240## abcd t |
| ATT02 100## a 245## abcd t |
| AU100 100## -e46 240## -h6 t |
| AU110 110## -e46 240## -h6 |
| AU111 111## -46 240## -h6 |
| AT100 100## -e46 245## -ch6 |
| AT110 110## -e46 245## -ch6 |
| AT111 111## -46 245## -ch6 |
tab_expand_sort

Location of the table: tab directory of the library

Purpose of the table: Sort sub-fields for expand

tab_expand_sort can be used in tab_expand in order to set the sort order of subfields of a field.

Structure of the table:
  Col. 1  Tag + indicators
  Col. 2  Sub-field/s to take for sort

Example of the table:

```
! 1 2
!!!!!!-!!!!!!!!!!!!!!!!!!!!
260## cba
```

tab_expand_split

Location of the table: tab directory of the library

Purpose of the table: Split field definitions

The tab_expand_split table is required for expand_doc_split, which splits a field into separate parts, splitting on the subfield specified in col.2. The split will occur on every occurrence of the subfield, thereby creating multiple occurrences of the field.

The "split" includes all the data up to the subfield, and all the data from the subfield, up to the next occurrence of the subfield, or to the end. For example:

```
700## t  A700  T700
causes
7001  $$aMendelssohn-Bartholdy, Felix,$$d1809-1847.
    $$tLider ohne Worte,$$mpiano,$$nop.62.$$nNo. 6.
    $$pFruhlings lied.$$f1970
    to become
A700  L $$aMendelssohn-Bartholdy, Felix,$$d1809-1847
T700  L $$tLider ohne Worte,$$mpiano,$$nop.62.$$nNo. 6.
    $$pFruhlings lied.$$f1970
```
The above example illustrates how the split facilitates correct indexing of author-title added entries.

If the field does not contain the subfield that is defined in column 2, the field is not treated and remains as is.

Structure of the table:
- Col. 1: Input tag + indicators
- Col. 2: Subfield to split on
- Col. 3: Output tag + indicators for text up to the breaking subfield
- Col. 4: Output tag + indicators for text after the breaking subfield

Example of the table:

```
! 1 2 3 4
!----!----!----!----!
700## t A700 T700
036## a
```

**tab_export**

Location of the table: alephe/tab directory

Purpose of the table: Retrieval definitions for p-export-01

Structure of the table:
- Col. 1: The target of the export file. It may be a single ALEPH sub-library or an artificial target name that represents multiple sub-libraries.
- Col. 2: Format
  - Specifies the format of the export file.
    - If MARC or MARC2 are specified, the resulting export file will be in standard MARC Communications Format (MARC - numeric fields only, MARC2 - the MARC format will include alphanumeric fields).
    - If ALEPH is specified, the file will be in ALEPH Sequential Format.
- Col. 3: Zip Flag
  - Y: the file will be zipped.
  - N: the file will not be zipped
- Col. 4: Retrieval Parameters
  - This column is composed of 8 comma-delimited parameters.
    - They represent 2 sets of parameters:
      - tag, sub-field, "from" value, "up to" value,
      - tag, sub-field, "from" value, "up to" value.
  - These two sets are used to determine which holdings data should be included in any given execution of the export process.
  - Parameters example:
The above line instructs the system to extract all holdings data that has:

- an 'XTP' tag with sub-field $a$ that contains the value 'Y'
- and also and 852 tag with sub-field $b$ that contains a value between ULINC and ULINC (i.e. only ULINC satisfies the condition).

**col. 5** Indicates the name of the directory into which the export file is to be placed.

**Example of the table:**

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ULINC MARC N XPT,a,Y,Y,852,b,ULINC,ULINC /aleph/u01/outdir</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MANY MARC Y XPT,a,Y,Y,852,b,GROUP,GROUP /aleph/u01/outdir</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UARCV MARC Y 852,b,UARCV,UARCV,,,, /aleph/u01/outdir</td>
</tr>
</tbody>
</table>

**tab_export_03**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Smart barcode export configuration

**Structure of the table:**

- **Col. 1** Size of output-field. If size is zero, it will combine with the next non-zero sized field.
- **Col. 2** Source of output field - can be a
  - tag name,
  - ERROR-CODE,
  - ERROR-TEXT,
  - or a fixed text surrounded by quotes – e.g. "fixed text" or "copy no." etc.
- **Col. 3** Subfields to be taken (if col.2 is tag name)
- **Col. 4** Alternative tag
- **Col. 5** Alternative subfields

**Example of the table:**

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00 Z30-BARCODE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>08 &quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00 Z30-SUB-LIBRARY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>00 &quot;,&quot;</td>
</tr>
</tbody>
</table>
**tab_fast_patron_register**

Location of the table: tab directory of the **USR** library

Purpose of the table: Defining default values for fast patron registration

Default values to be used when registering a new patron in the Circulation GUI via the "Patron Registration Form".

The lines in this table are in the format "FIELD=VALUE". FIELD is up to 30 characters long; VALUE is up to 20 characters long.

The following fields can be assigned default values from this table:

- **Z303-ILL-TOTAL-LIMIT**: 4 digits
- **Z303-ILL-ACTIVE-LIMIT**: 4 digits
- **Z303-EXPORT-CONSENT**: 1 character - "Y" or "N"
- **Z303-SEND-ALL-LETTERS**: 1 character - "Y" or "N"
- **Z305-BOR-TYPE**: 2 characters

**Z304-DATE-TO**: 8 digits; optional TEXT (1 character) - {Y (Year) / M (Month) / D (Day)}

**Z304-DATE-TO** relates to address of type "01" and can be either a specific date (8 digits only, e.g. Z304-DATE-TO=20041231), or 8 digits with an upper-case letter - Y (Year), M (Month) or D (Day), such that Z304-DATE-TO will be calculated by adding the specified number of years/months or days to the current date, e.g. Z304-DATE-TO=00000002Y.

This means that 2 years will be added to the current date and stored in the field Z304-DATE-TO.

Example of the table:

```
Z303-ILL-TOTAL-LIMIT=0555
Z303-ILL-ACTIVE-LIMIT=0444
Z303-EXPORT-CONSENT=N
Z303-SEND-ALL-LETTERS=N
Z304-DATE-TO=00000002Y
Z305-BOR-TYPE=DR
```
Purpose of the table: Definition of filing and normalization routines

This table is used for defining filing and normalization procedures that are used when building:

- Headings (Z01),
- Index (Z11) entries, and
- Sort keys (Z101).
- Brief Records (Z0101)

If a change is made to a filing procedure after an index has built, the index must be re-built.

Filing procedures for Z01 Headings have three sub-sections, defined in column 2 as "D", "N" or "F".

- **D** lines are used when Display Text is generated
- **N** lines are used to create Normalized text. It is based on the Display text, and is used by the system when deciding whether a heading (Z01) is new, or whether it matches an already existing heading.
- **F** lines are used to create Filing text. It is based on the Normalized text

Normalization is based on NACO Normalization Rules, and the procedures that make up this normalization are:

- comma
- del_sub_field_code
- compress (with set of punctuation)
- to_blank (with set of punctuation)
- char_conv (with NACO change case values)
- pack_spaces

The filing procedure identifier for Z01 and Z11 is set in col. 5 of tab00.<lng>. The filing procedure identifier for Z101 is set in col. 3 of tab01.<lng>. This setting is also used for the Service p-ret-21 (Sort Catalog Records) and p-ret-01 (Retrieve Catalog Records), as well as for FILTER in the OPAC (i.e. checking records in a set for presence of particular text).

If no procedure is listed in col. 3 of tab01.<lng>, the system defaults to procedure 99. Therefore, the table should always include procedure 99.

The table is also used for the Services Shelf Reading Reports (p-item-04, -05 and -06), using the location type.

Following are ALEPH's reserved routines:

- 90 - parsing FIND query in OPAC
- 97 - Brief Jump to text
- 98 - Z0101 (in BIB libraries)
- 98 - Search bibliographic records from authority library (in AUT libraries)
- 99 - Default for sort keys (Z101)

The table is limited to 5000 lines.

A single filing procedure is made up of a number of individual procedures, which are explained below. Some of the procedures are used only for normalization, but the filing text is based on the normalized text, and therefore includes the procedure's outcome.
Note that the order in which the procedures is listed is important. Text that has been manipulated in "N" lines is the basis for the "F" lines, and the order must take this into account.

For example,

- **numbers** (which compresses comma or dot in numbers to change 100,000 and 100.000 to 100000) must come before listing comma or dot in change-to-blank.
- When using both **numbers** and **expand_num** in the same section, **numbers** must precede **expand_num**.
- **suppress** (which suppresses text enclosed within "<...>") which must not be preceded by a procedure (such as **to_blank** or **compress**) if the procedure removes the <> signs.
- **non-filing** deletes characters, counting from the start of the field, as specified in the non-filing indicator. Therefore, characters must not be compressed before non-filing. For example, if the apostrophe is compressed before non-filing, the title “L'amour” will file as "mour", since the non-filing indicator “2” will remove the remaining La.

In addition, note that the routines defined in the tab_filing table require an 'F' section, therefore even if this section is not needed (because the 'N' section suffices), an 'F' line still needs to be present with "no" as the filing procedure.

For some of the procedures, characters to be considered are defined in column 4 (e.g. in **to_blank**, compress and **end_punctuation**). The character can be keyboard input, or can be in Unicode notation, by entering U+<hexa value> (e.g. U+002E)

**List of filing procedures:**

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Action</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbreviation</td>
<td>compress a dot between single characters. Note that This procedure is limited to characters in the 7-bit ASCII range</td>
<td>L.B.M. changes to IBM</td>
</tr>
<tr>
<td>bbk</td>
<td>special procedure for Russian filing standards. Sorting sequence is special characters, then Cyrillic then Latin, then numbers.</td>
<td></td>
</tr>
<tr>
<td>char_conv</td>
<td>perform character conversion procedure according to the procedure name listed in col. 4. This name must match procedure identification in col. 1 of /alephe/unicode /tab_character_conversion_line</td>
<td></td>
</tr>
<tr>
<td>cjk_pinyin</td>
<td>adds ! before each cjk &quot;character&quot;, translates the characters to pinyin, using the Z114 table, and adds the Unicode value in decimal notation. The ! causes the pinyin filing-text to be sequenced separately from regular Latin characters.</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Action</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cjk_stroke</td>
<td>same as cjk_pinyin, except that each character is translated to stroke value, using the Z114 table.</td>
<td></td>
</tr>
<tr>
<td>chi_pinyin</td>
<td>translates each character to pinyin, using the Z114 table, and adds the Unicode value for each character. Unicode is added in order to differentiate between different characters that have the same pinyin value. Because the pinyin filing-text will be sequenced together with regular Latin characters, this routine should be used for browse lists that use the language code from 008 to create separate browse lists (e.g. AUTC).</td>
<td></td>
</tr>
<tr>
<td>chi_stroke</td>
<td>same as chi_pinyin, except that each character is translated to stroke value, using the Z114 table.</td>
<td></td>
</tr>
<tr>
<td>comma</td>
<td>all commas are turned into blank except for the first comma in sub-field &quot;a&quot; (used for normalization).</td>
<td></td>
</tr>
<tr>
<td>compress</td>
<td>compress (i.e. strip) the characters listed in col. 4. The characters can be keyboard input, or in Unicode notation.</td>
<td>the full stop (period) can be input in column 4 as . or as U+002E.</td>
</tr>
<tr>
<td>compress_blank</td>
<td>delete blanks</td>
<td></td>
</tr>
<tr>
<td>compress_perblnk</td>
<td>compresses a blank when it is preceded by a period (&quot;.&quot;).This is needed for the filing of certain LC Call Numbers.</td>
<td></td>
</tr>
<tr>
<td>cut_prefix</td>
<td>removes a string at the start of a line that is contained in parentheses.</td>
<td>(OcoLC)12345 files as 12345</td>
</tr>
<tr>
<td>del_lead_space</td>
<td>this routine deletes leading spaces after a subfield. It should be placed before del_subfield, since it will not be active if there is no subfield. This solves the problem of unwanted spaces left by &quot;to_blank&quot; (e.g. &quot;$b(Abc)&quot; which becomes &quot;$b Abc &quot;, or spaces created by char_conv).</td>
<td></td>
</tr>
<tr>
<td>del_sub-field</td>
<td>delete sub-field sign</td>
<td>$$x</td>
</tr>
<tr>
<td>del_sub_field_code</td>
<td>&quot;$x&quot; is retained, but the sub-field code is replaced by &quot;.&quot;. This is used for normalization, so that headings will match when the sub-field content is the same, even if the sub-field codes are different.</td>
<td></td>
</tr>
<tr>
<td>dewey_call_no:</td>
<td>special procedure for correct sequencing of Dewey call numbers</td>
<td></td>
</tr>
<tr>
<td>end_punctuation</td>
<td>deletes the characters listed in col.4</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Action</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>expand_num</td>
<td>expand number (add leading zeroes to fill numbers to 7 digits, for numeric filing). It is possible to use the parameters column of the table (column 4) to specify the subfields in which the procedure should be applied. Use the &quot;-&quot; to specify that all subfields, except those listed in the parameters column should be expanded. If the column is left blank, then numbers are expanded in all subfields.</td>
<td>100 changes to 0000100</td>
</tr>
<tr>
<td>end_punctuation</td>
<td>deletes the characters listed in col.4 when it is the last character in the heading. Used to remove / : = etc. at end of title, etc. Intended for USAGE type &quot;D&quot; for Z01-DISPLAY-TEXT. The characters can be keyboard input, or in Unicode notation. Removes end punctuation only from the end of line, (if it exists).</td>
<td>the full stop (period) can be input in column 4 as . or as U+002E.</td>
</tr>
<tr>
<td>end_sub_punctuation</td>
<td>deletes the characters listed in col.4 when it is the last character of a subfield</td>
<td></td>
</tr>
<tr>
<td>get_sub-fields</td>
<td>use only the sub-fields, or subtract some using &quot;-&quot; as listed in col. 4</td>
<td></td>
</tr>
<tr>
<td>get_subfields_order</td>
<td>similar to the get_subfields routine except that it retains the order of the subfields specified in column 4 of the table</td>
<td></td>
</tr>
<tr>
<td>icelandic_name</td>
<td>changes the order of the subfields 7 and 1, placing subfield 7 after subfield 1. Intended for sorting OPAC browse lists.</td>
<td>Display text: $SaAlexander $7Alfred $1Jonsson$194 3- Becomes Filing text: Alexander Jonsson Alfred 1943-</td>
</tr>
<tr>
<td>icelandic_z303_namee.g.</td>
<td>this procedure is used for filing patron names in the Icelandic manner. If the name has more than 2 words, the last word in the name is positioned after the first word.</td>
<td>Hulda Maria Einarsdottir -&gt; Hulda Einarsdottir Maria</td>
</tr>
<tr>
<td>issn/isbn</td>
<td>These routines test the validity of the input according to the same checks done for issn/isbn in cataloging and then</td>
<td>If 0001-9054 is a valid isbn then 00019054 will be</td>
</tr>
<tr>
<td>Procedure</td>
<td>Action</td>
<td>Example</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>return only the valid data.</td>
<td>Note that with this option we can still have the brackets in the &quot;display-text&quot; but they won't &quot;participate&quot; in the matching procedure.</td>
<td>indexed. If the input is invalid, then the filing routine will return the input.</td>
</tr>
<tr>
<td>jpn</td>
<td>translates each character to the decimal value of the Unicode character. This causes the filing-text to be sequenced together with regular characters. Therefore, this routine should be used for browse lists that use the language code from 008 to separate by language, and are separate for &quot;jpn&quot; (e.g. AUTJ).</td>
<td></td>
</tr>
<tr>
<td>kor</td>
<td>translates each character to the decimal value of the Unicode character. This causes the filing-text to be sequenced together with regular characters. Therefore, this routine should be used for browse lists that use the language code from 008 to separate by language, and are separate for &quot;kor&quot; (e.g. AUTK).</td>
<td></td>
</tr>
<tr>
<td>lc_call_no</td>
<td>special procedure for correct sequencing of LC call numbers. Note that this routine adds the following three characters to the index records it creates: ! &quot; # For this reason, you cannot have a to_blank or compress line which includes these characters after the lc_call_no line. In addition, note that this procedure is complete within itself, and does not require additional treatment. However, in order to facilitate searching, it is recommend that del_subfield be added.</td>
<td></td>
</tr>
<tr>
<td>mc_to_mac</td>
<td>change initial mc to mac</td>
<td>McKay files as mackay</td>
</tr>
<tr>
<td>no</td>
<td>this routine is used when the procedure defined in the table does not contain an 'F' section. The procedures defined in this table require an 'F' section, therefore even if this section is not needed (because the 'N' section suffices), an 'F' line still needs to be present with &quot;no&quot; as the filing procedure.</td>
<td></td>
</tr>
<tr>
<td>non_filing</td>
<td>drop initial text using non-filing indicator. This procedure is relevant for the hobbit files as hobbit</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Action</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>only building headings (Z01).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non_numeric</td>
<td>delete non-numeric characters</td>
<td>c1965 changes to 1965</td>
</tr>
<tr>
<td>none</td>
<td>data remains as is</td>
<td></td>
</tr>
<tr>
<td>norm_author</td>
<td>intended for normalizing name entries in a merged index. Retains first word and first character of second word, removing punctuation.</td>
<td>e.g. $$aSmith, John, $$d1923 is changed to $$aSmith J</td>
</tr>
<tr>
<td>numbers</td>
<td>compress a comma and a dot between numbers</td>
<td>2,153 changes to 2153</td>
</tr>
<tr>
<td>pack_spaces</td>
<td>compresses all multiple spaces to a single space</td>
<td></td>
</tr>
<tr>
<td>subfield_mab</td>
<td>intended for filing of headings which are based on MAB-authority information. In order to exclude the identification number from sorting, the procedure adds three blanks at the beginning of each subfield, from the second subfield on, and adds four blanks to the beginning of $$9. Subfield codes are removed.</td>
<td></td>
</tr>
<tr>
<td>suppress</td>
<td>suppress using)&lt;&lt;...&gt;&gt; (drop all text contained within the signs, and the signs themselves). &lt;&lt; and &gt;&gt; are the default delimiters (when there are no parameters). Parameters can be &quot;88-89&quot; and/or &quot;&lt;&lt;&lt;&lt;&quot; (comma-delimited), e.g. &quot;88-89&quot;, &quot;88-89,&lt;&lt;&lt;&lt;&quot;, &quot;&lt;&lt;&lt;&lt;,88-89&quot; or &quot;&lt;&lt;&lt;&lt;&quot;. The parameter &quot;88-89&quot; means that the control characters U+0088 and U+0089 will be used instead of &lt;&lt; and &gt;&gt;. The parameter &quot;&lt;&lt;&lt;&lt;&quot; is the same as the default. If BOTH parameters are specified, the input text will undergo suppression TWICE: once with &lt;&lt; and &gt;&gt; as delimiters, and again - with U+0088 and U+0089 as delimiters.</td>
<td></td>
</tr>
<tr>
<td>to_blank</td>
<td>change characters listed in col. 4 to blank. The characters can be keyboard input, or in Unicode notation. Always define &quot;pack_spaces&quot; after &quot;to_blank&quot;, in order to compress resulting multiple spaces into one space</td>
<td>the full stop (period) can be input in column 4 as . or as U+002E.</td>
</tr>
<tr>
<td>to_lower</td>
<td>to_lower. Uses the table alephe/unicode/unicode_case.</td>
<td>MARC to marc</td>
</tr>
<tr>
<td>to_carat</td>
<td>Change sub-field sign to ^^</td>
<td>for hierarchical sorting of</td>
</tr>
<tr>
<td>Procedure</td>
<td>Action</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>unicode</td>
<td>translates each character to the decimal value of the unicode character. This causes the filing-text to be sequenced together with regular characters. This routine is the same as kor and jpn.</td>
<td></td>
</tr>
<tr>
<td>widener_call_no</td>
<td>this routine is used to file Widener Call Numbers. It separates the call number into seven fixed-size sections</td>
<td></td>
</tr>
<tr>
<td>year_uu</td>
<td>This routine replaces &quot;u&quot; with zero (&quot;0&quot;) in &quot;doubtful&quot; year formats, e.g. 19uu, 197u. It should be activated BEFORE the routine &quot;expand_num&quot;</td>
<td>19uu becomes 1900 197u becomes 1970</td>
</tr>
</tbody>
</table>

**Structure of table:**

<table>
<thead>
<tr>
<th>col. 1</th>
<th>ID - Two-digit identifier of the filing routine. This identifier is used in column 5 of tab00.&lt;lng&gt; and column 3 of tab01.&lt;lng&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Usage</td>
</tr>
<tr>
<td></td>
<td>Relevant only for headings (Z01)</td>
</tr>
<tr>
<td></td>
<td>D - use when creating Z01-DISPLAY-TEXT</td>
</tr>
<tr>
<td></td>
<td>N - use when creating Z01-NORMALIZED-TEXT</td>
</tr>
<tr>
<td></td>
<td>F or SPACE - use when creating Z01-FILING-TEXT</td>
</tr>
<tr>
<td>col. 3</td>
<td>Name of the filing procedure</td>
</tr>
<tr>
<td>col. 4</td>
<td>Parameters for the filing procedure (when relevant)</td>
</tr>
</tbody>
</table>

**Special note:** non-filing should not be used for display and normalized if used for filing. See example of correct usage below (procedure 11):

**Example of the table:**

```
!1 2 3 4
!!-!-!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
01 D end_punctuation ;,=/.
01 N compress '[]|
!*rem use following line if you use <<...>>
!01 N to_blank !"()-{};:.?/\@*%=^_`~
01 N to_blank !"()-<>;:.?/\@*%=^_`~
01 N comma
01 N icelandic_name
01 N del_subfield_code
01 N char_conv FILING-KEY-10
01 N del_lead_space
01 N to_lower
01 N pack_spaces
!01 N norm_author
01 F del_subfield
```
**tab_filing_call_no**

Location of the table: tab directory of the ADM library

Purpose of the table: Defines filing routines for items

This table defines filing routines for items. It is used for defining filing procedures that can be used when building the filing key for item records, which is stored in Z30-CALL-NO-KEY and Z30-CALL-NO-2-KEY.

The table is used for the Services Shelf Reading Reports (p-item-04, -05 and -06), using the location type.

A single filing procedure is made up of a number of individual procedures, which are explained following. The procedures are the same as for tab_filing, and some may appear to be irrelevant.

**Note** that the order in which the procedures is listed is important. For example, *numbers* (which compresses comma or dot in numbers to change 100,000 and 100.000 to 100000) must come before listing comma or dot in *change-to-blank*.

For list of filing procedures refer to tab_filing. Note, however, the following addition for tab_filing_call_no:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Action</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_call_no_type</td>
<td>adds the value in the call number type field as the first position of Z30-CALL-NO-KEY plus one blank. Because this procedure can cause multiple contiguous blanks, it should be followed by pack_spaces.</td>
<td></td>
</tr>
</tbody>
</table>

Structure of table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>ID - One-digit identifier of the call no type</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>ALPHA</td>
</tr>
<tr>
<td>col. 3</td>
<td>Name of the filing procedure</td>
</tr>
<tr>
<td>col. 4</td>
<td>Parameters for the filing procedure (when relevant)</td>
</tr>
</tbody>
</table>

Example of the table:
Fix routines are "logical names" for defining a group of fix programs. They are invoked by the Fix Document option under the Edit menu in the cataloging client.

Reserved fix routines also define when the programs are run. For example, it is possible to define a group of fix programs to run when the record is loaded to the server. To de-activate a routine either comment out the line, or delete the line.

The following are reserved routines:

- **INS** - all programs linked to the INS group will be automatically activated when a record is sent to the server.
- **INS2** is run automatically whenever a record is updated in the Catalog GUI. The difference between INS and INS2 is that INS2 is executed just before the update in the database, and therefore can make use of the document's system number even if its a new document. However, note that check_doc programs cannot check the outcome of INS2 programs!!!
- **INSFS** - is run automatically whenever a fast cataloging is done from Circulation GUI.
- It is also performed when bibliographic records are created using the Special Request option in the OPAC and when bibliographic records are created in the Course Reading module.
- **MERGE** - all programs linked to MERGE will be activated when the editing option paste document (alt+D) is invoked in the cataloging client.
- **LOCAT** - is run automatically whenever the Locate record function is used in the Catalog GUI.
- **HOL** - is run automatically whenever a HOL record is created in Items or Serials GUI. Note that this instance should be defined in the holdings library (XXX60).
- **P-31** is run automatically whenever authority records are loaded to the authority database by the *Load Authority Records* batch process (p_manage_31).
- **MNG50** is run automatically whenever authority records are loaded using p_manage_50. This includes ADM and HOL records only.
- **ILL-L** is run automatically whenever the Locate function is used in the ILL module (Locate button in the BIB Info tab of the ILL request). Note that this instance should be defined in the ILL library (XXX20).
- **M-36** is run on the records in the input file for the Check Input File Against Database (manage-36) service.
- **BNA** is run automatically when the Load BNA Records (file-98) service is used to load BNA records.

**Program arguments:**
Certain fix_doc programs require additional information such as table names. These additional parameters are defined in column 3. The documentation for each fix_doc program indicates whether it takes parameters, and if so, how they should be formatted. Note that if a fix_doc program does not use parameters, it will ignore the contents of column three.

**Note:** that 58-IN-RECORD-FIX has been implemented for gate definition files. This entry is valid in the conf files in `$alephe_root/gate/*_conf` and can direct a fix when a record is retrieved from a z39 target. The value of Z58-IN-RECORD-FIX should be a fix routine defined in tab_fix of the EXTNN library.

Note that FIX-DOC programs are defined in this table. The table fix_doc.<lng> is used for defining the text that displays in the window when user invokes ‘fix record’ from the edit menu in the cataloging client.

**Structure of the table:**
- col. 1  logical name/routine used for a group of fixing programs
- col. 2  Up to 20 program names may be assigned to each user-defined routine name.
  - For tab04 (conversion from one cataloging format to another), use tab04-nn where nn is the entry defined in tab04.
  - For fix_doc_merge use fix_doc_merge_nn where nn is the merge set in tab_merge_overlay
- col. 3  Program arguments: Certain fix_doc programs require additional information such as table names. These additional parameters are defined in column 3. The documentation for each fix_doc should indicate whether or it takes parameters from this column, and if so, how they should be formatted. Note that if a fix_doc does not use parameters, it will ignore the contents of column three.

**Example of the table:**

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
</tr>
</thead>
</table>
INS fix_doc_tag_008
INS fix_doc_tag_008_open_date
...
INS2 fix_doc_lkr_up
INS2 fix_doc_001
!*
INSFS fix_doc_tag_008
INSFS fix_doc_tag_008_open_date

**tab_fix_local_notes.conf**

**Location of the table:** tab directory of the BIB library
**Purpose of the table:** Order of HOL records to be expanded into the BIB record.

There is an option for expanding local data held in HOL records into the BIB record, using mapping, per logical base. The mapping sets which HOL records are expanded into the BIB record for WEB OPAC display, dependent on the logical base being viewed.

The expand procedure, called expand_doc_bib_local_notes.cbl, expands the local notes from the BIB’s HOL’s into the BIB. Two parameters control the behavior of the program - the search base and its corresponding section in tab_expand_local_notes.conf, under the right search base paragraph. Under this paragraph, we can define the order of the HOL to be expanded according to their owners.

Tab_expand must have
WEB-FULL expand_doc_bib_local_notes

defined in order for this to be activated.

Example of the table:

```
local notes = 590##,690##
owners subfield = 9
owners list = AA, BB, LIN
merge section = 98
mapping section = LCN-2-HOL
```

**tab_fix_z103**

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** Item and HOL filtering

For consortium catalogs or systems with many sublibraries the options for the sorting of item and HOL display has been enhanced by the GUI-TREE routine in this table. It enables a sorted display in the navigation tree.

**Routine name:**

- GUI-FULL
• GUI-TREE
• HOL-LIST
• WEB-FULL
• WEB-SHORT
• WEB-Z103
• WEB-SET

Fix z103 Procedures:
The following is a partial list of available programs.
• fix_z103_filter_base
• fix_z103_filter_suppress
• fix_z103_sort_852_b
• fix_z103_sort_852_b_item_attr
• fix_z103_sort_base
• fix_z103_sort_doc_no
• fix_z103_sort_existing_key
• fix_z103_sort_lkr_doc_no

Structure of the table:
Col. 1 Routine name
Col. 2 Program name
Col. 3 Program arguments

**tab_hld_stmt**

Location of the table: tab directory of the ADM library

Purpose of the table: Holdings summary statement building

When expand_doc_hld_stmt is invoked, the system builds a holdings summary statement in the HOL record, using the Z30 item records that are linked to the HOL record.
The item records that are taken into consideration are:
1) items with enumeration field equal or greater than the HOL S63/4/5 field in the HOL record (or all items, if there is no S63/4/5)
2) items with processing status NP (not published) or NA (not arrived) are not included in the summary statement, but they are used for generating a break indication.

Items which do not fall into the above two categories can also be excluded from the summary holdings statement by mapping them to be considered as if they have processing status NP or NA. This mapping can be done for items that have a particular sublibrary and/or collection, and/or item status, and/or item process status, and/or Break indicator.

Structure of the table:
Col. 1 Sublibrary code
Col. 2 Collection code
Col. 3 Item status code
Col. 4  Item process status code  
Col. 5  Break indicator value  
Col. 6  Map to process status NA or NP  

Example of the table:  

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td></td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>NA</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_hol_item_create**

**Location of the table:** tab/import directory of the **BIB** library

**Purpose of the table:** Defines where in the bibliographic record item holdings data elements are to be found for the creation of items and holdings

This table is an example of a table that can be used by the p-manage-50 batch process for the creation of items and/or holdings records. The table defines where in the bibliographic record item and holdings data elements are to be found.

The user can customize more than one table of this type for the convenience of conversions and customers who need to use the p-manage-50 batch job for more than one source data.

The table defines the fields and sub-fields of the bibliographic record that are mapped into the given Z30 (item) fields and as fields in the HOL record. Fields for HOL are identified by HOL- as a prefix to the field in the first column. The name of the table can then be used as a parameter to the p-manage-50 batch job.

Note that when overwriting an existing item with a new one, only the fields defined in this table are overwritten; other fields remain untouched.

The list of valid fields follows. Note that the convention being followed is to use the Z30 field name, except that if a field is valid in both the holdings and item record, the ‘Z30-’ prefix is omitted.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>HOL Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z30-BARCODE</td>
<td>none</td>
</tr>
<tr>
<td>SUB-LIBRARY</td>
<td>852$b</td>
</tr>
<tr>
<td>Z30-MATERIAL</td>
<td>none</td>
</tr>
<tr>
<td>Z30-ITEM-STATUS</td>
<td>none</td>
</tr>
<tr>
<td>Z30-CATALOGER</td>
<td>none</td>
</tr>
<tr>
<td>Z30-ALPHA</td>
<td>none</td>
</tr>
<tr>
<td>COLLECTION</td>
<td>852$c</td>
</tr>
</tbody>
</table>
Field Name | HOL Equivalent
---|---
CALL-NO-TYPE | 852 first indicator
CALL-NO | 852$hijklm
Z30-CALL-NO-2-TYPE | none
Z30-CALL-NO-2 | none
Z30-DESCRIPTION | none
Z30-NOTE-OPAC | none
Z30-NOTE-CIRCULATION | none
Z30-NOTE-INTERNAL | none
Z30-INVENTORY-NUMBER | none
Z30-PAGES | none
Z30-ITEM-STATISTIC | none
COPY | 852$t
Z30-TEMP-LOCATION | none

Note that the Z30-CATALOGER is automatically generated by the batch job.

Following are the fields specific to holdings records:

<table>
<thead>
<tr>
<th>Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>866-IND (866 indicators)</td>
</tr>
<tr>
<td>866-ALPHA</td>
</tr>
<tr>
<td>866-TEXT</td>
</tr>
<tr>
<td>867-IND (867 indicators)</td>
</tr>
<tr>
<td>867-ALPHA</td>
</tr>
<tr>
<td>867-TEXT</td>
</tr>
<tr>
<td>868-IND (868 indicators)</td>
</tr>
<tr>
<td>868-ALPHA</td>
</tr>
<tr>
<td>868-TEXT</td>
</tr>
</tbody>
</table>

Currently the program does not send textual holdings to different holdings records attached to the same bibliographic record. If the values are present they are inserted into all the holdings records created.

Note that it is possible to define fields to add to the holdings records by adding the "HOL-" prefix to the field codes. For example:

<table>
<thead>
<tr>
<th>HOL-OWN</th>
<th>Y a OWN a</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOL-852</td>
<td>Y a 950 l</td>
</tr>
<tr>
<td>HOL-866</td>
<td>Y 040</td>
</tr>
</tbody>
</table>

In the above example, the service:
- adds the OWN field to the holdings record with subfield $a based on the OWN field, subfield a, of the bibliographic record
- adds subfield $a to the 852 field of the holdings record based on subfield $l of the 950 field of the bibliographic record
- maps the 040 field of the bibliographic record to the 866 field of the holdings record.
If the field specified in column 1 already exists, the data extracted from the bibliographic record is added to the existing field.

In addition, the following options have been added to the table:

- CREATE-ITEM-1ST (for first indicator)
- CREATE-ITEM-2ND (for second indicator)
- CREATE-ITEM-SUB (for subfields)

These options are used to define conditions where only the holdings record is created without an item. For example, if the table contains the following line, then the item is only created if the first indicator of the main field is 0:

```
CREATE-ITEM-1ST           0
```

If the table contains the following line, then the item is only created if the main field contains subfield $p$:

```
CREATE-ITEM-SUB           p
```

Note that it is possible to define that, for example, the call number field should be created from different fields from the bibliographic record.

Following are sample lines for the creation of the call number from various fields:

```
CALL-NO                   k          Y k
CALL-NO                   h          Y h          099   a          090   a
```

In the above example,
- subfield $k$ of the item's Z30-CALL-NO field and the holdings 952 field should be created from subfield $k$ of the main field.
- Subfield $h$ of the item's Z30-CALL-NO field and the holdings 952 field should be created either from $h$ of the main field (if present), or from subfield $a$ of the 099 field (if it does not exists in the main field), or from subfield $a$ of the 090 field (if it does not exists in the 099 field).

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>HOL/Z30 Field name</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Sub-field in main tag</td>
</tr>
<tr>
<td>col. 3</td>
<td>Retain sub-field labels</td>
</tr>
<tr>
<td></td>
<td>together with the field</td>
</tr>
<tr>
<td></td>
<td>content. For example: $h954.32</td>
</tr>
<tr>
<td>col. 4</td>
<td>Destination sub-fields</td>
</tr>
<tr>
<td></td>
<td>(for sub-field labels</td>
</tr>
<tr>
<td></td>
<td>retained)</td>
</tr>
<tr>
<td>col. 5</td>
<td>First tag</td>
</tr>
<tr>
<td>col. 6</td>
<td>First sub-fields</td>
</tr>
<tr>
<td>col. 7</td>
<td>Second and third tags</td>
</tr>
<tr>
<td></td>
<td>and sub-fields</td>
</tr>
<tr>
<td>col. 11</td>
<td>Normalization code</td>
</tr>
</tbody>
</table>
Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>...</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!-!!!!!!!!!!!!!!-!!!!!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-
-!!!!!!!!!!-!!!!!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!

SUB-LIBRARY s N b LOC## b
COLLECTION c N c LOC## c
Z30-CATALOGER r N
Z30-BARCODE b N
Z30-MATERIAL m N
Z30-ITEM-STATUS i N
CALL-NO Y hi 090## ab 050## ab

**tab_hol_item_map**

Location of the table: tab/import directory of the BIB library

Purpose of the table: Mapping bibliographic values into the relevant holdings and/or items values

This table is an example of a table that can be used by the p-manage-50 batch process for the mapping of the supplied bibliographic values into the relevant holdings and/or items values.

The list of valid fields is as follows. Note that the convention being followed is to use the Z30 field name, except that if a field is valid in both the holdings and item record, the ‘Z30-’ prefix is omitted.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>HOL Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z30-BARCODE</td>
<td>none</td>
</tr>
<tr>
<td>SUB-LIBRARY</td>
<td>852$s</td>
</tr>
<tr>
<td>Z30-MATERIAL</td>
<td>none</td>
</tr>
<tr>
<td>Z30-ITEM-STATUS</td>
<td>none</td>
</tr>
<tr>
<td>Z30-CATALOGER</td>
<td>none</td>
</tr>
<tr>
<td>Z30-ALPHA</td>
<td>none</td>
</tr>
<tr>
<td>COLLECTION</td>
<td>852$c</td>
</tr>
<tr>
<td>CALL-NO-TYPE</td>
<td>852 first indicator</td>
</tr>
<tr>
<td>CALL-NO</td>
<td>852$hijklm</td>
</tr>
<tr>
<td>Z30-CALL-NO-2-TYPE</td>
<td>none</td>
</tr>
<tr>
<td>Z30-CALL-NO-2</td>
<td>none</td>
</tr>
<tr>
<td>Z30-DESCRIPTION</td>
<td>none</td>
</tr>
<tr>
<td>Z30-NOTE-OPAC</td>
<td>none</td>
</tr>
<tr>
<td>Z30-NOTE-CIRCULATION</td>
<td>none</td>
</tr>
<tr>
<td>Z30-NOTE-INTERNAL</td>
<td>none</td>
</tr>
<tr>
<td>Z30-INVENTORY-NUMBER</td>
<td>none</td>
</tr>
<tr>
<td>Z30-PAGES</td>
<td>none</td>
</tr>
<tr>
<td>Z30-ITEM-STATISTIC</td>
<td>none</td>
</tr>
</tbody>
</table>
Following are the fields specific to holdings records:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>HOL Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z30-ITEM-PROCESS-STATUS</td>
<td>none</td>
</tr>
<tr>
<td>COPY</td>
<td>852$t</td>
</tr>
<tr>
<td>Z30-TEMP-LOCATION</td>
<td>none</td>
</tr>
</tbody>
</table>

Structure of the table:
- col. 1 HOL/Z30 Field name
- col. 2 Case-sensitive matching
- col. 3 Bibliographic value
- col. 4 HOL/Z30 value

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUB-LIBRARY</td>
<td>Y SBL1</td>
<td></td>
<td>UEDUC</td>
<td></td>
</tr>
<tr>
<td>SUB-LIBRARY</td>
<td>Y SBL2</td>
<td></td>
<td>ULINC</td>
<td></td>
</tr>
<tr>
<td>SUB-LIBRARY</td>
<td>Y MAIN</td>
<td></td>
<td>UEDUC</td>
<td></td>
</tr>
<tr>
<td>SUB-LIBRARY</td>
<td>Y main</td>
<td></td>
<td>MAIN</td>
<td></td>
</tr>
<tr>
<td>CALL-NO-TYPE</td>
<td>N 0</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CALL-NO-TYPE</td>
<td>N #</td>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

**tab_hol_mapping**

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** Definition of mapping values when creating records with p-manage-500

This table is an example of a table that can be used by the p-manage-500 batch process for the creation of items and/or holdings records. The table defines where in the bibliographic record item and holdings data elements are to be found.
Structure of the table:

Col. 1 Holding Code. Value in 049 (OCLC code). Note that this value acts as a filter. If an 049 value is not found in this table, no HOL or Items will be created for that bib. The value can be wildcarded. However, you cannot setup defaults at the top of the table and specifics at the bottom.

Col. 2 HOL/Z30 Field name. Z30 always refers to location data and HOL refers to tags in the HOL record.

Col. 3 source tag/arguments. The arguments can vary. When a value is enclosed in quotation marks, the value is taken as is and mapped to the locations. When mapping 852 $$h and $$I the first set of values is subfields in bib record and the second set is subfield in the hol record.

Col. 4 Second Source

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>AAAA</td>
<td>Z30-SUB-LIBRARY</td>
<td>XLINC</td>
<td></td>
</tr>
<tr>
<td>AAAA</td>
<td>Z30-COLLECTION</td>
<td>GEN</td>
<td></td>
</tr>
<tr>
<td>AAAA</td>
<td>Z30-CALL-NO-TYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>HOL-852</td>
<td>050$$,ab,hi</td>
<td>0900,ab,hi</td>
<td></td>
</tr>
<tr>
<td>HOL-852</td>
<td>&quot;$$aExL&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOL-852</td>
<td>053,a,t</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_wait_request**

Location of the table: tab directory of the ADM library.

Purpose of the table: checks for hold requests.

This table is used to define which checks should be performed by the system when a hold request is placed. The checks can influence:

- display (or not) of the Request link on the list of items
- display of information regarding the group of items and the patron's place in the request queue
- display of meaningful messages concerning the reason why a request is refused.

There are **FOUR** types of checks:

- **PRE** is used when the list of items is displayed in the Web OPAC
- **INFO** is used when displaying the hold request fill-in form
- **POST** is used when the operator sends the request to the server
- **CIRC** is used in the Circulation client
If a check that requires patron information is used as a **PRE** check, then the patron must first be identified, and you must set check\_hold\_request\_l as a **PRE** check as well.

The following checks can be performed:

<table>
<thead>
<tr>
<th>Check</th>
<th>Action</th>
<th>Heading</th>
<th>Suggested use</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>if tab15.&lt;lng&gt; col. 8 =N, item cannot be requested</td>
<td>1100</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>b</td>
<td>if sub-library and item status are not listed in tab37, item cannot be requested.</td>
<td>1105</td>
<td>PRE, POST, CIRC</td>
</tr>
<tr>
<td>c</td>
<td>if tab15.&lt;lng&gt; col. 12=Y, the item can have only one request</td>
<td>1120</td>
<td>PRE, POST, CIRC</td>
</tr>
<tr>
<td>d</td>
<td>z305 hold-permission</td>
<td>1110</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>e</td>
<td>if combination of sub-library, item status and patron status are not listed in tab37, item cannot be requested</td>
<td>1115</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>f</td>
<td>z305 multi-hold allowed</td>
<td>1125</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>g</td>
<td>z305 hold-on-shelf (item availability)</td>
<td>1130</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>h</td>
<td>tab16 col. 13, maximum number of hold requests</td>
<td>1135</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>I</td>
<td>if the item is on loan, displays due date and information on whether lost or recalled</td>
<td>6000-6002</td>
<td>INFO</td>
</tr>
<tr>
<td>j</td>
<td>if item has requests, displays on hold, patron's place in queue</td>
<td>6005,6006</td>
<td>INFO</td>
</tr>
<tr>
<td>k</td>
<td>tab15.&lt;lng&gt; col. 8 value (Y,N,C)</td>
<td>6010-6012</td>
<td>INFO</td>
</tr>
<tr>
<td>l</td>
<td>user-id has not been entered</td>
<td>1140</td>
<td>PRE</td>
</tr>
<tr>
<td>m</td>
<td>patron has item on loan and is attempting to request a like copy</td>
<td>1145</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>n</td>
<td>patron tries to reserve an item that was declared as 'Lost' or 'Claimed Returned'</td>
<td>1095</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>o</td>
<td>checks if there is a &quot;like&quot; copy available (same year, volume, part, issue, processing status, AND depending on tab100 setup</td>
<td>1150</td>
<td>INFO</td>
</tr>
<tr>
<td>p</td>
<td>checks if all &quot;like&quot; copies are on loan (not available) and informs the patron that the request cannot be placed, that he must apply to the circ desk. This check should be used by a library that wants library staff to decide which copy should be requested (because recall is made for the copy requested, and not all copies)</td>
<td>1155</td>
<td>POST, CIRC</td>
</tr>
<tr>
<td>q</td>
<td>tab16 col. 13 check for maximum number of hold requests for the sub-</td>
<td>1160</td>
<td>POST, CIRC</td>
</tr>
</tbody>
</table>
library ("99" line)

r patron expiry date has been reached 1165 POST, CIRC
s patron has the exact same item on loan 1170 POST, CIRC
t tab16 col. 13 check for maximum number of hold requests for the ADM library (ADM line(s) in tab16) 1175 POST, CIRC
u z305 reading room permission 1176 POST, INFO, CIRC
v if the item belongs to a remote storage and it is available, the expected delivery time is calculated (using tab17+tab24). The message displays in INFO in place of the "Your place in queue" message (j) 6003 INFO

Structure of the table:

- col. 1 defines when check is performed:
  - PRE, POST, CIRC, INFO
- col. 2 defines check procedure to perform – as listed in the table above.
  - The check code should be entered as per the following format check_hold_request_a/b/c..

NOTE: If a check that requires patron information is used as a PRE check, then the patron must first be identified, and check_hold_request_l must be set as a PRE check as well. In this scenario, **if the patron is not signed-in no request links will display.**

Note that some of the checks relate to the item, some to the patron and some to a combination of item-patron. Some checks are relevant only to PRE, POST or INFO.

If a check is listed for PRE, it need not be listed for INFO or POST, since the INFO/POST scenario will not happen.

When setting which checks to use, take into account the system resources required for checks that are made when the items list displays (PRE checks). If the checks require comparing item records and existing requests, and there are many item records, this can create an undesirable load on the server.

Example of the table:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>1</td>
</tr>
<tr>
<td>!</td>
<td>2</td>
</tr>
<tr>
<td>!</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
</tr>
<tr>
<td>PRE</td>
<td>check_hold_request_a</td>
</tr>
<tr>
<td>PRE</td>
<td>check_hold_request_b</td>
</tr>
<tr>
<td>PRE</td>
<td>check_hold_request_c</td>
</tr>
<tr>
<td>PRE</td>
<td>check_hold_request_l</td>
</tr>
<tr>
<td>PRE</td>
<td>check_hold_request_n</td>
</tr>
</tbody>
</table>
| !* | *
| POST | check_hold_request_d |
| POST | check_hold_request_e |
| POST | check_hold_request_f |
**tab_hold_request_form**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Defining hold request forms for use in the OPAC.

This table sets the option to configure the system to use different hold request forms when the patron places a request through the Web OPAC. When a request is placed, the system will look for a matching line in the table. The correct line will be selected according to one of the available parameters in this table:

- the item's sublibrary, -
- item status,
- item process status,
- patron status and
- item's availability that is determined according to the -AVAILABILITY-ROUTINE flag in tab100.

If no match is found, or the table does not exist, the system will display the regular hold request forms. The same happens if the table does not exist.

Note that specific lines should come before the general ones. For example:

```
    ###### 32 DP ## # depo32
    ###### ## DP ## # depo
```

The forms that can be changed are:

- item-hold-request-body
- item-hold-request-head
- item-hold-request-tail

All three forms need to be set with the defined suffix for the system to use them if the request placed match the settings in the table.

**Structure of the table:**

| Col. 1 | SubLibrary. Use ## to denote any (all) sub-library |
| Col. 2 | Item status. Use ## to denote any (all) item status |
| Col. 3 | Item Process status |
| Col. 4 | Patron status. Use ## to denote any (all) patron status |
| Col. 5 | Item Availability Status (Using TAB100-AVAILABILITY-ROUTINE) |
| | • Y = Available item |
| | • N = Non available item |
| Col. 6 | WWW Form suffix |
Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

```
#### 70 DP ## # depo
#### 71 DP ## # depo
```

**tab_ill_charge**

Location of the table: tab directory of the ADM library

Purpose of the table: Parameters for charging patrons for ILL requests

This table defines the parameters for charging patrons for ILL requests. This table will only be activated if tab18 has transaction line 0015 – ILL material arrival set to Y in col. 4, and with a zero amount (0) in col. 5.

- Col. 1. Patron status. Use # for wildcard
- Col. 2 Supplier status (type). Use # for wildcard. Free text as defined by the library. In the example below,
  - DO has been used to denote ‘domestic’ suppliers, and
  - FO to denote ‘foreign’ suppliers.
- Col. 3 Page size. Use # for wildcard
- Col. 4 Media type (Loan or Photocopy)
- Col. 5 Fixed charge1. Last two positions are after the decimal point – e.g. 0100=1.00
- Col. 6 Fixed charge2
- Col. 7 Fixed charge3
- Col. 8 Start counting after page X
- Col. 9 Fee per page. Last two positions are after the decimal point – e.g. 0100=1.00

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
</tbody>
</table>

```
## DO ## L 0000002500 0000000000 0000000000 000 000
## FO ## L 0000005000 0000000000 0000000000 000 000
## DO ## C 0000002500 0000000000 0000000000 005 0000000200
## FO ## C 0000005000 0000000000 0000000000 005 0000000250
```

**tab_ill_in_index**

Location of the table: tab directory of the ADM library

Purpose of the table: Display of bibliographic fields in ILL Incoming requests index

This table defines the bibliographic fields for the ILL Incoming Requests Index.
The name of the index must also be added to list of field values
pc_tab_exp_field.<lng> - ILL_INDEX_TYPE.

In order to create the index retrospectively run 'Rebuild ILL and Acquisitions Order Index' (p-acq-04) from the ILL batch service.

Note: Make sure that the Index Type (Col. 1 of table) is unique and is not used in either of the following 2 ADM library tables:

- tab_ill_index - Display of bibliographic fields in ILL Outgoing requests' index
- tab_acq_index - Display of bibliographic fields in ACQ Order Index

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>col. 2</th>
<th>col. 3</th>
<th>col. 4</th>
<th>col. 5</th>
<th>col. 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index name (up to 5 characters)</td>
<td>bibliographic field code+sub-field code</td>
<td>1st alternative</td>
<td>2nd alternative</td>
<td>3rd alternative</td>
<td>4th alternative</td>
</tr>
</tbody>
</table>

Example of the table:

```
! 1 2 3 4 5 6
!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!
TITIN 245##a
AUTIN 100## 110## 111## 130##
```

**tab_ill_index**

Location of the table: tab directory of the ADM library

Purpose of the table: Display of bibliographic fields in ILL Outgoing requests index

This table defines the bibliographic fields for the ILL Outgoing Requests Index.

The name of the index must also be added to list of field values
pc_tab_exp_field.<lng> - ILL_INDEX_TYPE.

In order to create the index retrospectively run 'Rebuild ILL and Acquisitions Order Index' (p-acq-04) from the ILL batch service.

Note: Make sure that the Index Type (Col. 1 of table) is unique and is not used in either of the following 2 ADM library tables:
• tab_ill_in_index - Display of bibliographic fields in ILL Incoming requests' index
• tab_acq_index - Display of bibliographic fields in ACQ Order Index

Structure of the table:

col. 1  Index name (up to 5 characters)
col. 2  bibliographic field code+sub-field code
col. 3  1st alternative
col. 4  2nd alternative
col. 5  3rd alternative
col. 6  4th alternative

Example of the table:

```
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!!!!-</td>
<td>!!!-</td>
<td>!!!-</td>
<td>!!!-</td>
</tr>
</tbody>
</table>
TITL| 245##a|
ARTIC| 740##|
AUTOR| 100## 110## 111## 130##|
ISBNO| 020##|
ISSNO| 022##|
```

**tab_ill_unit.<lng>**

Location of the table: alephe/tab directory

**Purpose of the table: Definition of ILL centers for the External ILL module**

This table defines the ILL Centers that are listed when a new ILL request is created via an ILL Center. The table lists the ILL Centers and their server address.

- Col. 1  Code of the displayed unit
- Col. 2  Name of the displayed unit
- Col. 3  Y/N
  - Y=Unit displayed,
  - N=Unit not displayed in dropdown list. When "N" is used a direct request is still possible. For example, for control using Z303-ILL-LIBRARY, set the value to "N" and the ILL UNIT will automatically be transferred from Z303
- Col. 4  ILL Center's server address

Example of the table:
tab_io_remote

Location of the table: alephe/tab directory

Purpose of the table: Remote library specifications

This table is used when a particular library (database) is in a remote server. For example, if the authorities library is on another server, there is a need to define to which server the authority library belongs.

Structure of the table:
Col. 1 Library code
col. 2 Hostname
col. 3 ALEPH version

Example:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!</td>
<td>--!</td>
<td>!!!!!!!!!!!!!!!!!!!!!!--!</td>
</tr>
<tr>
<td></td>
<td>U60AR</td>
<td>U60AR ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
<tr>
<td></td>
<td>U60ED</td>
<td>U60ED ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
<tr>
<td></td>
<td>U60EL</td>
<td>U60EL ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
<tr>
<td></td>
<td>U60HS</td>
<td>U60HS ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
<tr>
<td></td>
<td>U60LN</td>
<td>U60LN ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
<tr>
<td></td>
<td>U60LW</td>
<td>U60LW ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
<tr>
<td></td>
<td>S60GD</td>
<td>S60GD ILL UNIT</td>
<td>Y 10.1.235.47:12001</td>
</tr>
</tbody>
</table>

tab_item_history.<lng>

Location of the table: tab directory of the ADM library

Purpose of the table: Item History trigger and description

This table defines in which cases a field change or an action will cause an item history record to be written. This information is accessible via the “history” button in the Item module.

In addition, the table is also used for setting the description of the action. This description is used for display in the item history window.

The actions which can trigger a history record are changes in:

ITEM-LOCATION   L Call No.
ITEM-CALL-NO-2  L Call No.2
ITEM-COLLECTION L Collection
ITEM-SUB-LIBRARY L Sublibrary
ITEM-PROCESS-STATUS L Process Status
ITEM-BARCODE L Barcode
BIND L Deleted (Bound)
DELETE L Deleted
ITEM-STATUS L Status
ALL-FIELDS L All fields

Use "ALL-FIELDS" in order to register a change in any other fields. If using ALL-FIELDS, list it last. If it is listed before specific changes, registration of the specific changes is ignored.

Structure of the table:
Col. 1 Field changed or action that causes history record to be written
Col. 2 ALPHA
Col. 3 Description of change made to the item (15 characters only)

Example of the table:
<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM-SUB-LIBRARY</td>
<td>L Sublibrary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM-BARCODE</td>
<td>L Barcode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM-PROCESS-STATUS</td>
<td>L Process Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM-LOCATION</td>
<td>L Call Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL-FIELDS</td>
<td>L All</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_item_list_order**

Location of the table: tab directory of the BIB library

Purpose of the table: Display order of the items

This table determines the display order of the items in the Web OPAC and GUI SEARCH module according to the sublibraries for each base. It is possible to define a different display order for each base. If you create a table for each base, the table name should include the base name as an extension of the table name (e.g. tab_item_list_order_serials for the logical base "serials"). In the event that there is no table for a base, the program uses the "tab_item_list_order" table.

NOTE: The item order from this table will be consulted only when "02" is defined in tab_sub_lib_sort column 2.

Structure of the table:
col. 1 sub-library

Example of the table:
**tab_itm_mapping**

Location of the table: tab directory of the BIB library

Purpose of the table: Definition of mapping values when creating records with p-manage-500

Structure of the table:
- Col. 1 Z30 Field name
- Col. 2 Alpha
- Col. 3 Source value
- Col. 4 Target value

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!!!!-!!!!!!!</td>
<td>UHLTH</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ULAW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ULINC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UMUSI</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USCI</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UARCV</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!!!!!-!!!!!!!!</td>
<td>Z30-MATERIAL</td>
<td>BOOK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-ITEM-STATUS</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-CATALOGER</td>
<td>RLIN-LOAD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-ALPHA</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-DESCRIPTION</td>
<td>Z30-DESCRIPTION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-NOTE-OPAC</td>
<td>Z30-NOTE-OPAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-NOTE-CIRCULATION</td>
<td>Z30-NOTE-CIRCULATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-NOTE-INTERNAL</td>
<td>Z30-NOTE-INTERNAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-PAGES</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z30-ITEM-STATISTIC</td>
<td>ITEM-STATISTIC</td>
<td></td>
</tr>
</tbody>
</table>

**tab_label.<lng>**

Location of the table: tab directory of the ADM library

Purpose of the table: label prefixes for spine labels

This table defines the prefixes that will print on the spine label. The number of lines that can be defined in this table is 2000
Structure of the table:

Col. 1  Sub-library code
Col. 2  Collection code
Col. 3  Prefix (or suffix) delimited by ';' (semi-colon) which denotes line feed

For information on how suffixes are handled refer to the document, “How To Set Up Label Printing.”

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!!!!!!!</td>
<td>!!!!!!!!</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!&gt;</td>
</tr>
<tr>
<td>UGDOC</td>
<td>GEN</td>
<td>UGDOC; General</td>
<td></td>
</tr>
<tr>
<td>UHLTH</td>
<td>REF</td>
<td>UHLTH; Reference</td>
<td></td>
</tr>
<tr>
<td>UMUSI</td>
<td></td>
<td>UMUSI;</td>
<td></td>
</tr>
<tr>
<td>UEDUC</td>
<td>GEN</td>
<td>UEDUC; General</td>
<td></td>
</tr>
</tbody>
</table>

**tab_label_parse**

Location of the table: tab directory of the ADM library

Purpose of the table: Parsing call numbers for label printing

This table specifies the routines used to parse (split) the call number and item description when generating item labels.

**Column 1** specifies the field to be parsed.
- C = call number
- D = item description

Columns 2, 3, and 4 are used only if column 1 is set to C.

**Column 2** specifies the call number type to match.

**Columns 3 and 4** specify an optional sub-field and contents to filter the call number for.

For example:
if ‘2’ and '[mesh]' are listed in these columns, the parsing routine in column 6 will be used only if the call number type matches and the call number contains a sub-field $$2 [mesh]$.  
Note that the sub-field contents must be entered in lower case.

**Column 5** is used only if column 1 is set to D. It specifies the item material type to filter for. If left blank, acts as a default entry for parsing item descriptions.

**Column 6** specifies the routine to use to parse the call number or item description.

**Programs for parsing call numbers (type C):**
- parse_call_no_default
- parse_call_no_lc_1
- parse_call_no_lc_2
• parse_call_no_lc_3
• parse_call_no_dw_1

The first routine is general, and breaks on a space or subfield i.
The other routines include breaking on space or subfield i, together will additional features.
The three ‘lc’ routines are intended to be used with LC (Library of Congress) and NLM (National Library of Medicine) call numbers.

Examples for the routines are given, using
$$hKJV444.21804 A7$L63 1805$$ and
$$hG635.H4$A3 1989^a$$

parse_call_no_default:
splits the call number at spaces and subfield i.
If the character ^ is present in the call number, converts it to a space but does not break at that point.

parse_call_no_lc_1:
In addition to parse_call_no_default, breaks between the letter and number components of the classification part of the call number (even if there is no space) and breaks on a decimal preceding a letter (even if it is not preceded by a space).
  e.g.
  KJV     G
  444.21804   635
  A7     .H4
  L63   A3
  1805  1989 a

parse_call_no_lc_2:
In addition to parse_call_no_lc_1, when a line would end up longer than eight characters, it will be split on a ".".
  e.g.
  KJV     G
  444     635
  .21804  .H4
  A7     A3
  L63   1989 a

parse_call_no_lc_3:
The same as parse_call_no_lc_2, except that no space is added to force a line break after the class code.
  e.g.
  KJV444   G635
  .21804  .H4
  A7     A3
  L63  1989 a
  1805

parse_call_no_dw_1:
This is a site-specific parsing routine.
Break at the end of the first string of numerals terminated with a space or a full stop. The full stop remains on the first line.

Then the following rules apply:

- **Non periodicals:** Start a new line after every three numeric characters or after every space (whichever occurs first).
- **Periodicals:** All call numbers that begin with P (other than those beginning with P/) and end with either a suffix consisting of a five-digit string or a five-digit string followed by a decimal point and a two-digit string. This suffix breaks at the decimal point, or after the five-digit string. All characters/digits preceding the suffix for these "P" call numbers are handled as non-periodicals.

Note: if no matching program is specified, or if the program does not exist, `parse_call_no_default` will be used.

**Programs for parsing item descriptions (type D):**

- **`parse_desc_default`:**
  splits the item description at spaces.

- **`parse_desc_niso_8`:**
  parse item descriptions formatted according to ANSI/NISO Z39.71, ("Holdings Statements for Bibliographic Records"). It also attempts to limit each line of the item description to 8 characters.
  This routine uses the following algorithm:
  - remove any trailing information in parentheses, square brackets, or angle brackets
  - break at "+" or "&", which indicate bibliographic units, retaining the "+" or "&"
  - within a bibliographic unit, break on ".:" but do not retain the colon in the label
  - within a level, break on "," and ";;", retaining the punctuation
  - if a segment produced by (d) is longer than 8 characters, break up to two times on the ", -" and "/" characters, retaining them on the label
  - if more than ten lines of item description would be produced based on (b)-(e), append the rest of the item description to the tenth line, even if it ends up longer than 8 characters.

Example: the item description

```
```

would be parsed into

```
v.120
no.1
pt.A-B,
D
+ Index
+ Supplement
```

**PARSE_DESC_SEMICOLON**
if a semicolon is present in the description, split on semicolons. Otherwise, split it on
commas.

**parse_desc_semi_no_parens:**
same as parse_desc_semicolon except that except that only the part of the item
description up to the first (, [, or < is used.

Note: if no matching program is specified, or if the program does not exist,
parse_call_no_default will be used.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Item field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Call number type to match</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Filter sub-field</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Filter sub-field contents</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Item material type filter</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Parsing program</td>
</tr>
</tbody>
</table>

Example of the table:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>parse_call_no_lc_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>8</td>
<td>parse_call_no_default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>parse_call_no_dw_1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_late_return**

Location of the table: tab directory of the ADM library

Purpose of the table: Time and rate for fine method "F"

This table defines late return periods and amounts for late return fines, when the fine
method F is set in col. 14 in tab16. Using method "F" a set rate is charged for a time
period.

For example:
A fine is set for 5 Euro for the first 10 days and 50 Euro for the next 5 days. If an item
is returned any time within the first 10 days, the fine charged will be 5 Euro. If the
item is returned any time from day 11 through 15, the fine charged will be 50 Euro.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Sublibrary;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Item status. Use ## to denote any (all) item status.</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Patron status. Use ## to denote any (all) patron status.</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Number of days since due date</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Sum: left-aligned</td>
</tr>
</tbody>
</table>
Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!!!</td>
<td>!</td>
<td>-</td>
<td>-</td>
<td>!</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>!</td>
<td>!</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>!</td>
<td>-</td>
<td>!</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

|   |   |   |   |   |   |
|---|---|---|---|---|
|   |   |   | 0000 | 1.00 |
|   |   |   | 0011 | 2.50 |
|   |   |   | 0021 | 5.00 |
|   |   |   | 0031 | 10.00 |

**tab_library_group**

**Location of the table:** alephe/tab directory

**Purpose of the table:** Grouping of libraries to share the same batch processes.

The table defined a group of libraries to share the same lib_batch and UE processes.

**Structure of the table:**
- Col. 1: Hosting library;
- Col. 2-9: Member libraries

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!!!</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>USM01</td>
<td>USM10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM11</td>
<td>USM12</td>
<td>USM14</td>
<td>USM15</td>
<td>USM19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM20</td>
<td>USM21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM50</td>
<td>USM51</td>
<td>USM52</td>
<td>USM53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_locate**

**Location of the table:** tab directory of the library

**Purpose of the table:** Procedures for locating a record in other libraries through Web OPAC, Cataloging and ILL.

The *locate* function enables the user to search for similar records in other databases.

The locate query is built from the source document, using the target database lines in this table. The query includes all lines for which there is data in the source document, with a Boolean "and" between them.

Multiple lines can be set up for one library, in which case ALL lines will be taken with an AND condition between them. The tab_locate table should include both the source and the target library.
**Extract Functions:**

- **locate_str_0** - Takes sub-field content as is.
- **locate_str_1** - Runs "build_filing_key" on sub-field and takes 2 longest words. A word must be at least 2 characters in order to be considered to be a "word". If the sub-field has only one word, the one word will be taken.
- **locate_str_2** - Takes year from 008## Position 8 Length 4
- **locate_str_3** - Works similarly to locate_str_1, but takes as many longest words as specified in Col. 6 (Parameters).
- **locate_str_no_locate** - Does not perform locate (implemented for ILL for "last resort" supplier)
- **locate_str_sys_no** - Uses a doc number in a specified field to perform an exact match.

**Note:** locate_str_0, locate_str_1 use the standard word breaking section to normalize search term. The default is "97", but the parameter column can be used to override this.

**Structure of the table:**

<table>
<thead>
<tr>
<th>col. 1</th>
<th>col. 2</th>
<th>col. 3</th>
<th>col. 4</th>
<th>col. 5</th>
<th>col. 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>base code of the base where you wish to locate records</td>
<td>tag</td>
<td>Sub-field used as a &quot;locate&quot; parameter. Syntax:</td>
<td>Find command (WRD code) that is used in the target base for searching similar records</td>
<td>extract function that defines in which way are the contents of the field going to be treated</td>
<td>Parameters to be sent to Extract function (Col. 5) Current programs disregard the parameters.</td>
</tr>
</tbody>
</table>

**Example of table:**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!USM01</td>
<td>245## b</td>
<td>wti=</td>
<td>locate_str_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!USM01</td>
<td>100## a</td>
<td>wau=</td>
<td>locate_str_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!USM01</td>
<td>008##</td>
<td>wyr=</td>
<td>locate_str_2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM10</td>
<td>650## a</td>
<td>wsu=</td>
<td>locate_str_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM11</td>
<td>650## a</td>
<td>wsu=</td>
<td>locate_str_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM12</td>
<td>650## a</td>
<td>wsu=</td>
<td>locate_str_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!UNI01</td>
<td>100## a</td>
<td>wau=</td>
<td>locate_str_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNI01</td>
<td>245## a</td>
<td>wti=</td>
<td>locate_str_1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
tab_location_name.<lng>

Location of the table: alephe/tab directory

Purpose of the table: Location names for Z39.50 server

The tab_location_name table defines name and codes for the ALEPH global library (e.g. USM50).

This definition is required for the Z3950 Holdings Schema, which includes location information. Location information can include institution name, institution code and isilcode (ISL code). The location can be made up of Institution and Sub-Institution Name and codes, which parallels the global library (e.g. USM50) and sublibraries in ALEPH.

The sublibrary codes and names are defined in tab_sub_library.<lng>. This table is used to define the global library codes and names.

Structure of the table:

Col. 1 Code of the physical library, e.g. USM50
Col. 2 Standard location code. This is the same as 852 $$a location, which identifies the institution or person holding the item or from which access is given. Standard codes are entered in this column, such as codes in the MARC Code List for Organizations or codes in Symbols and Interlibrary Loan Policies in Canada.
Col. 3 ISL code - isil - ISO 15511 code of the location
Col. 4 Location name - Name of the institution represented by the global library code

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th></th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>USM50</td>
<td>HUJI</td>
<td>Hebrew University</td>
<td></td>
</tr>
<tr>
<td>XYZ50</td>
<td>DNLM</td>
<td>National Library of Medicine</td>
<td></td>
</tr>
</tbody>
</table>

tab_mapping

Location of the table: tab directory of the library

Purpose of the table: Mapping values when creating records with the RLIN Loader (UTIL E/3)

Structure of the table:
Col.1 Holding Code
col. 2 Source Code
col. 3 Source Sub-fields
col. 4 Target Code
col. 5 Target Sub-fields
col. 6 Text Command
col. 7 Overlay Flag
col. 8 New Line Flag

- Y - for all occurrences
- N - for first occurrence of source tags

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
</tr>
<tr>
<td>UMUSI</td>
<td>950##</td>
<td>1</td>
<td>852</td>
<td>b</td>
<td></td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMUSI</td>
<td>950##</td>
<td>1</td>
<td>852</td>
<td>c</td>
<td>s/UMUSI/GEN/</td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMUSI</td>
<td>090##</td>
<td>a</td>
<td>8520</td>
<td>h</td>
<td>s////g</td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UMUSI</td>
<td>090##</td>
<td>b</td>
<td>8520</td>
<td>i</td>
<td></td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAIN</td>
<td>050##</td>
<td>a</td>
<td>8528</td>
<td>h</td>
<td></td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAIN</td>
<td>050##</td>
<td>b</td>
<td>8528</td>
<td>i</td>
<td></td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XZLA</td>
<td>050##</td>
<td>ab</td>
<td>8520</td>
<td>hi</td>
<td></td>
<td>Y N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_mapping_mng500**

Location of the table: tab directory of the BIB library

Purpose of the table: Definition of mapping values when creating records with p-manage-500

Structure of the table:
- Col. 1 Holding Code
- Col. 2 HOL/Z30 Field name. Z30 always refers to location data and HOL refers to tags in the Hol record
- Col. 3 source tag/arguments, the arguments can vary
- Col. 4 Second Source

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
</tr>
<tr>
<td>XXXX</td>
<td>Z30-SUB-LIBRARY</td>
<td>ULINC</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z30-COLLECTION</td>
<td>GEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z30-CALL-NO-TYPE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOL-852</td>
<td>050##,ab,hi</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0900,ab,hi</td>
<td>HOL-852</td>
<td>&quot;$$aExL&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
tab_match

Location of the table: tab directory of the library

Purpose of the table: Match specifications for importing files

Note that this table is used by the following services

- p-manage-36 – Check input file against database
- p-file-96 - Load YBP Records
- p-file-99 - Load MARCIVE Records
- RLIN Loader
- OCLC Loader

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Match routine code. Each routine performs a particular match. Note the reserved match code:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT:</td>
<td>This match code is used to specify the matching routines performed by the check_doc_match program.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>col. 2</th>
<th>Match program:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- match_doc_uid: Match is based on a direct index (Z11). The parameters column (column 3) should contain either the index name (column 5) or the tag code (column 1) in tab11_ind. For example, if tab11_ind has defined for the ISBN direct index:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>020 ISBN az the parameters for a match based on the ISBN can be defined as follows:</td>
</tr>
<tr>
<td></td>
<td>XXX match_doc_uid I-ISBN</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>XXX match_doc_uid T-020</td>
</tr>
<tr>
<td></td>
<td>Where I=&lt;index code&gt; and T=&lt;tag code&gt;. When using T=&lt;tag code&gt; there must be an exact match. If tab11_ind col.1 has 020##, this table must have T-020## as well.</td>
</tr>
<tr>
<td></td>
<td>- match_doc_uid_2. Match is based on a direct index (Z11). The parameters column (column 3) should contain the index name and the tag code as a single value. This will only work if they are the same (example tag 035 and index 035) in tab11_ind. For example, if tab11_ind is defined as follows for the 035 direct index:</td>
</tr>
<tr>
<td></td>
<td>035 035 the parameters for a match based on the 035 can be defined as follows:</td>
</tr>
</tbody>
</table>
XXX match_doc_uid_2 035

- **match_doc_acc**: Match is based on a headings (ACC) index. The argument defined in column 3 is a table name. This table lists the tags in the record that should be checked against the headings index.

**col. 3 Program arguments**

For **match_doc_uid** this column contains the index code or the tag code used for the direct match.

For **tab_match_acc** this column contains the table name of the table that contains the tags that should be checked against the headings index. Note that this table must be located in the library's tab directory.

**Example of the table:**

```
F99  match_doc_gen               TYPE=IND,TAG=909,SUBFIELD=a,CODE=909
YBP  match_doc_uid               I-ISBN
RLIN match_doc_uid               T-020
STIDN match_doc_uid              I-STIDN
OCLC match_doc_uid               T-010
```

**tab_match_acc**

**Location of the table: tab directory of the library**

**Purpose of the table: Fields to be checked against the headings index**

The tab_match_acc table is a sample table used to define the fields in the records that should be checked against the headings index when the Check Input File Against Database (p-manage-36) service is used or when the check_doc_match routine is performed.

The table name should be passed as a parameter for the match_doc_acc program in the tab_match table of the library's tab directory.

Note that this table is also used by the following services:

- Load YBP Records (p-file-96)
- Load MARCIVE Records (p-file-99)
- RLIN Loader
- OCLC Loader

**Structure of the table:**

**col. 1** Field Code of the fields that should be checked against the headings index.

**Example of the table:**

```
! 1
!!!!!
245##
240##
```
tab_match_script

Location of the table: tab directory of the BIB library

Purpose of the table: Match specifications for checking input file against the database

This table is used in order to find "candidate matching records" for record loading, or for union view setup of a catalog that holds multiple records for the same work.

This table's functionality can be carried out by a table of any name. The name is not set in the programs, it is set in the third column of the tab_match table when match_doc_script is set as the program in column 2.

There are two match programs available:
- match_doc_gen
- match_doc_filter_hvd

For the **match_doc_gen** match program there are 3 sections to the program arguments: TYPE, TAG + SUBFIELD, and CODE. The ACC type can have an additional TRUNCATION argument:

- **TYPE** defines the search method for finding a match:
  - TYPE = SYS: searches against DB system number, which is expressed as CODE=001
  - TYPE = IND: searches against IND Z11 index
  - TYPE = ACC: searches against the filing text field of the ACC Z01 headings index

- **TAG + SUBFIELD** relates to incoming record only. The tag content is normalized using the same filing routine that is used for IND or ACC code.

- **CODE** index name defines the code of the index that is searched in order to find database record.

- **TRUNCATION**=Y can be added to the ACC match type. If this argument is present, the match will be performed using a truncated search, i.e. the incoming record's field will be considered a match if it is contained within the heading.

The **match_doc_filter_hvd** uses the program arguments:
SE-TABLE-NAME= and MO-TABLE-NAME=.

The match procedure will use the additional table(s) registered here for more specific matching arguments. The program automatically rejects all matches if the incoming record format is not SE or BK, and automatically rejects matches if there is a mismatch on the FMT field.

For each argument in this table you can define what will happen next, depending on the number of candidate records found in database.

In column 3 you set the parameter for the number of candidate records (e.g. 20+ means more than 20, 20- means 20 or less, 0+ means more than none, 0 means none.

In column 4 you register the action:
- skip (to skip to the next match set)
• stop (to stop script execution)
• goto <xx> (to jump forwards/backwards to a different match set <xx>
• <any text> or blank acts in the same manner as skip.

Structure of the table:
Col. 1  Match set identifier
Col 2  Match program name.
Col. 3  Number of candidate records found in database. This refers to the number of records in the database that match the incoming record.
Col. 4  The action to be taken where the condition of number of candidate records is true.
Col. 5  Match Program arguments

Example of the table:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>01 match_doc_gen</td>
<td>1</td>
<td>stop</td>
<td>TYPE=SYS,TAG=035,CODE=001</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>0+</td>
<td>goto 02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02 match_doc_gen</td>
<td>1</td>
<td>stop</td>
<td>TYPE=IND,TAG=035,CODE=035</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>0+</td>
<td>goto 03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 match_doc_gen</td>
<td>1</td>
<td>stop</td>
<td>TYPE=IND,TAG=024,SUBFIELD=a,CODE=024</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>0+</td>
<td>goto 04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_merge**

Location of the table: tab directory of the library

Purpose of the table: Merge routines

The fix_doc_merge program is used to merge or overlay cataloging records according to the merging program defined in this table.

The connection from table to table in the fix/match/merge procedures is:

1) Column 1 of the tab_fix table identifies a section of tab_fix. The section is used to match the "fix_doc" parameter given in various batch services.

2) Column 1 of tab_merge should match column 3 of fix_doc. Note that the following are reserved merge routines:
• MANAGE-20
• SID
• UE11-1
• UE11-2
• OCLC

Column 2 of tab_merge defines the merge program to use:
• merge_doc_overlay - Overlays records using the tab_merge_overlay table
• merge_doc_adv_overlay - Overlays records using the tab_merge_adv_overlay table
• merge_doc_replace - Replaces contents of first record with the second, but includes all CAT fields

Structure of the table:
  col. 1 Routine name
  col. 2 Program name or action
  col. 3 Program arguments. Can contain the section identifier of tab_merge_overlay

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERLAY-01 merge_doc_overlay</td>
<td>01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERLAY-02 merge_doc_overlay</td>
<td>02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OVERLAY-03 merge_doc_overlay</td>
<td>03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HVD merge_doc_adv_overlay</td>
<td>01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRCV merge_doc_overlay</td>
<td>02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_merge_adv_overlay**

Location of the table: tab directory of the library

Purpose of the table: Advanced routines for the merging of document records

The tab_merge_adv_overlay table is used to define which fields are retained from which record when merging of two records occurs. The table and its section are called from the tab_merge table, when merge_doc_adv_overlay is the program defined in column 2 in tab_merge.

The table functions for the same purpose and in a similar manner to tab_merge_overlay, with added sophistication. The additional functionality is based on knowing which record is "preferred" when the merge is performed. When merge_doc_adv_overlay is chosen from tab_merge, the system first consults tab_preferred to set the "preferred" program that will be used (e.g. preferred_doc_cdl) and the accompanying "weights" table (e.g. union_preferred) that is used to evaluate the two records. This evaluation decides which of the two records is "preferred".

The fact that one record is preferred over another record has no direct effect on the merge/overlay. The effect it has depends on the setup of this table. If the system
cannot evaluate which record is preferred, the database record is set as the preferred record, by default.

Structure of the table:

Col. 1  Merge set: Two digits that define the lines that make up a single merge set. This merge set parameter is used in two tables:
   • col.3 of tab_merge
   • col.3 of tab_fix, when the fix_doc_merge is defined as the fix program to use

Col. 2  Merging direction:
   • 1: refers to the database record
   • 2: refers to the record in the incoming load file

Col. 3  Merging direction:
   • Y: defines lines for the preferred record.
   • N: defines lines for the non-preferred record
   This is used in conjunction with use of the tab_preferred table, which is used as the starting point for setting which of the two records (database or incoming) is the preferred record.

Col. 4  Action:
   Final form of document will include fields from the database record and the incoming record. Which fields are chosen and which are dropped depends on the following action values:
   • Y
     o For database record (1) - retains field
     o For incoming record (2) - copies field
   • N - Does not retain field
   • C - Retains field only if the field does not occur in the other record
   • U - Retains field only if the field data does not appear in the other record. This action can be used only for fields that are indexed in tab11_acc or tab11_ind. The text is normalized before comparison, using the filing procedure defined for the field.

Col. 5  Field tag and indicators.
   This column is used to define the tag to compare.
   Additional parameters, for sub-field and contents or external table to match on, are defined in columns 6 and 7. Note that specifying sub-field-content is optional;

Col. 6  Subfield code

Col. 7  Content parameters for the match. The parameters can be set as follows
   
   "+|-,T,<table name>" , or
   "+|-,V,<value1> [,<value2>] [,<value3>] ...
   + (occurs) or - (does not occur)
   T (following value is a table name)
   V (following value is an actual value).
Examples:

01 1 # Y ###### 5 +,V,usci
The above line defines that all fields in the database record that have subfield 5 with value "usci" are retained

01 # # N ###### 5 +,T,list_of_values
The above line defines that all fields that have subfield 5 containing a value listed in the "list_of_values" table are removed.

Example of the table:

```
! 1234 5 6 7
!!--!!--!!!--!---!!!!!!!>
01 1# Y Y ######
! if db record has field with subfield 5=usci, always retain
01 1 # Y ###### 5 +,V,USCI
! prefer vernacular pair
01 1 # Y ###### 6 +,V,01,02,03,04,05
! retain 001 from db
01 1 # Y 001
! if non-preferred has 006, and preferred does not have 006, take 006
01 # N C 006##
01 # N C 010##
01 # N C 5050#
01 # N U 035##
```

**tab_merge_overlay**

Location of the table: tab directory of the library

Purpose of the table: Fields to retain when overlaying (merging) catalog records

The table defines the fields to be retained without change, when overlaying cataloging records, (copy record/paste record).

Example of the table:

```
01 1 Y ######
01 1 N OWN##
01 1 C 245##

01 2 Y 245##
```

In this example:

- all fields are taken from the first document except the OWN field.
- 245 is always taken from the copied record. If the copied record does not have a 245, the 245 of the original record is retained. Otherwise it is overlaid from the second to the first document
Note that the search for the code is sequential. For example:

```
01 1 N OWN##
01 1 Y ####
```

At first, the system will not take the OWN field (because of the N in col. 3 for the OWN field). Then, the system "sees" the next line, which says to take all fields. The result will be that the OWN field will be taken, too.

Structure of the table:

| col. 1 | Merge set: two digits that define the merging set. **Column 3 of the tab_fix** table is used to define the merging routine that is performed by the fix_doc_merge program. Merging routines are defined in the tab_merge table. For example, if **column 3 of the tab_fix** table has OVERLAY-01 attached to the fix_doc_merge program, then the records are overlaid or replaced according to the definitions of the tab_merge table attached to, in this case, OVERLAY-01. **Column 3 of the tab_merge** table contains the merging set that is performed when the routine - OVERLAY-01 - is selected and it matches the values of the merging set defined in this column (e.g. 01) |
| col. 2 | Merge direction |
| col. 3 | Action: Y/N/C |
| col. 4 | Tag code and indicators. This column can also be used to |

- **1** - refers to the database record
- **2** - refers to the record in the incoming load file
- When setting a "preferred" record before merge is performed (e.g. p-manage-38 with tab_preferred),
  - **1** - refers to the preferred record
  - **2** - refers to the non-preferred record
If there is no tab_preferred, or if the records are of equal weight, the database record is taken as the preferred record (1). Using tab_merge_adv_overlay allows setting both database/incoming and preferred/non-preferred.
- When performing COPY/PASTE RECORD in the Cataloguing GUI interface,
  - **1** - refers to the record into which record is copied
  - **2** - refers to the record from which record is copied

Final form of document will include fields from the original record, together with fields from the copied record, depending on:

- **Y** - For original record (1) - retains field
- For copied record (2) - copies field
- **N** - Does not retain field
- **C** - Retains field only if it does not appear in the other document
define sub-field and contents to match on. Example:

01 2 Y 590##.5,*abc*

In this case tag 590 is disregarded if sub-field $5 of the field does not contain the string "abc" as part of its contents. Note that specifying sub-field-content is optional.

Example of the table:

<table>
<thead>
<tr>
<th>Format number</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 1 N #####</td>
<td></td>
</tr>
<tr>
<td>01 1 Y LDR</td>
<td></td>
</tr>
<tr>
<td>01 1 Y 001</td>
<td></td>
</tr>
<tr>
<td>01 1 Y 09###</td>
<td></td>
</tr>
<tr>
<td>01 1 Y 5###</td>
<td></td>
</tr>
<tr>
<td>01 1 Y 79###</td>
<td></td>
</tr>
<tr>
<td>01 1 Y 9###</td>
<td></td>
</tr>
<tr>
<td>01 1 Y OWN</td>
<td></td>
</tr>
</tbody>
</table>

**tab_mimetype**

Location of the table: alephe/tab directory

Purpose of the table: Extension definitions for use in save-function in the OPAC.

This table controls which file extension will be used if an OPAC-user locally saves records in a certain document format. For example you may want that all documents in format 777 will get extension .end so that it later can be used in Endnote or other citation management products. If a format is not defined in this table .sav will be used as an extension by default.

Sending the correct file extension to the client is important so it knows how to handle the content of the file.

Defined file extensions that are used in this table should also be defined in file mime.types (in $httpd_root/conf) so that the extension is known to the Apache Server.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1 Format number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2 Extension</td>
</tr>
</tbody>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>Format number</th>
<th>Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>777</td>
<td>end</td>
</tr>
</tbody>
</table>

**tab_month**

Location of the table: alephe/tab directory

Purpose of the table: Month codes and abbreviations
This table allows for the definition of the names of the months for use in the system in any environment where the name rather than the number of the month will be displayed.

Structure of the table:
- col. 1 Language
- col. 2 Month numeric code
- col. 3 Month abbreviation
- col. 4 Month name

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>ENG 01 Jan January</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG 02 Feb February</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG 03 Mar March</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG 04 Apr April</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG 05 May May</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG 06 Jun June</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENG 07 Jul July</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_move_record**

Location of the table: tab directory of the BIB library

Purpose of the table: List of programs for moving records

This table is used to define the moving routines that are performed when records are moved through the Overview Tree in the Cataloging module.

Following are the available moving programs:

**move_adm_to_adm:** moves all instances under an ADM record to another ADM record. This move includes:
- Items + history (Z30)
- Holding requests + history (Z37)
- Loans + history (Z36)
- Photocopy requests (Z38)
- Advance booking - time slots (Z320)
- Short loans - status (Z321)
- Serials claim (Z20)
- Linked HOL records
- Subscription information (Z16)
- Routing lists (Z18)
- Members of routing lists (Z14)
- Routing trace (Z22)
- Orders (Z68)
- Order and subscription log (Z71)
• Arrival information (Z78)
• Acquisition claims (Z501)
• Transactions - Budget (Z601)
• Invoice - Line item (Z75)

The move is not performed if:
• One of the items is linked to an incoming or to an outgoing ILL request.

**move_z30_to_adm**: moves a selected item to another ADM record. This move includes:
• Items + history (Z30)
• Holding requests + history (Z37)
• Loans + history (Z36)
• Photocopy requests (Z38)
• Advance booking - time slots (Z320)
• Short loans - status (Z321)
• Serials claim (Z20)

The move is not performed if:
• The item is linked to an incoming or to an outgoing ILL request.
• The item is linked to an HOL record
• The item is linked to a subscription record
• The item is linked to an order record

**move_z16_to_adm**: moves a selected subscription to another ADM record. This move includes:
• Subscription information (Z16)
• Items + history (Z30)
• Routing lists (Z18)
• Members of routing lists (Z14)
• Routing trace (Z22)

The move is not performed if:
• The item is linked to an incoming or to an outgoing ILL request.
• The item is linked to an HOL record
• The item is linked to an order record

**move_z68_to_adm**: moves a selected order to another ADM record. This move includes:
• Orders (Z68)
• Items + history (Z30)
• Order and subscription log (Z71)
• Arrival information (Z78)
• Acquisition claims (Z501)
• Transactions - Budget (Z601)
• Invoice - Line item (Z75)

The move is not performed if:
• The item is linked to an incoming or to an outgoing ILL request.
• The item is linked to an HOL record
• The item is linked to a subscription record
move_items_to_adm: moves all items under the selected items node to another ADM record. The move includes all instances specified under move_z30_to_adm. This program should be used with ITEMS in column 1.

move_copies_to_adm: moves all subscriptions under the selected subscriptions node to another ADM record. The move includes all instances specified under move_z16_to_adm. This program should be used with COPIES in column 1.

mover_orders_to_adm: moves all orders under the selected orders node to another ADM record. The move includes all instances specified under move_z68_to_adm. This program should be used with ORDERS in column 1.

Note that if the relevant program for a particular move is not listed in this table, then an error message is displayed in the GUI when trying to perform the selected move.

Structure of the table:

Col. 1  Moving from Record
Available options are: ADM, Z30, Z16, Z68, ITEMS, COPIES, ITEMS

Col. 2  Moving to Record; available options are: ADM

Col. 3  Moving Procedure

Example of the table:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>!!!!</td>
<td>!!!</td>
<td>!!!</td>
</tr>
<tr>
<td>ADM</td>
<td>ADM</td>
<td>ADM</td>
</tr>
<tr>
<td>Z30</td>
<td>ADM</td>
<td>ADM</td>
</tr>
<tr>
<td>Z16</td>
<td>ADM</td>
<td>ADM</td>
</tr>
<tr>
<td>Z68</td>
<td>ADM</td>
<td>ADM</td>
</tr>
<tr>
<td>ITEMS</td>
<td>ADM</td>
<td>ADM</td>
</tr>
<tr>
<td>COPIES</td>
<td>ADM</td>
<td>ADM</td>
</tr>
<tr>
<td>ORDERS</td>
<td>ADM</td>
<td>ADM</td>
</tr>
</tbody>
</table>

move_adm_to_adm
move_z30_to_adm
move_z16_to_adm
move_z68_to_adm
move_items_to_adm
move_copies_to_adm
move_orders_to_adm

Note that all the error messages related to tab_move_record is located in $alephe_error_eng/move_record.

**tab_oclc**

Location of the table: alephe/tab directory

Purpose of the table: OCLC server configuration table

This table allows for configuring defaults for loading data from the OCLC server.

Structure of the table:

- col. 1  OCLC server port number
- col. 2  Record type, AUTHority or BIBliographic
col. 3  Destination library for incoming records  
col. 4  Fix procedure - in addition to the OCLC procedure.  
col. 5  Document insertion fix procedure. This column can include up to 5 special fixes. Currently we provide 2 fixes:  
  • 1. Inserts the data from column 10 to the OWN field.  
  • 2. Modify/create 001 field with data from column 10 and the record system number in the following way:  
    o Column 10 data-system number;  
    o If column 10 is empty, all above will be done with column 2 of tab_z30. If empty, no OWN field will be created.

col. 6  Item creation flag  
col. 7  HOL creation flag  
col. 8  Merge routine (specified in tab_merge)  
col. 9  Match section (as specified in tab_match)  
use only the first 5 positions of this column - because tab_match section is only 5 characters  
col. 10  OWN field value.  
  The value in this column is used as a filter for record match. This might be needed for setup in which libraries share a common bibliographic database, but catalog separate records.

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>5771</td>
<td>BIB</td>
<td>USM01</td>
<td>12</td>
<td>Y</td>
<td>Y</td>
<td>OCLC</td>
<td>OCLC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5771</td>
<td>AUT</td>
<td>USM10</td>
<td></td>
<td>N</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Note that it is possible to assign more than 5 different OWN values of cataloging records to a user's OWN value by using the hash (#) character as a wildcard. Following is a sample of the table in which the # sign is used to cover more OWN values:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CAT</td>
<td>ME###</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>CAT1</td>
<td>####</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>CAT2</td>
<td>#######</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the above sample:

**ME###** includes, for example, MEDUC, MELEC, etc.

### includes all OWN values that are up to five characters.

###### includes all possible OWN values (this is equal to the GLOBAL authorization).

The filter - if activated - affects the following displays:
- ITM (HOL) and ITML in the full + link display
- navigation tree (HOL)
- tab "HOL records" in cataloging

Structure of the table:
- Col. 1 User's OWN. This column contains the value of the cataloging OWN Permission field assigned to the user(s).
- Cols. 2-6 Record's OWN. Columns 2 to 6 contain the record's OWN values which the user with the OWN permission defined in column one is allowed to update.

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MOZART</td>
<td>HAYDN</td>
<td>HAYDNJ</td>
<td>BACH</td>
<td>BACHJ</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>HAYDN</td>
<td>HAYDNJ</td>
<td>BACH</td>
<td>BACHJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>HAYDNJ</td>
<td>HAYDN</td>
<td>BACH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BACH</td>
<td>BACHJ</td>
<td>HAYDNJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BACHJ</td>
<td>BACH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>EDUC5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>YOHANAN</td>
<td>UARCV</td>
<td>UEDUC</td>
<td>UELEC</td>
<td>ULINC</td>
<td>ULAW</td>
</tr>
</tbody>
</table>

**tab_photo_request**

Location of the table: tab directory of the **ADM** library

Purpose of the table: checks for photocopy requests

This table is used to define which checks should be performed by the system when a photocopy request is placed. It is similar in functionality to **tab_hold_request**. In some
cases, the checks on the photo requests for the WEB OPAC determine whether the photo link appears on the item line.

Structure of the table:
- **col. 1** defines when check is performed
- **col. 2** defines check procedure to perform; most procedures are relevant for POST

Types of checks:
- **PRE** - Web OPAC, before showing request link, usually does not depend on patron identification
- **POST** - Web OPAC, after filling request form
- **CIRC- GUI** client (patron identification is always present)

The following checks can be performed:

<table>
<thead>
<tr>
<th>Type of check</th>
<th>Check Procedure</th>
<th>Explanation</th>
<th>Heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRE a</td>
<td>tab15.&lt;lng&gt; – column 9=Y (photocopy request is available)</td>
<td>1200</td>
<td></td>
</tr>
<tr>
<td>PRE b</td>
<td>tab38 – sub-library and item status are listed (item can be requested)</td>
<td>1205</td>
<td></td>
</tr>
<tr>
<td>POST c</td>
<td>z305 - photo-permission Y/N</td>
<td>1210</td>
<td></td>
</tr>
<tr>
<td>POST r</td>
<td>z305 - Patron expiry date check</td>
<td>1165</td>
<td></td>
</tr>
</tbody>
</table>

Example of the table:

```
! 1 2

PRE check_photo_request_a
PRE check_photo_request_b
PRE-GUI check_photo_request_a
PRE-GUI check_photo_request_b
POST check_photo_request_c
POST check_photo_request_f
POST check_photo_request_r
```

**tab_pinyin**

Location of table: tab directory of the BIB library

Purpose of table: Definition of fields for fix_doc_add_pinyin_check_sub9 and fix_doc_add_pinyin_insert_sub9

This table is consulted to determine the fields for which fix_doc_add_pinyin_check_sub9 and fix_doc_add_pinyin_insert_sub9 should be performed.
The fix_doc_add_pinyin programs act on the fields defined, if the content is CJK. The programs take the content of $a$ and create a parallel $9$ subfield in pinyin, using 

chi_segmentation (Z113) and pinyin translation (Z114).

The _check program can only be used in the cataloging module, with cataloger intervention. In this program, in cases where a character has more than one pinyin option, the created subfield contains <option1,option2,...>. The cataloger can decide which to use, deleting the others.

The _insert program chooses <option1> if there is more than option for pinyin.

This table is limited to 200 lines.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Document record field code. # can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (e.g., 245## for 2451, 2452, 24501)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Not used; Subfield code</td>
</tr>
</tbody>
</table>

For _sub9 only subfield $a$ is relevant, and it does not have to be entered.

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>!1</td>
</tr>
<tr>
<td>100##</td>
<td>100##</td>
</tr>
<tr>
<td>245##</td>
<td>245##</td>
</tr>
<tr>
<td>260##</td>
<td>260##</td>
</tr>
<tr>
<td>440##</td>
<td>440##</td>
</tr>
<tr>
<td>600##</td>
<td>600##</td>
</tr>
<tr>
<td>700##</td>
<td>700##</td>
</tr>
</tbody>
</table>

**tab_preferred**

Location of the table: tab directory of the library

Purpose of the table: Defines settings for preferred records when merging document records

This table defines – together with its associated program – the settings for the preferred record when merging two documents.

It is used when merge_doc_adv_overlay is chosen in the tab_merge table as the program to use when merging records.

The tab_preferred table is used to set the program and the weighting table to use in order to set which of two matching documents is the preferred document.
The only program currently available is preferred_doc_cdl.

The first column is used for listing the identifier that is used to match the merge type parameter input in the p_manage_38 (Check Input File Against Database) service.

This parameter is also used in the first column of tab_merge.

Structure of the table:
- Col. 1 Section name identifier. This identifier is referred to from the p_merge_type parameter in p-manage-38.
- Col. 2 Program name.
- Col. 3 Program arguments. Currently the name of the weighting table to be used.

Example of the table:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1!</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AD-OVERLAY</td>
<td>preferred_doc_cdl</td>
<td>union_preferred</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HVD</td>
<td>preferred_doc_cdl</td>
<td>union_preferred</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_rlin**

Location of the table: tab directory of the BIB library

Purpose of the table: RLIN server configuration table

Structure of the table:
- Col. 1 Directory code
- Col. 2 Record type, AUTHority or BIBliographic
- Col. 3 Destination library for incoming records
- Col. 4 Fix procedure in addition to RLIN fix which will be run automatically
- Col. 5 Item creation flag (Y/N)
- Col. 6 HOL creation flag (Y/N)
- Col. 7 Match section as defined in tab_match
- Col. 8 Merge type

Example of the table:

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1!</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>!-!!-!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>BIB</td>
<td>USM01</td>
<td>Y</td>
<td>Y</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>AUT</td>
<td>USM10</td>
<td>N</td>
<td>N</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>BIB</td>
<td>USM01</td>
<td>Y</td>
<td>Y</td>
<td>RLIN</td>
<td>RLIN</td>
<td></td>
</tr>
</tbody>
</table>
tab_service

Location of the table: tab directory of the library

Purpose of the table: Table for defining services in the WEB OPAC

This table defines the services available for the fields displayed through edit_doc_999 in the WEB OPAC. The services are defined for a field and are:

- **scan_acc and scan_ind**
  activates display of a browse list. By default, all the browse lists defined for the field in tab11_acc and tab11_ind are displayed to choose from. This can be limited by entering the lists to include in the service in col. 3.

- **find**
  activates the find command. All the browse lists defined for the field in tab11_acc are displayed to choose from.

- **find_m_acc**
  activates find command on multiple databases. All the browse lists defined for the field in tab11 are displayed to choose from. The browse lists in the various databases must be defined with the same code. The find is performed on the databases defined in the "www_parallel_search_base" environment variable in the www_server.conf configuration file. In order to ensure that long titles are retrieved, the AUTO-TRUNCATE-Z01-FIND switch in tab100 should be set to "Y".

- **aut**
  activates display of the authority record to which the field heading is linked

- **aut_bib**
  used only for AUT library. This service offers the following functionality:
  The patron can initiate a search in an AUT database to find a relevant record. From this record, the aut_bib service can be used in order to create a set of the relevant BIB records, that are indexed using the same text as one of the AUT fields. This service initiates performing a FIND query from the AUT record on a SCAN (Z01) list in the BIB library. The parameters for the query (which BIB library and BIB SCAN list) are defined in the www_a_<lng>/service-aut-bib html page. The FIND command is truncated at 50 characters and normalized using filing procedure 99. This is particularly useful for libraries which use UDC/DDC classification for subject indexing of BIB records, and an AUTHority database in which each UDC/DDC is an authority record with textual descriptor. When the FIND command is executed and the resulting display is presented to the patron, the database environment is automatically switched to the BIB database. In order to initiate a new AUT search, the patron will have to change database. Therefore, for libraries that use this service, it is recommended that a link to the AUT database be added to the html "banner".

- **engine**
  activates an external search engine, using details defined in the tab_z121 table.
Structure of the table:

Col. 1 Tag+Indicators of the record field
Col. 2 Service
Col. 3 Parameters for the service.

For SCAN and FIND, parameters can be the Z01 lists that can be used for the service

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_find_acc</td>
<td>AUT</td>
</tr>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_scan_acc</td>
<td>AUT</td>
</tr>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_find_m_acc</td>
<td>AUT</td>
</tr>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_aut</td>
<td>AUT</td>
</tr>
<tr>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>#</td>
<td>www_f_service_find_acc</td>
<td>AUT,TIT</td>
</tr>
<tr>
<td>7</td>
<td>#</td>
<td>www_f_service_scan_acc</td>
<td>AUT,TIT</td>
</tr>
<tr>
<td>7</td>
<td>#</td>
<td>www_f_service_aut</td>
<td>AUT</td>
</tr>
</tbody>
</table>

Example of the tab_service table as defined in usm10:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_find_acc</td>
<td>PER,COR,MET,TOP,TIT,GEO</td>
</tr>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_scan_acc</td>
<td>PER,COR,MET,TOP,TIT,GEO</td>
</tr>
<tr>
<td>1</td>
<td>#</td>
<td>www_f_service_aut_bib</td>
<td>PER,COR,MET,TOP,TIT,GEO</td>
</tr>
<tr>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>#</td>
<td>www_f_service_find_acc</td>
<td>PER,COR,MET,TOP,TIT,GEO</td>
</tr>
<tr>
<td>4</td>
<td>#</td>
<td>www_f_service_scan_acc</td>
<td>PER,COR,MET,TOP,TIT,GEO</td>
</tr>
<tr>
<td>4</td>
<td>#</td>
<td>www_f_service_aut_bib</td>
<td>PER,COR,MET,TOP,TIT,GEO</td>
</tr>
</tbody>
</table>

**tab_sfx**

Location of the table: tab directory of the BIB library

Purpose of the table: control under which conditions the SFX button will appear on the Web OPAC.

This table is used to set the icon that will be displayed for the SFX link in the Web OPAC. The table can be used to define what SFX services should be available for a record that was found on a specific base, and which icon will be displayed to signify that the services are available.

Structure of the table:

Col. 1 Base name. The base on which the search was conducted
Col. 2 SFX button name. The icon that will be displayed if the record has the required SFX services available. The icon name must be a valid name
Col. 3  SFX button identifier. The index on the HTML page to which the button will be inserted.

Col. 4  Direct link - The window type that will be opened for the link

Col. 5  Service name. The service name that must be present in order for the link to be active. Note that if a few services are specified under the same identifier (column 1), at least one of them can be present in order the link to be active. Below is a list of valid service names (all case-sensitive):

- getWebService
- getWebSearch
- getTOC
- getSubject
- getStatus
- getReference
- getPayPerView
- getMessageNoFullTxt
- getHolding
- getFullTxt
- getDOI
- getDocumentDelivery
- getCitedReference
- getCitedRecord
- getCitedJournal
- getCitedGenome
- getCitedBook
- getCitedAuthor
- getBookReview
- getAuthorEmail
- getAuthor
- getAlikeRecord
- getAbstract

Example of the table:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>USM01</td>
<td>www_f_sfx_cj</td>
<td>Y</td>
<td>getCitedJournal</td>
<td></td>
</tr>
<tr>
<td>USM01</td>
<td>www_f_sfx_ft</td>
<td>N</td>
<td>getFullTxt</td>
<td></td>
</tr>
<tr>
<td>USM01</td>
<td>www_f_sfx_ae</td>
<td>Y</td>
<td>getAuthorEmail</td>
<td></td>
</tr>
</tbody>
</table>

In the above example, three different buttons are defined

- www_f_sfx_cj.
- www_f_sfx_ft and
for three different services:
- getCitedJournal,
- getFullTxt and
- getAuthorEmail.

Each one will appear only when its corresponding service is available via SFX. In addition, in the HTML pages, each one will be identified differently: "getCitedJournal" button place-holder is "$1100", getFullTxt button is "$1200" etc.

**tab_sip2_alert**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Self-check field alert definitions

The table defines the values for the self check fields alert and alert type for items returned at a self check station. Alert and alert type are defined dependent on item process status, sublibrary of item and pickup location of hold request (if the item has a hold).

Example of a setup for a self check machine located in sublibrary ULINC:

```
CT ##### ##### Y 99
## ULINC N 00
## ##### Y 04
## ##### ULINC Y 01
## ##### ILLDT Y 03
## ##### ##### Y 02
```

- All items with process status CT will get an alert with alert type 99.
- The second and third line define the handling of items without holds (col. 3 is blank): No alert for items from sublibrary ULINC, all items from other sublibraries will get an alert with alert type 04 (sent to other branch).
- Lines four to six define the handling of items with holds:
  - If the pickup location is ULINC the item will get an alert with alert type 01 (hold for this branch),
  - for pickup location ILLDT the alert type is 03 (hold for ILL).
  - For all other pickup locations except ULINC or ILLDT the item will get an alert with alert type 02 (hold for other branch).

**Note:**

If you are using self check machines in several branches you have to set up this table separately for every branch. Add the port number of the SIP2-Server used for the branch to the name of the table: tab_sip2_alert.<port>

**Structure of the table:**
- Col. 1  Item process status
- Col. 2  Sublibrary of the item;
Col. 3  Pickup location of a requested item  
Col. 4  Alert  
Col. 5  Alert type:  
  • 00 - undetermined  
  • 01 - hold for this branch  
  • 02 - hold for other branch  
  • 03 - hold for ILL  
  • 04 - send to other branch  
  • 99 - other

**tab_sip2_conf**

Location of the table: tab directory of the ADM library  
Purpose of the table: Self-check configuration  
This table defines configuration elements for the SIP2 Self Check.  
Example of the table:

```plaintext
[HOST]  
error_correction = N  
institution_id = ALEPH  
online_status = Y  
checkin_ok = Y  
checkout_ok = Y  
stat_update_ok = Y  
offline_ok = N  
desensitize = Y  
resensitize = Y  
alert = Y  
sc_language = 001  
pin_required = N  
m_match_id_type = 01
```

**tab_sip2_translate**

Location of the table: tab directory of the ADM library  
Purpose of the table: Self-check field definitions  
This table defines the expansion for self-check fields:  
Structure of the table:  
  Col. 1  Type of line:  
  • MEDIA-TYPE - enumerated material type  
  • STATUS - circulation status
Col. 2  self check definition
Col. 3  ALEPH definition
Col. 4  Description

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEDIA-TYPE</td>
<td>000</td>
<td>other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIA-TYPE</td>
<td>001</td>
<td>BOOK</td>
<td>book</td>
<td></td>
</tr>
<tr>
<td>MEDIA-TYPE</td>
<td>002</td>
<td>ISSUE</td>
<td>magazine</td>
<td></td>
</tr>
<tr>
<td>MEDIA-TYPE</td>
<td>003</td>
<td></td>
<td>bound journal</td>
<td></td>
</tr>
<tr>
<td>MEDIA-TYPE</td>
<td>004</td>
<td></td>
<td>audio tape</td>
<td></td>
</tr>
<tr>
<td>MEDIA-TYPE</td>
<td>005</td>
<td></td>
<td>video tape</td>
<td></td>
</tr>
<tr>
<td>MEDIA-TYPE</td>
<td>006</td>
<td></td>
<td>CD/CDROM</td>
<td></td>
</tr>
</tbody>
</table>

...!

| ITEM-STATUS | 01 | other |
| ITEM-STATUS | 02 | on order |
| ITEM-STATUS | 03 | available |

**tab_sort**

Location of the table: tab directory of the library

Purpose of the table: Defines fields for sorting

The tab_sort table defines the field and sub-field(s) assigned to a sort key. Up to 5 alternative field/sub-field combinations can be defined for each sort key. In this table, each of the alternatives is defined in a group of three columns. The sub-fields can be defined as "use" or "remove", where a minus sign is used to denote "remove". The last two columns (18 and 19) are used for defining positions from a fixed field. This table, used in conjunction with:

- tab_sear.<lng> (option SO)
- www_tab_short.<lng> (for WEB OPAC)
- sort option in www_server.conf
- sort option in pc_server_defaults
- tab00.<lng>.lng - Access file codes and names (for building z101 sort keys according to filing procedure).

Structure of the table:

Col. 1  Sort key number (99 different sort options may be defined)
Col. 2  Filing procedure as defined in tab_filing Used for building the filing key for sort key (Z101). The code entered here should match the desired filing procedure.
Col. 3  Field to be taken for sorting. Wild card may be used.
Col. 4  Sub-field to be taken for sorting. If left blank the entire field is taken.
Col. 5  Non-filing indicator for field (1/2)
Col. 6-17
2nd – 4th alternative field codes + sub-field + non-filing indicator

Col. 18 For fixed fields, starting position for sorting information; enter 00 for variable fields

Col. 19 For fixed fields, number of characters to take for sorting of field; enter 00 for variable fields

Example of the table:

<table>
<thead>
<tr>
<th>Col</th>
<th>Field Code</th>
<th>Subfield</th>
<th>Non-filing Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>95 008</td>
<td>260## c</td>
<td>08 0</td>
</tr>
<tr>
<td>02</td>
<td>01 1#### a</td>
<td></td>
<td>00 0</td>
</tr>
<tr>
<td>03</td>
<td>11 245## a</td>
<td>2 240##</td>
<td>2</td>
</tr>
<tr>
<td>04</td>
<td>22 050##</td>
<td>LOC## hjl</td>
<td>00 0</td>
</tr>
<tr>
<td>05</td>
<td>11 440## av</td>
<td></td>
<td>00 0</td>
</tr>
</tbody>
</table>

**tab_sub_lib_sort**

Location of the table: tab directory of the BIB library

Purpose of the table: Sub-libraries list sort types

This table controls the order of the list of sublibraries in the following ALEPH functions:

- WEB edit_doc_999.<lng> display for ITM3 and ITM4 lines:
  - WWW-FULL-ITM3
  - WWW-FULL-ITM4
- WEB Holdings screen - The list of sub-libraries in the combo-box:
  - WWW-ITEM-LIST-FILTER
- www_tab_short.<lng> - columns ITM,ITM+, LOC, LOC+, PSTS
  - WWW-SHORT
- list of item locations in SEARCH GUI full record display:
  - PC-FULL-ITM
- list of cash owing in WEB OPAC->USER->Cash Transactions:
  - WWW-BOR-CASH
- Cash list divided by sub-libraries in the GUI:
  - CASH-LIST
- Vendor list, order units in the GUI:
  - VENDOR-LIST

The sort identifiers are:

- 00 - by sublibrary name, taken from tab_sub_library.<lng>
- 01 - by profile. This sets which sublibrary will display first in the list, after which the sort is by sublibrary name. Not relevant for GUI: PC-FULL-ITM, VENDOR-LIST, CASH-LIST
- 02 - by profile to determine the first sublibrary, and then by the tab_item_list_order table. The table can have .<base extension>, in which case there can be multiple tables. The system will choose the table that matches the logical base that is in effect. If there is no matching .<base extension> table, the system will use the generic tab_item_list_order table. Relevant only for: WWW-FULL-ITM3, WWW-FULL-ITM4, WWW-SHORT, WWW-ITEM-LIST-FILTER, PC-FULL-ITM

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Function name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Sort: option:</td>
</tr>
</tbody>
</table>

- 00 - by library name
- 01 - by library name, with "preferred library" (determined by profile) in first position
- 02 - by order of tab_item_list_order, with "preferred library" (determined by profile) in first position.

Example of the table:

```
! 1 2
!!!!!!!!!!!!!!!!!!!-!!
WWW-FULL-ITM3 00
WWW-FULL-ITM4 00
WWW-SHORT 00
WWW-BOR-CASH 01
WWW-ITEM-LIST-FILTER 00
PC-FULL-ITM 00
VENDOR-LIST 00
CASH-LIST 00
```

**tab_sub_library.<lng>**

Location of the table: alephe/tab directory

Purpose of the table: List of library code and sublibrary codes

This table defines the sub-libraries of all the ADM libraries on the server. In the list of items display in the ALEPH WEB OPAC, it is possible to gather together all the sub-libraries in one list, regardless of the connection to separate ADM libraries. For all other ALEPH functions (USER, CIRC, ITEMS, SERIALS, ILL, ACQ) it is required that the user connect to a specific ADM library (e.g. USM50).

The number of sublibraries in the table is not limited. However, there are places in the system where the total number of sublibraries relevant to the situation cannot exceed 200. For example, when sorting a list of items or list holding libraries for a single bibliographic record.
Connect in GUI updating modules is to an ADM library (which is a group of sub-libraries). This is a physical group, and one sub-library can be assigned to one ADM library ONLY. Each sub-library code must be unique in this table (unless a special setup for testing conversions has been installed by Ex Libris).

### Library names (col. 5):
Names of the sub-libraries display in WEB OPAC and GUI modules. For libraries that use more than one language interface, each language is a separate tab_sub_library.<lng> table. All rows and columns should be exactly the same, except for column 5 (library name). Different library names for WEB and GUI can be assigned by adding the extension PC or WWW to the table name.

For example:
- `tab_sub_library.eng.PC`
- `tab_sub_library.eng.WWW`

### Library types (col. 2):
In addition to actual sub-libraries (library type 1), some of the sub-libraries are defined for:
- patron records (type 2), in which case the code can be ALEPH or the ADM library code,
- delivery of requested material (type 3, must be coordinated with tab37 and tab38), and for
- transfer of material to a reading room (type 4, must be coordinated with tab37).

### Link to tab15.<lng> (col. 6):
The code registered in this column identifies the lines in tab15.<lng> (Item statuses) that are relevant for this sub-library. The code here must match the code in column 1 of tab15.<lng>.

This allows for the definition of statuses in tab15.<lng> that are used in common by a number of sub-libraries.

### Link to tab16 (col. 7):
The code registered in this column identifies the lines in tab16 (Fines, due dates and loan limits) that are relevant for this sub-library. The code here must match the code in column 1 of tab16.

This allows for the definition of lines in tab16 that are used in common by a number of sub-libraries.

### Link to tab17 (col. 8):
The code registered in this column identifies the lines in tab17 (Library hours) that are relevant for this sub-library. The code here must match the code in column 1 of tab17.

This allows for the definition of lines in tab17 that are used in common by a number of sub-libraries that share a set of open/close hours.

### Link to patron record (col. 9-13):
Columns 9 through 13 are used to define the preferred patron record, when matching the patron and the item to a line in tab16 (loan due dates, loan, request and renew...
limits, etc.). The five columns are alternatives, and the system checks one after the other, in an attempt to find a matching patron record.

For example, if the first of these 5 columns has sub-library XYZAB, but the patron does not have an XYZAB patron record, the next of the 5 columns will be consulted, and so on. If, in the end, no matching patron record is found the patron does not have privileges for this item, and he cannot take it on loan.

Therefore, at least one sub-library code is mandatory for sub-libraries type 1, and sub-library type 2 must have one sub-library code.

For example, for library type 1:

ULINC USM50 ALEPH
- checks for ULINC patron record, if not found checks for
- USM50 (general library patron) and lastly checks for
- ALEPH (global patron)

UHLTH
- patron can loan items from UHLTH library only

For example, for library type 2:

ALEPH
- the system uses the patron status in the ALEPH patron record in order to find the matching line in tab16 for global loan and hold limits.

Structure of the table:

| Col. 1 | sub-library |
| Col. 2 | type         |
|        |             |
|        | 1 - full sub library (patron and items) |
|        | 2 - patron sub library only (no items) |
|        | 3 - pickup location that is not a library |
|        | 4 - reading room sub library (1 + special request handling) |
|        | 5 - Ordering unit |
|        | 6 - ILL unit (for use with ISO ILL) |
| Col. 3 | ADM library |
| Col. 4 | ALPHA        |
| Col. 5 | Sub-library name |
| Col. 6 | Group ID from tab15.<lng> |
| Col. 7 | Group ID from tab16 |
| Col. 8 | Group ID from tab17 (Only for Item Sub-libraries) |
| Col. 9 | First Patron privileges record (Z305 sub-library or ALEPH). This field is mandatory. If you don't want to give alternatives put the sub-library code as listed in column 1. |
| Col. 10-13 | Second to fifth Patron privileges record |

Example of the table:
### tab_sub_library_address.<lng>

**Location of the table:** alephe/tab directory OR tab directory of the library

**Purpose of the table:** Addresses of sub-libraries for printouts

This table defines addresses of branch libraries for printouts (e.g. orders to vendors, overdue letters to patrons, etc.)

It is used in conjunction with the library's /tab/form_sub_library_address table, using column 2 as the link between them. Each address can be up to 50 characters per line, and up to 10 lines in length.

**Note:**
This table can be defined in both alephe/tab and in the tab directory of each of the ALEPH_LIBS libraries. This means that the table can be split to sub-tables.

The system checks first for tab_sub_library_address.<lng> in the library. If no file is found there, it checks for tab_sub_library_address.<lng> in alephe/tab.

**Structure of the table:**
- **Col. 1** Code (##### can be used as wildcards). Alternatively, the code of the ADM library can also be used to denote an address common to all branches.
- **Col. 2** Address type – if not defined defaults to type 1.

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Col. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALEPH</td>
<td>General Patron</td>
</tr>
<tr>
<td>USR00</td>
<td>L USR00 Library</td>
</tr>
<tr>
<td>HOME</td>
<td>L Home address</td>
</tr>
<tr>
<td>BOX</td>
<td>L In-house mailbox</td>
</tr>
<tr>
<td>ILLDT</td>
<td>L ILL Department</td>
</tr>
<tr>
<td>USM50</td>
<td>USM50 L Exlibris Demo Libraries</td>
</tr>
<tr>
<td>UARCV</td>
<td>15A UARCV UARCV UARCV ALEPH</td>
</tr>
<tr>
<td>UEDUC</td>
<td>15A 16A 17A UEDUC USM50 ALEPH</td>
</tr>
<tr>
<td>UELEC</td>
<td>15A UELEC UELEC UELEC USM50 ALEPH</td>
</tr>
<tr>
<td>URELC</td>
<td>15A UELEC UELEC UELEC USM50 ALEPH</td>
</tr>
</tbody>
</table>
• 1 - default address
• 2 - Address for Circulation forms
• 3 - Address for Acquisition forms
• 4 - Address for ILL forms

Col. 3  Address line – up to 10 lines, 50 characters each line

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!</td>
<td>!-</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
</tr>
<tr>
<td>USM50</td>
<td>Ex Libris University Libraries</td>
<td></td>
</tr>
<tr>
<td>222 Aleph Causeway</td>
<td><a href="mailto:thechoice@exlibris.co.il">thechoice@exlibris.co.il</a></td>
<td></td>
</tr>
<tr>
<td>Chicago, IL 60614</td>
<td>Tel# 773.404.5527</td>
<td></td>
</tr>
<tr>
<td>UARCV</td>
<td>Archives Department</td>
<td></td>
</tr>
<tr>
<td>Lincoln Library</td>
<td><a href="mailto:thechoice@exlibris.co.il">thechoice@exlibris.co.il</a></td>
<td></td>
</tr>
<tr>
<td>808 Log Lane</td>
<td>Chicago, IL 60614</td>
<td></td>
</tr>
</tbody>
</table>

**tab_subfield_punctuation**

Location of the table: tab directory of the library

**Purpose of the table: End-subfield punctuation definitions**

This table is used to define sub-field punctuation for fields. Punctuation for fields is necessary when the system automatically updates the bibliographic record from a linked authority record.

When the bibliographic record is updated from the authority database the system always uses the preferred term (1XX) from the authority record. Originally the bibliographic record may have more data then the authority record. This data should be retained. In MARC, authority records do not have end punctuation while bibliographic records do.

The tab_sub-field_punctuation table is used to add end punctuation to the updated field. The table can be also used to add punctuation between the end of the preferred term from the authority record and the additional sub-fields retained from the bibliographic record (e.g. between sub-field $a – personal name - and sub-field $t - title of MARC21 600 field).

**Structure of the table:**

Col. 1  Program code

• A - Punctuation for fields updated from the authority database

Col. 2  Tag + Indicators
Col. 3    Sub-field code  
Col. 4    Following sub-field code  
Col. 5    Punctuation to add  
Col. 6    If punctuation: if the field already ends with one of these 
          punctuation signs, punctuation from the previous column will not 
          be added.

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!-</td>
<td>!-</td>
<td>!-</td>
<td>!-</td>
<td>!-</td>
<td>!-</td>
</tr>
</tbody>
</table>

A 1#### a .  .
A 1#### d .  .
A 110## b .  .
A 600## a .  ) .
A 600## d ,  ,

**tab_suf_default**

Location of the table: alephe/tab directory

Purpose of the table: Shared user definitions

This table defines two things:

- which fields are shared, and which fields are local. Shared fields are 
  registered in the USR01 record, and the local record field is updated from 
  the USR01 record. Local fields are not written in the shared record.
- the default values for the fields that are not shared (i.e. local fields).

Structure of the table:

Col. 1    Internal Z30X code
Col. 2    Shared field status
          - S = for shared
          - L = for local
Col. 3    values for the local variations. In the USR01 library server, this 
          column is blank. In the ADM library server this column contains 
          the required value.

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!-</td>
<td>!-</td>
<td>!-</td>
<td>!-</td>
</tr>
<tr>
<td>!-</td>
<td>!-</td>
<td>!-</td>
<td>!-</td>
</tr>
<tr>
<td>z303-id</td>
<td>S</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
z303-primary-id | L |
z303-proxy-id | L |
z303-proxy-id-type | L |
z303-open-date | S |
z303-profile-id | L |
**tab_suf_lib**

Location of the table: alephe/tab directory

Purpose of the table: Shared user definition

The condition for the participation of an ADM library in the shared user mechanism is the presence of a definition in this table. Whenever the library is defined in the table, the SUF mechanism is applied. If the library is not defined the system assumes that the library does not participate.

The default of fields defined as local (according to the table alephe/tab_suf_default, cols. 2 + 3) will only be applied when a local record is created for the first time in the replication process. Updates will not consult this table for defaults. The contents of local fields will not be overwritten by the replication and original contents will be kept.

Structure of the table:
- Col. 1 Library code
- Col. 2 Alias code
- Col. 3 Host name

Example of the table:

```
! 1   2   3
!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
USR01 USR01 metalib:6530
USR50 USR50 metalib:6530
USR51 USR51 metalib:6530
```

**tab_tag_text**

Location of the table: tab directory of the library

Purpose of the table: Default tag values when creating records

This table allows for the defining of default tag values when creating records.

Structure of the table:
- Col. 1 Tag
- Col. 2 Record type (e.g. BK, SE, CF)
- Col. 3 Default Value

Example of the table:

```
! 1   2   3
!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
LDR BK ^^^^^nam^a22^^^^^^a^4500
```
The tab_type_config table is used to define the specifications for the creation of the new field created by the expand_doc_type expand/fix program.

This program - together with the table - can be used, for example, to create a field that contains the format of the record based on the contents of other field(s) present in the record (e.g. a combination of the LDR and the 008 field).

In the following example, the new field will contain the string 'FILM' according to a match performed on the values of both the LDR and the 008 field:

```
TYP Film LDR F06-01 EQUAL g
008 F33-01 EQUAL m
```

In the above example:
- the TYP field ($aFilm) is created when:
  - position 06 of the LDR contains a 'g' and
  - position 33 of the 008 field contains an 'm'.

Following is the structure of the new field:

```
TYP L $$aFilm
```

Note: the tab_type_config table is a sample table. If necessary, similar tables can be added and passed as parameters.

Note that in the tab_expand table, the name of the configuration file - e.g. tab_type_config - should be added as a parameter in column 3. Following is a sample of the setup needed in the tab_expand table to use the new program:

```
! 1 2 3
| !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! | | 
|-------------|-------------|-------------|
| WORD expand_doc_type | tab_type_config |
```

Note in addition that this program can be also used as a fix program to actually add the new fields to the record.
In this case, the tab_fix table should be used instead of the tab_expand table.
Following is a sample of the setup needed in the tab_fix table to use this program:
<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!&gt;</td>
<td>expand_doc_type</td>
<td>tab_type_config</td>
</tr>
</tbody>
</table>

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Target tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>Format code</td>
</tr>
<tr>
<td>col. 3</td>
<td>Format name</td>
</tr>
<tr>
<td>col. 4</td>
<td>Field tag</td>
</tr>
<tr>
<td>col. 5</td>
<td>Sub-field(s)/position</td>
</tr>
<tr>
<td>col. 6</td>
<td>Match criteria</td>
</tr>
</tbody>
</table>

The value entered in this column is expanded into sub-field $a$ of the new field created by `expand_doc_type`. If the column is left blank, then the format name (value of column 3) is added to sub-field $a$ of the new field. For example, if the table contains:

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

then a new TYP field with the following content will be added when position 06 of the LDR field contains an 'a':

```
TYP   L $$aBK$$bBook
```

If this column is left blank, the new field will be created/expanded as follows:
```
TYP   L $$aBook
```

Format name

If a format code is present (column 2), then the format name is added/expanded into sub-field $b$ of the new field. If no format code is defined, then the format name is added/expanded into sub-field $a$ of the new field. For example,

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

a new TYP field with the following structure will be added when position 06 of the LDR field contains an 'a':

```
TYP   L $$aBK$$bBook
```

If the format code column is left blank, the new field will be created/expanded as follows:
```
TYP   L $$aBook
```

Field tag

Field from the record used for determining the material type that is expanded into the new field. In the following line, the LDR (position 06 with 'a') is used to define that the record is for a book:

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

Sub-field(s)/position

The sub-field codes or the fixed field position and length (of the field defined in column 4) to be checked

For e.g., the program checks position 06 of the LDR for one position

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

Or the program checks sub-field $a$ of the 490 field:

```
TYP   Thesis                 4901# a MATCH masters*
```
The match criteria in relation to the contents of the sub-field or the fixed field positions defined in column 5. The available options are:

- EQUAL
- N-EQUAL
- EXIST
- N-EXIST
- MATCH
- N-MATCH

Col. 7 Contents of the field or of the fixed field position that are used to match on (according to the match criteria defined in column 6). Use [ ] (square brackets) to enclose multiple values to match on. The relationship between the values is of type OR.

In the following line, the match is based on values 'e' or 'f' of position 06 of the LDR field:

```
TYP Map LDR F06-01 EQUAL [e,f]
```

Col. 8 Case-sensitive matching flag

The table is limited to 500 lines.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!!!!</td>
<td>!!!!!</td>
<td>!!!!</td>
<td>!!!!!</td>
</tr>
<tr>
<td>TYP</td>
<td>Mixed Materials</td>
<td>FMT</td>
<td>EQUAL</td>
<td>MX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>008</td>
<td>F29-01</td>
<td>EQUAL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Conference</td>
<td>111##</td>
<td>EXIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Conference</td>
<td>711##</td>
<td>EXIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Conference</td>
<td>811##</td>
<td>EXIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Conference</td>
<td>006</td>
<td>F00-01</td>
<td>EQUAL</td>
<td>a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Serial</td>
<td>006</td>
<td>F12-01</td>
<td>EQUAL</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Computer file</td>
<td>006</td>
<td>F00-01</td>
<td>EQUAL</td>
<td>s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Microform</td>
<td>007</td>
<td>F00-01</td>
<td>EQUAL</td>
<td>h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYP</td>
<td>Videorecording</td>
<td>007</td>
<td>F00-01</td>
<td>EQUAL</td>
<td>v</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_vat_percent.lng**

Location of the table: tab directory of the ADM library

Purpose of the table: Codes for VAT percentages (for ACQ invoices)

This table defines codes, code texts and VAT percentages for VAT calculation in general invoices (Z77), line items (Z75) and budget transactions (Z601).

**Note:**
To use the VAT code mechanism, set the variable UseVATCode (under section "[invoice]") to "Y" in the file \Acq\Tab\Acq.ini:
When UseVATCode is set to "Y", a VAT Code can be assigned to a general invoice by an edit field equipped with a drop-down list (appearing instead of the "VAT Percent" edit field), in the General Invoice pane.

Structure of the table:
- Col. 1: Code;
- Col. 2: VAT Percent
- Col. 3: Description

Example of the table:

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!-!!!!-</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
<td></td>
</tr>
<tr>
<td>NOVAT 00.00 No VAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BOOKS 18.00 Books VAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SERIA 16.50 Serials VAT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab_vendor_sub_lib**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Sublibrary specific and generic vendor fields

Tab_vendor_sub_lib is applicable for libraries that work with a two-level vendor. When creating/updating a sub-level vendor record, this table defines which group of fields of the Z70 table will be taken from the generic (USM50) vendor record and will be read-only for the sub-level record and which group of fields can be set specifically for the sub-level vendor record.

If $Y$ is defined in Col.2:
- the group of fields can be defined specifically for the sub-level vendor record.

If $N$ is defined in Col.2:
- the group of fields is controlled by the generic vendor record and can be read-only in the sub-level vendor record.

Following are the group definition of Z70 fields:

**Group ID: GENERAL**
- Z70-COUNTRY
- Z70-MATERIAL-TYPE
- Z70-STATUS

**Group ID: CONTACTS**
- Z70-CONTACT-n (1 to 5)
Z70-NOTE

Group ID: ACCOUNTING
Z70-TERMS-SIGN
Z70-TERMS-PERCENT
Z70-ACCOUNT-M
Z70-ACCOUNT-S
Z70-VENDOR-ACCOUNT

Group ID: ORDER-DISPATCH
Z70-DEFAULT-ORDER-DELIVERY
Z70-LE-LETTER-TYPE
Z70-LI-LETTER-TYPE
Z70-LE-SEND-METHOD
Z70-LI-SEND-METHOD

Group ID: EDI
Z70-EDI-VENDOR-CODE
Z70-EDI-VENDOR-CODE-TYPE
Z70-ED-SEND-METHOD
Z70-ED-LETTER-TYPE

Group ID: MATERIAL-DELIVERY
Z70-DELIVERY-TYPE-n (1 to 5)
Z70-DELIVERY-DELAY-n (1 to 5)

All other Z70 fields are either system set or taken from the generic vendor.

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Group ID of Z70 field</th>
</tr>
</thead>
<tbody>
<tr>
<td>col. 2</td>
<td>The group's fields are/are not read from the vendor sub-library/order unit record</td>
</tr>
</tbody>
</table>

Y=read from vendor sub-library/order unit Z70 record.
N=read from generic vendor Z70 record.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>GENERAL</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>CONTACTS-NOTE</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>ACCOUNTING</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>ORDER-DISPATCH</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>EDI</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>MATERIAL-DELIVERY</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>


**tab_version**

Location of the table: alephe/tab directory

Purpose of the table: Client definitions to access the pc_server

Defines which clients are allowed to access the pc_server. Note that definitions such as SER1.0 and other constants are used for non ALEPH GUI clients, for example self-check etc... These constants are defined by the developers of the external product and not by *Ex Libris*.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Format Type</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Compress</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Encryption</td>
</tr>
<tr>
<td></td>
<td>- 0 - none</td>
</tr>
<tr>
<td></td>
<td>- 1 - ALEPH</td>
</tr>
<tr>
<td></td>
<td>- 2 - DES</td>
</tr>
<tr>
<td></td>
<td>- 3 - Kerberos</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Encryption key</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Service Prefix</td>
</tr>
</tbody>
</table>

Example of the table:

```
! 1 2 3 4 5 6
!!!!!!-!!!!!-!-!-!!!!!!!-!!!!!!!!!!
SER1.0 ALEPH Y 0
16.02 ALEPH Y 0
...```

**tab_weights**

Location of the table: tab directory of the BIB library

Purpose of the table: Weight factors for determining candidate records

The tab_weights table is called from the tab_match_script table as a parameter for the match_doc_filter program. It performs checks on the match candidate record, and as a result of these checks, can accept or reject the candidacy. Each check can be assigned a positive or a negative value (weight). The sum of these values is compared to the threshold value. If the sum is less than the threshold value, the record is removed from the pool of candidates. If the sum is less than the threshold value, it remains in the pool of candidates.
The match_doc_filter_hvd program automatically rejects all matches if the incoming record format is not SE or BK, and automatically rejects matches if there is a mismatch on the FMT field.

First, a threshold weight is defined, with TRSLD in the first column. If the record does not meet the threshold weight, it is no longer regarded as a candidate.

The following lines define various programs, for checking a field, a subfield, or a position range in a fixed field. The outcome of each program check assigns a positive or negative value. Each program line can end in a script command for continuation. If no action is listed, the next line is performed.

Script commands for continuation are:
- empty, the default command is to skip to the next tag
- skip one line
- skip two lines
- skip three lines
- skip to next tag
- unmatch (remove record from match candidate consideration)
- continue (to next line);

Currently available programs are:
- **match_year_2**: checks that the date is within +/-2 years, or is exact match if date is less than the date given in parameters (col.4)
- **match**: for a given tag defined in col.1, the content of any occurrence of the field in the incoming record is the same as the content of any occurrence of the same tag in the database record.
- **mismatch**: content of incoming record tag and db tag are different.
- **collat_msmch**: checks for the presence of "v." in the field/subfield. If present in one and not the other, the collat_msmch check succeeds. If present in neither, or present in both, this condition fails.
- **keywords**: compares the percentage of matching keywords to the threshold percentage given as parameter in col.4. If the percentage of matching keywords passes the threshold, a weight is assigned. The weight assigned is the percentage of matching keywords multiplied by the weight parameter in col.5.
- **1st_wd_mch** - matches on the first word to occur before punctuation. For use with the imprint field. E.g. If the 260$$a contained the text Washington, D.C. the match would be on Washington only.
- **order**: program is the same as "keywords", with an additional check of word order (i.e. the keywords found are in the same order). If the keywords are not in the same order, this weight check fails. Therefore, this check should be performed BEFORE the "keywords" check.
- **edition**: examines the edition (250) field, using the following parameters in col.4:
  - BOTH_MISSING - both records lack fields
  - ONE_MISSING_1 - field is missing from one of the two records and contains no numeric characters
- ONE_MISSING_2 - field is missing in one record and contains "first" or "1st" in the other record
- ONE_MISSING_3 - field is missing in one record and contains text other than "first" or "1st" in second record
- FIRST - both records have field that contains "first" or "1st"
- NUMERIC_MATCH - both records have fields that contain numbers - and the numbers match
- NUMERIC_MISMATCH - both records have fields that contain numbers but the numbers don't match

Structure of the table:
- **Col. 1** Tag for comparison, or TRSLD. The table MUST begin with a TRSLD line, which defines the threshold weight that a record must pass in order to be considered a match. The TRSLD line has only col.5 (weight) defined.
- **Col. 2** Subfield code, or FNN-NN for fixed fields
- **Col. 3** Action – as described above
- **Col. 4** Action Parameters
- **Col. 5** Weight assigned, according to the success or failure of the action. The weight is expressed as plus or minus (+020, -010)
- **Col. 6** Script command for continuation:
  - empty, the default command is to skip to the next tag
  - skip one line
  - skip two lines
  - skip three lines
  - skip to next tag
  - unmatch (remove record from match candidate consideration)
  - continue (to next line)

Example of the table:

```
! 1 2 3 4 5 6
!!!!!!-!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!>
TRSLD +050
100## aqd match +055
100## abc mismatch -055
```

**Note:** use the keyword "NORMALIZE" in the 4th column of "match"/"mismatch" line in order to compare normalized data. In the "order"/"keywords" lines, specify the word breaking routine to use after the percent sign, separated by a comma. For example, to use normalization of the 100 field and the word breaking routine "01":

```
100## aqd match NORMALIZE +055
100## abc mismatch -055
245## abn order 33% +005
245## abn keywords 33%,01
```

© Ex Libris Ltd., 2004

Updated: 3/21/2004
**tab_word_breaking**

Location of the table: tab directory of the library

Purpose of the table: Defining word breaking procedures.

A word is a group of characters between blanks, from beginning of field to first blank and from last blank to end of field. Most of the following procedures either create blanks, or change blanks into something else (so they will not split). In addition, after the text has been broken into words, a character conversion table is used to change texts (e.g. u with umlaut to ue). The system uses the character conversion table that is listed for the WORD-FIX line in the tab_character_conversion_line table. Conversion is used for both parsing the FIND query and for word building.

**Note:** ! Word routine "90" is reserved for the system. It is used when parsing a FIND query.

A list of available procedures that can be used:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Action</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>2_hyphen</td>
<td>changes 2 adjacent hyphens to blank. Particularly necessary for 505 (contents) field, which uses -- as a separator.</td>
<td></td>
</tr>
<tr>
<td>abbreviation</td>
<td>compress a dot between single characters</td>
<td></td>
</tr>
<tr>
<td>blank_to_carat</td>
<td>change blanks to carat</td>
<td>to ^</td>
</tr>
<tr>
<td>compress</td>
<td>compress (i.e. strip) the characters listed in col. 4</td>
<td></td>
</tr>
<tr>
<td>compress_blank</td>
<td>delete blanks</td>
<td></td>
</tr>
<tr>
<td>compress_h_quote</td>
<td>strips the Hebrew Gershayim sign (&quot; - inverted commas) between two Hebrew letters, in words used as initialisms or abbreviations</td>
<td>Tana*kh - The Bible</td>
</tr>
<tr>
<td>del_sub-field</td>
<td>delete sub-field sign</td>
<td>$$x</td>
</tr>
<tr>
<td>force_delimiter</td>
<td>change sub-field sign ($$x) to blank-z-blank. This prevents words across sub-fields being considered adjacent</td>
<td></td>
</tr>
<tr>
<td>marc21_41</td>
<td>for separating languages in MARC21 field 041</td>
<td>engfreger changes to: eng fre ger</td>
</tr>
<tr>
<td>numbers</td>
<td>compress a comma and a dot between numbers</td>
<td>100,000 or 100.000 changes to 100000</td>
</tr>
<tr>
<td>split_cjk</td>
<td>adds blank between all CJK characters, therefore each character constitutes a word. CJK characters are recognized by the</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>Action</td>
<td>Example</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>2_hyphen</td>
<td>changes 2 adjacent hyphens to blank. Particularly necessary for 505 (contents) field, which uses -- as a separator.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>system as &quot;ALPHA C&quot;. Therefore, column 2 must contain &quot;C&quot; for this procedure.</td>
<td></td>
</tr>
<tr>
<td>subf_to_sign</td>
<td>change 2nd and subsequent sub-field signs to the single character listed in col. 4</td>
<td></td>
</tr>
<tr>
<td>to_blank</td>
<td>change characters listed in col. 4 to blank</td>
<td></td>
</tr>
<tr>
<td>to_blank_2</td>
<td>change characters listed in col. 4 to blank. For example: 01 # to_blank_2 ,,:;</td>
<td>Schiller, Friedrich&quot; will be changed to &quot;Schiller Friedrich&quot; if comma is listed in col. 4.</td>
</tr>
<tr>
<td></td>
<td>Note that the above is relevant only if the actual value of these characters is retained in the /alephe/unicode/unicode_to_word_gen table.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>e.g. Jones, Arthur - comma is changed to blank</td>
<td></td>
</tr>
<tr>
<td></td>
<td>one,two,three - commas are retained</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XI.1.1.3 - dots are retained. This can be particularly useful for shelf or class numbers.</td>
<td></td>
</tr>
<tr>
<td>The system automatically performs: to_lower:</td>
<td>change case to lower</td>
<td>BOOK to book</td>
</tr>
</tbody>
</table>

Note that the system automatically carries out triple posting for hyphens and apostrophes:
- as separate words
- as is (with hyphen/apostrophe)
- with hyphen/apostrophe compressed.

For example, twenty-five is indexed as:
- twentyfive
- twenty
- five
- twenty-five

**Both the hyphen and the apostrophe MUST NOT be included in any of the word breaking procedures defined in this table.**
In addition, the "hyphen" (solidus) and the apostrophe must be left with their actual value (002F) in alephe/unicode/unicode_to_word_gen file.
Structure of the table:

| col. 1 | ID - Two-digit identifier of the word breaking routine. This identifier is used in column 6 of tab11_word |
| col. 2 | ALPHA (not used) |
| col. 3 | Name of the word breaking procedure |
| col. 4 | Parameters for the word breaking procedure (when relevant) |

Example of the table:

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!</td>
<td>!</td>
<td>-</td>
<td>!!!!</td>
</tr>
<tr>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
</tr>
<tr>
<td>01 # del_subfield</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # abbreviation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # numbers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # 2_hyphen</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # compress_h_quote</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # to_blank</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # split_cjk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 # to_lower</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 # del_subfield</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 # abbreviation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03 # numbers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
... 

**Note:** If you wish to allow patrons to search for a term such as C++ the plus sign (+) should **NOT** be defined in the list of characters that converts the character to a blank.

**tab_xyz**

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** define the "XYZ" index that should be used to strip subject subdivisions.

This table is used to define the "XYZ" index that should be used to strip subject subdivisions (subfields $v, $x, $y and $z) from the bibliographic subject heading in order to find the more general term in cases where there is no match between the original subject heading from the bibliographic library and the headings in the authority library.

**Structure of the table:**

Col. 1  Headsings index code  
Col. 2  Stripped headings index used to find the more general term in cases where there is no match between the original subject heading from the bibliographic library and the headings in the authority library

**Example of the table:**

| ! | 1 | 2 |
tab_yankee_def

Location of the table: tab/import directory of the BIB library

Purpose of the table: Default values the YBP loader (p-file-96)

This table defines the default values for fields in the Item, Order and Budget records that are created using the Yankee Book Peddler loader (p-file-96).

Structure of the table:
Col. 1 Z30 (item) /Z68 (order) /Z601 (budget) field name
Col. 2 Value to match in the incoming bibliographic record or hashes to indicate that the value in column 3 will be used regardless of the bibliographic record
Col. 3 Value to place in the field defined in column 1

Example of the table:

<table>
<thead>
<tr>
<th>ORDER-SEQ</th>
<th>ybp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z30-SUB-LIBRARY</td>
<td>BXMM UMUSI</td>
</tr>
<tr>
<td>Z30-SUB-LIBRARY</td>
<td>BXMA UEDUC</td>
</tr>
<tr>
<td>Z30-SUB-LIBRARY</td>
<td>UEDUC</td>
</tr>
<tr>
<td>Z30-COLLECTION</td>
<td>BXMM GEN</td>
</tr>
<tr>
<td>Z30-COLLECTION</td>
<td>BXMA GEN</td>
</tr>
<tr>
<td>Z30-COLLECTION</td>
<td>GEN</td>
</tr>
<tr>
<td>Z30-MATERIAL</td>
<td>BOOK</td>
</tr>
<tr>
<td>Z30-ITEM-STATUS</td>
<td>42</td>
</tr>
<tr>
<td>Z30-CATALOGER</td>
<td>YOHANAN</td>
</tr>
<tr>
<td>Z30-CALL-NO-TYPE</td>
<td>0</td>
</tr>
</tbody>
</table>

tab_z0101

Location of the table: tab directory of the BIB library

Purpose of the table: Brief record structure based on format, index and field of origin

This table is used to define for each particular heading, depending on the record's format and field of origin, the structure of the brief record (Z0101). The Z0101 record has both display text and filing text fields. The filing text is constructed using filing procedure 98 from tab_filing.

Structure of the table:
Col. 1 Format code (e.g. BK for book). Enter a specific record format,
or use ## as a wild card to indicate that the definitions are appropriate for any format.

col. 2  Headsings code

Specify the headings code for which the brief specifications are relevant. For example, if the tab_z0101 table has the following line:

```
## AUT 100## TITLE-MAIN
```

Then the TITLE-MAIN text code defined in tab_z0101_text (UTIL G/1/o) is the base for the brief record created for AUT headings (originated from the 100 field and for all record formats).

col. 3  Field tags and indicators for which the brief specifications are relevant. For example, if the tab_z0101 table has the following line:

```
## AUT 100## TITLE-MAIN
```

Then the TITLE-MAIN text code defined in tab_z0101_text (UTIL G/1/o) is relevant for headings from the AUT only if they originated from a 100 field.

col. 4  The text code is used to define in which way the brief is constructed. Text codes are defined in the tab_z0101_text table. Note that the brief is actually constructed from the text codes defined in columns 4, 5 and 6.

col. 5-8  Text code (see explanation for column 4).

col 9  AUT match code. Used to define the text that is used for searching a match in the authority database. Note that the system searches for a match in the Z0101 index of the authority library. Note that the match text is actually constructed from the match codes defined in columns 7, 8 and 9;

col. 10-11

As col. 9

Example of the table:
## tab_z0101_text

Location of the table: tab directory of the BIB library

Purpose of the table: Text codes list and definitions for brief record (Z0101)

The tab_z0101_text table is used to define the text codes that are used by the tab_z0101 table to create brief records (Z0101).

Note that the relation between the lines (for various occurrences of the text code) is an OR type of relation.

The text code defined in column 1 of the table can be repeated to determine that if the field tag (column 2) defined for the first occurrence of the text code is not present, then the text code can be built from other fields of the record.

Structure of the table:

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Text codes that are used in the tab_z0101 table to define the structure of the new brief record (Z0101).</th>
</tr>
</thead>
</table>

Example:

If the tab_z0101 table has the following line:

```plaintext
## AUT 100## TITLE-MAIN
```

(The line defines that the TITLE-MAIN text code is the base of the brief record -Z0101- created for AUT headings from all bibliographic formats and created from the 100## field)

and if this table - tab_z0101_text - has the following lines:

```
TITLE-MAIN 100## t
TITLE-MAIN 240##
```
then the brief record for this type of headings will contain the
title from sub-field $t$ of the 100 field if present. If the sub-field is
not present, then the brief record will be created from the 240
field.
col. 2 Field from which the text code is created. If the field is not
present in the record, the system continues reading the table until
it finds a line with a matching field definition.
For example, if this table has the following lines:

<table>
<thead>
<tr>
<th>Title-MAIN</th>
<th>100## t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title-MAIN</td>
<td>240##</td>
</tr>
</tbody>
</table>

then brief records based on the TITLE-MAIN definition, will be
built from sub-field $t$ of the 100 field if present. If the sub-field
is not present, the system builds the brief record from the 240
field if present (and so on). Note that it is possible to add the
following line as a place-holder in case none of the specified
fields are present:

| Title-MAIN ^^^ a |

col. 3 Filter sub-field. For example, if this table has the following lines:

<table>
<thead>
<tr>
<th>Title-MAIN</th>
<th>100## t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title-MAIN</td>
<td>240##</td>
</tr>
</tbody>
</table>

brief records created based on the TITLE-MAIN text code will be
based on the 100 field only if sub-field $t$ is present in the field.
If the sub-field is not present, the brief will be created from the
240 field if present.
col. 4 Sub-fields to include in the brief record (Z0101). If this column is
left blank, the Z0101 record is built from all sub-fields present in
the field defined in column 2.
col. 5 New sub-field. If this column is left blank the original sub-fields
will remain otherwise they will be stripped.

Example of the table:
The library’s tab_z103 table defines the procedure(s) which are used to create links between records. The linking information is held in the document record, and the links themselves are held in the library's Z103 table. Retaining the linking information in the document record acts as a safeguard, so that the Z103 table can be re-built from scratch if necessary.

Links can be created based on:
- the ALEPH LKR field,
- MARC21 linking fields,
- UNIMARC linking fields,
- the 5xx reference fields of the AUT library.

The ALEPH structure of BIB, ADM and HOL records requires an LKR field in order to link between these records. The MARC21, UNIMARC and AUT links depend on various fields and sub-fields.

Links between BIB records can be displayed in the WWW and GUI SEARCH in the edit_doc_.<lng>9 display. When a link field displays in the full record, clicking on the field invokes display of the linked record.
In addition to the LKR field there are some special programs for building links. The possible programs:

<table>
<thead>
<tr>
<th>Name of Program</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update_z103_lkr</td>
<td>ALEPH link</td>
</tr>
</tbody>
</table>
| Update_z103_aut        | for thesaurus NT,BT,RT  
- based on field 550, sub-field w.  
Should be used in an AUT type library only. |
| Update_z103_dan        | for DANMARC  
- based on 001 (Faust number) and 015 which contains the UP link. This requires that 001 be assigned to the FST index. |
| Update_z103_ids        | based on MARC21 linking fields, for creation of UP/DN, PAR and ANA links. Although based on MARC21 links, there are departures from the MARC21 standard. |
| Update_z103_mab        | similar to ‘dan’ for MAB (German MARC)  
- based on tag 010 which contains the UP link and should be assigned to the FST index. |
| Update_z103_mab_ana    | For MAB format:  
- to be run for building analytical links between records. To support this, two fields, ANU (analytical up) and AND (analytical down) may be added to edit_doc_999.<lng>.
| Update_z103_uni        | based on UNIMARC linking fields, as developed for Italy |
| Update_z103_uni_2      | based on UNIMARC linking fields, as developed for Latvia |

Structure of the LKR field

<table>
<thead>
<tr>
<th>Sub-field</th>
<th>Content</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$a</td>
<td>UP</td>
<td>&quot;up&quot; link to another BIB type record. A record can have only one link of this type.</td>
</tr>
<tr>
<td></td>
<td>DOWN</td>
<td>&quot;down&quot; link to another BIB type record. Multiple links are possible. This link is rarely used, since the link should be made in the UP direction</td>
</tr>
<tr>
<td></td>
<td>PAR</td>
<td>parallel link from BIB record to BIB record</td>
</tr>
<tr>
<td></td>
<td>HOL</td>
<td>link from BIB record to HOL record</td>
</tr>
<tr>
<td></td>
<td>ADM</td>
<td>link from ADM record to BIB library</td>
</tr>
<tr>
<td></td>
<td>ITM</td>
<td>link to items - see below</td>
</tr>
<tr>
<td></td>
<td>ANA</td>
<td>analytic links - see below</td>
</tr>
<tr>
<td>$$b</td>
<td>&lt;sysno&gt;</td>
<td>system number of linked record</td>
</tr>
<tr>
<td>$$l</td>
<td>&lt;library&gt;</td>
<td>library (default is current library and is not mandatory)</td>
</tr>
<tr>
<td>$$r</td>
<td>&lt;MARC tag&gt;</td>
<td>Reason for link. From $$r the system will create a caption to display before $$n and $$m. The caption table is defined in tab05 located in the tab directory of the library</td>
</tr>
<tr>
<td>$$n</td>
<td>UP link text</td>
<td>Text explaining up link</td>
</tr>
<tr>
<td>$$m</td>
<td>DN link text</td>
<td>Text explaining down link</td>
</tr>
<tr>
<td>$$y</td>
<td>&lt;year&gt;</td>
<td></td>
</tr>
<tr>
<td>$$v</td>
<td>&lt;volume&gt;</td>
<td></td>
</tr>
<tr>
<td>Sub-field</td>
<td>Content</td>
<td>Explanation</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>$p</td>
<td>&lt;part&gt;</td>
<td></td>
</tr>
<tr>
<td>$i</td>
<td>&lt;issue&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Structure of LKR/ITM field**

<table>
<thead>
<tr>
<th>Sub-field</th>
<th>Content</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a</td>
<td>ITM</td>
<td></td>
</tr>
<tr>
<td>$b</td>
<td>&lt;sysno&gt;</td>
<td>system number of linked record – should be the system number of the ADM library, where the physical item is registered.</td>
</tr>
<tr>
<td>$n</td>
<td>UP link text</td>
<td>Text explaining up link (if necessary)</td>
</tr>
<tr>
<td>$m</td>
<td>DN link text</td>
<td>Text explaining down link (if necessary)</td>
</tr>
<tr>
<td>$y</td>
<td>&lt;year&gt;</td>
<td>as registered in ITEM record</td>
</tr>
<tr>
<td>$v</td>
<td>&lt;volume&gt;</td>
<td></td>
</tr>
<tr>
<td>$p</td>
<td>&lt;part&gt;</td>
<td></td>
</tr>
<tr>
<td>$i</td>
<td>&lt;issue&gt;</td>
<td></td>
</tr>
<tr>
<td>$k</td>
<td>Filter for pages</td>
<td></td>
</tr>
</tbody>
</table>

Links of type ITM build Z103 records that link the cataloging records:
- Z103 to closest "up" ($$y$$v$$p$$i is truncated until nearest match is found)
- Z103 "up" to root, and creation of "dn" from root to all ADM's
- Z103 "dn" to closest "up"

Note that this link type is used from a BIB record to an ADM record. An example of use is in the case of items being bound together. It does not create a link to the bibliographic record, only to the ADM. Therefore there will normally be no need to register sub-fields n and m for up and down notes.

**LKR/ANA field**
ANA links are between two BIB records and are intended to be used when linking an analytic record to the main series records.
The ANA link is made from the lower record to the higher one and the system automatically builds an UP/DOWN link between the two records as well as an ADM link to the items linked to the higher record.
If there are more than two tiers the UP/DOWN link will be from the lowest to the middle and from the middle to the highest. The ADM link will be to the ADM of the highest - i.e. the third tier.

**tab_z105**

**Location of the table:** tab directory of the library

**Purpose of the table:** Record update messages between libraries
In some instances, updating of a record will trigger update of a record in a different database. For example:
- update of a central BIB record should cause the same update in a local bib record;
update of an AUT records should cause an update of a z01 heading record. The system accomplishes this by sending a message (z105) as controlled by the tab_z105 table.

Structure of the table:

Col. 1  action that triggers a message (system set)

- UPDATE-DOC = Update of records (Z00)
- UPDATE-Z30 = Update of items (Z30)
- SEND-ILL-MSG = ILL transactions (SLNP-server)
- SUF-LOC = Update of patron information (for SUF)

Col. 2  message type

- 1=central to local message; local record is updated as a result of central record update
- 2=update of central HOL record from local HOL record
- 4=update of bibliographic z01 record because of authority database record update
- 5=update of central BIB record from local Z30 (item) record
- 6=ILL message
- 7 = Shared User File (SUF) - used to announce to the ADM library that an update has occurred in the USRxx lib
- 8 = Shared User File (SUF) - used to announce to the USRxx library that an update has occurred in the local ADM lib

Col. 3-12  libraries to which message is sent

Example of the table:

```
1 2 3 4 5 6 7 8 9 10 11 12
!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!
UPDATE-DOC 4 USM01
```

**tab_z105_filter**

Location of the table: tab directory of the library

Purpose of the table: Fields to retain when replicating z105 messages

Structure of the table:

Col. 1  Message type:

- 1=central to local message. Local record is updated as a result of central record update
- 2=update of central HOL record from local HOL record
- 4=update of bibliographic z01 record because of authority database record update
- 5=update of central BIB record from local Z30 (item) record

Col. 2  Action
• Y - Retains field
• N - Does not retain field

Col. 3 Tag code

tab_z121

Location of the table: tab directory of the BIB library

Purpose of the table: Definitions for the web service search engine

This table defines the URL and search query string for the search engine in the web service window.

Structure of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCITE</td>
<td><a href="http://search.excite.com/search.gw?search=$0100">http://search.excite.com/search.gw?search=$0100</a></td>
</tr>
<tr>
<td>GOOGLE</td>
<td><a href="http://www.google.com/search?q=$0100&amp;meta=lr%3D%26hl%3Den">http://www.google.com/search?q=$0100&amp;meta=lr%3D%26hl%3Den</a></td>
</tr>
<tr>
<td>NORTHERNLIGHT</td>
<td><a href="http://www.northernlight.com/nlquery.fcgi?cb=0&amp;qr=$0100">http://www.northernlight.com/nlquery.fcgi?cb=0&amp;qr=$0100</a></td>
</tr>
<tr>
<td>BRITANICA</td>
<td><a href="http://search.britanica.com/search?query=$0100">http://search.britanica.com/search?query=$0100</a></td>
</tr>
</tbody>
</table>

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCITE</td>
<td><a href="http://search.excite.com/search.gw?search=$0100">http://search.excite.com/search.gw?search=$0100</a></td>
<td></td>
</tr>
<tr>
<td>GOOGLE</td>
<td><a href="http://www.google.com/search?q=$0100&amp;meta=lr%3D%26hl%3Den">http://www.google.com/search?q=$0100&amp;meta=lr%3D%26hl%3Den</a></td>
<td></td>
</tr>
<tr>
<td>NORTHERNLIGHT</td>
<td><a href="http://www.northernlight.com/nlquery.fcgi?cb=0&amp;qr=$0100">http://www.northernlight.com/nlquery.fcgi?cb=0&amp;qr=$0100</a></td>
<td></td>
</tr>
<tr>
<td>BRITANICA</td>
<td><a href="http://search.britanica.com/search?query=$0100">http://search.britanica.com/search?query=$0100</a></td>
<td></td>
</tr>
</tbody>
</table>


tab_z30

Location of the table: tab directory of the ADM library

Purpose of the table: Default values for item records created using OCLC server and RLIN loader

This table is used for setting default values of item records that are system-created through the OCLC server and ue_03 (UTIL E/3) (RLIN loader). The table can have up to 200 lines.

Structure of the table:

| Col. 1 | Holding code in tag 049 of BIB record |
| Col. 2 | OWN Data. Bibliographic OWN Data used by the OCLC server. The data entered in this column should be added to the OWN field of records received from the OCLC server. Data from this column will be used if column 10 of tab_oclc is empty and the number 1 appears in tab_oclc column 5. |
| Col. 3 | Sub-library of the item |
| Col. 4 | Collection of the item |
Col. 5  Status of item being created
Col. 6  Material type code of the item
Col. 7  Oversize indicator
Col. 8  Call number of the item

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

| DEF | UEDUC GEN | 11 | BOOK | N |
| XZLA | UEDUC REF | 72 | BOOK | N |
| XZLD | UEDUC GEN | 11 | BOOK | N |
| ! VIDEO | VIDEO GEN | 32 | BOOK | N |
| ! XZLA | UMUSI GEN | 32 | BOOK | N |

**tab_z30_sort**

Location of the table: tab directory of the **BIB** and **ADM** libraries

Purpose of the table: Sort options for items

This table allows for the defining of sort options for items. Different sort routines can be used in different parts of the system.

Note that the instances that are related to WEB OPAC and SEARCH should be listed in the BIB library's tab_z30_sort. Other instances should be listed in the ADM library's tab_z30_sort.

A default item sort is set in the ITEMS, SERIALS and CIRCULATION clients' .ini files in the "ItemSortType" line. In addition, the list of items window includes a drop-down menu for choosing a sort type, in order to view the list in a different order than the default. The drop-down list is defined in pc_tab_exp_field.<lng> table under SERIAL-SORT-TYPE (for SERIALS client) and ITEM-SORT-TYPE (for ITEMS and CIRC clients).

The sort types are system-set and identified by a two-digit number. The sort and identification is separate for "issue" items and all other item types.

The issue/non-issue sort types are not completely parallel, although the following is a guideline:

- Issue 01 -> Non-issue 00 (enum)
- Issue 02 -> Non-issue 04 (HOL sublib + enum)
- Issue 03 -> Non-issue 05 (sublib + item sequence)

The choice might be whether or not to put the sublibrary as the first sort factor, where:

- library is first sort - ISSUE 02, 03, 04, 06 | NON-ISSUE 04, 05, 06
- library is not first sort - ISSUE 00, 01, 07, 08 | NON-ISSUE 00, 01, 02 or 03
Sort identifiers for ISS.. "issue" items

00 - by numbering, ignores location, reverses enum b/c
   • if chronological-i(year) is spaces and
   • enumeration-a(volume) is spaces then
     description+
     item-sequence.
   • if chronological-i(year) not spaces then
     chronological-i(year)+
     enumeration-a(volume)+
     enumeration-c(part)+
     enumeration-b(issue)+
     item-sequence

01 - by numbering, ignores location
   • if chronological-i(year) is spaces and
   • enumeration-a(volume) is spaces then
     description+
     item-sequence.
   • if chronological-i(year) not spaces then
     chronological-i(year)+
     enumeration-a(volume)+
     enumeration-b(issue)+
     enumeration-c(part)+
     item-sequence

02 - by location (using HOL records), then numbering
   • if chronological-i(year) is spaces and
   • enumeration-a(volume) is spaces then
     hol-doc-number+
     description+
     item-sequence.
   • if chronological-i(year) not spaces then
     hol-doc-number+
     chronological-i(year)+
     enumeration-a(volume)+
     enumeration-b(issue)+
     enumeration-c(part)+
     item-sequence

03 - by location (using sublibrary), then sequence
   • sub_library+
   • item-sequence

04 - by location (using sublibrary), then numbering
   • sub_library+
   • if chronological-i(year) is spaces and
• enumeration-a(volume) is spaces then
description+
item-sequence.
• if chronological-i(year) not spaces then
chronological-i(year)+
enumeration-a(volume)+
enumeration-b(issue)+
enumeration-c(part)+
item-sequence

06 - by sublibrary (preferred sublibrary by IP)
• sub-library name+
• collection code+
• chronological-i(year)+
• chronological-j(year)+
• chronological-k(year)+
• description+
• copy-id+
Note: ! * tab_attr_sub_library type 7 defines which sublibrary will be first in the
list.

07 - by 85x
• 85x-type+
• sub-library+
• collection+
• linking-number+
• if 85x-type is 4 or 5 then
  supp-index-o+
• if chronological not spaces and
  enumeration not spaces then
  chronological+ enumeration.
• if chronological is spaces then
description.

08 - by 85x [like 07, with addition of Z30-COPY-ID]
• 85x-type+
• sublibrary+
• collection+
• linking-number+
• copy-id+
• if 85x-type is 4 or 5 then
  supp-index-o+
• if enumeration not spaces then
  enumeration+
• if chronology not spaces then
  chronology+
• if enumeration not spaces or
chronology not spaces then
description.

The sort identifiers for non "issue" items are:

00 - by volume (enum or description) and location

- if enumeration-a(volume) not spaces then
  enumeration-a(volume)+
  enumeration-b+
  enumeration-c(part)+
- sub-library+
- collection.
- if enumeration-a(volume) is spaces then
  description+
  sub-library+
  collection

01 - by volume (enum + description) and location

- enumeration-a(volume)+
- enumeration-b+
- enumeration-c(part)+
- description+
- sub-library

02 - by volume (enum) and location

- if enumeration-a(volume) not spaces then
  enumeration-a(volume)+
  enumeration-b+
  enumeration-c(part)+
  sub-library.
- else
  description+
  sub-library

03 - by volume (description) and location and item status

- if description is blank then
  enumeration-a(volume)+
  chronological-i(year)+
  enumeration-b+
  enumeration-c+
  sublib+
  item status
- else
  description+
  sublib+
  item status

04 - by HOL and volume (enum or description) and location

- if enumeration-a(volume) not spaces then
hol-doc-number +
enumeration-a(volume)+
enumeration-b+
enumeration-c+
sub-library+
collection.
• else
hol-doc-number +
description+
sub-library+
collection

05 - by location (using sublibrary), then sequence
• sub_library+
• item-sequence;

06 - by sublibrary (preferred sublibrary by IP)
• sub-library name+
• collection code+
• description+
• copy-id
Note: * tab_attr_sub_library type 7 defines which sublibrary will be first in the list.

07 - by 85x
• 85x-type+
• sublibrary+
• collection+
• linking-number+
• if enumeration not spaces then
  enumeration.
• else
  description.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Function code:</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWW-A</td>
<td>WEB OPAC (should be present in BIB library)</td>
</tr>
<tr>
<td>WWW-R</td>
<td>WEB Course reading (should be present in BIB library)</td>
</tr>
<tr>
<td>WWW-X</td>
<td>&quot;circ status&quot; of X-Server</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Search client (should be present in BIB library)</td>
</tr>
<tr>
<td>SERIAL</td>
<td>Serials client</td>
</tr>
<tr>
<td>CIRC</td>
<td>Circulation client</td>
</tr>
<tr>
<td>ITEM-no.</td>
<td>Items client</td>
</tr>
<tr>
<td>ITEM-BIND</td>
<td>Items for binding</td>
</tr>
<tr>
<td>ACQ</td>
<td>Acquisitions client</td>
</tr>
<tr>
<td>TREE</td>
<td>Navigation window</td>
</tr>
</tbody>
</table>
• **BATCH** - Services
• **CIR-16** - Lost Item Report
• **HII** - Holdings Item Information (87x)
• **AHS** - Automatic Holdings Summary (86x)
• **86x** - for use by fix_doc_create_86x procedure;

Col. 2 Sorting order - ascending/descending for issue type items
Col. 3 Issue sort
Col. 4 Sorting order - ascending/descending for non-issue type items
Col. 5 Non-issue sort

Example of the table in the ADM library:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-!-!-!-!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOLDING-STMT</td>
<td>A</td>
<td>07</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>WWW-R</td>
<td>A</td>
<td>90</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CIRC-0</td>
<td>A</td>
<td>00</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CIRC-1</td>
<td>D</td>
<td>00</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CIRC-2</td>
<td>A</td>
<td>00</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CIRC-3</td>
<td>D</td>
<td>00</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>CIRC-4</td>
<td>D</td>
<td>00</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>SERIAL-0</td>
<td>A</td>
<td>03</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
<tr>
<td>SERIAL-1</td>
<td>A</td>
<td>01</td>
<td>D</td>
<td>01</td>
<td></td>
</tr>
</tbody>
</table>

Example of the table in the BIB library:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-!-!-!-!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!-!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WWW-A</td>
<td>A</td>
<td>01</td>
<td>A</td>
<td>05</td>
<td></td>
</tr>
<tr>
<td>SEARCH</td>
<td>D</td>
<td>00</td>
<td>D</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>

**tab_z311**

Location of the table: tab directory of the ADM library

Purpose of the table: Programs for building the Z311 table

The Z311 can now be used to store counters for z30-call-no/z30-call-no-2 and z30-inventory-number. This function is also available for items that are part of the HOL record. In order to define which kind of counter should be activated and which program can be used to create those counters tab_z311 has been implemented.

Structure of the table:

Col. 1 Item field name. Note that this is available for:

- call-no,
- z30-call-2-no,
- z30-inventory-number
Col. 2  Z311 Record type
• 0 - location
• 1 - inventory-number

Col. 3  Program name

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>1</td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!</td>
</tr>
<tr>
<td>z30-call-no</td>
<td>0</td>
<td>create_z30_location_i</td>
</tr>
<tr>
<td>z30-call-no-2</td>
<td>0</td>
<td>create_z30_location_i</td>
</tr>
<tr>
<td>z30-inventory-number</td>
<td>1</td>
<td>create_z30_inv_number</td>
</tr>
</tbody>
</table>

**tab00.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Field codes and names for indexes. There should be one such table for each language defined.

**ALEPH** allows various definitions of access paths (indexes) to document records, to suit the needs of each application. Indexes can be based on specific fields, a combination of fields, and individual words from specific document fields.

For example, the Titles table can contain titles, additional titles, uniform titles, etc. The Authors table may contain authors, co-authors, additional authors, translators, etc.

There are three types of indexes:

**Headings (ACC) table**
Examples of headings tables: authors, subjects, titles, publishers, etc. Headings are added to the table automatically from a document field during cataloging.

Each heading entry is unique has a list of documents associated with it (i.e. records in which the entry appears). A heading can be linked to an AUTHority database record, which will enable the user to display references linked to the heading.

**Index (IND) file**
Examples of Index tables: ISBN, ISSN, call number, Library of Congress, etc. Indexes serve as pointers to documents. Each Document field defined as Index opens an entry in the INDex table. The index entry is not necessarily unique, and the system number of the related documents sequences multiple occurrences of the same entry.

An index record is created automatically by the system from a document record field, during cataloging.

Note that the system automatically creates the SYS index for system numbers and the BAR index for barcodes. Therefore, although they do not need to be defined in the indexing tables (e.g. tab11_ind), nor created by an indexing service, they should be defined here in order to define the index name in column 11.
Words (W-) table:
Users select fields that serve as sources for entries in the Word tables. The system extracts each unique word from the specific fields of the document records, stores it in the Word table, and maintains pointers to the document in which it appears. A document will be counted as one in the count of occurrences, no matter how many times the word appears in it.

The default definition of *word* is: a character string from blank to blank, or from beginning of line to the first blank, or from last blank to the end of line.

Words are assigned to specific word *groups*. Thus, all words from the various types of title can be assigned to the words from titles group. Words from subjects can be assigned to a different words group. OPAC searches can apply to the general table or to any specific *group* table.

Users must define a tab00 line for a general Word list (WRD, W-001). Additional word groups can be added, using any code. The word group internal codes (W-nnn) must be unique, and should not jump large ranges of numbers.

Note that for Course Reading the CNO index code must not be renamed.

Structure of the table:
Col. 1 Not in use, should always be H
Col. 2 Heading code. Code of heading/word/index file. Can be up to 5 characters for headings and indexes, but only up to 3 characters for words
Col. 3 Index type. Use ACC for headings, IND for direct indexes and W-nnn for word indexes. For word indexes, each index must have a unique W- number, you must start counting from W-001, and you should not jump large ranges of numbers;
Col. 4 Not implemented
Col. 5 Filing procedure (specified in tab_filing)
Col. 6 Not implemented
Col. 7 Not implemented
Col. 8 Not implemented
Col. 9 Location weight (last 2 digits in the W section are for weighting as used in the Rank command of the Web OPAC)
Col. 10 Not in use, leave blank
Col. 11 Name of heading/word/index file
Example of the table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!-</td>
<td>---</td>
<td>-----</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>H SYS IND</td>
<td>21</td>
<td>00</td>
<td>00</td>
<td>System Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H BAR IND</td>
<td>21</td>
<td>00</td>
<td>00</td>
<td>Barcode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H 010 IND</td>
<td>22</td>
<td>00</td>
<td>00</td>
<td>LCCN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H ISBN IND</td>
<td>21</td>
<td>00</td>
<td>00</td>
<td>ISBN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H ISSN IND</td>
<td>21</td>
<td>00</td>
<td>00</td>
<td>ISSN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H TIT ACC</td>
<td>11</td>
<td>00</td>
<td>00</td>
<td>Titles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H TITC ACC</td>
<td>61</td>
<td>00</td>
<td>00</td>
<td>Titles Chinese</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H TITJ ACC</td>
<td>62</td>
<td>00</td>
<td>00</td>
<td>Titles Japanese</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H TITK ACC</td>
<td>63</td>
<td>00</td>
<td>00</td>
<td>Titles Korean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H TRT ACC</td>
<td>11</td>
<td>00</td>
<td>00</td>
<td>Titles/Related Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H AUT ACC</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>Authors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H WRD W-001</td>
<td>00</td>
<td>00</td>
<td>01</td>
<td>Words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H WTI W-002</td>
<td>00</td>
<td>00</td>
<td>02</td>
<td>W-titles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H WAU W-003</td>
<td>00</td>
<td>00</td>
<td>02</td>
<td>W-authors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H WSU W-004</td>
<td>00</td>
<td>00</td>
<td>03</td>
<td>W-subjects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H WUT W-005</td>
<td>00</td>
<td>00</td>
<td>02</td>
<td>W-Unif .Titles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab01.<lng>**

Location of the table: tab directory of the library

Purpose of the table: List of tags for records

The tab01 table defines the tags in the library's record. It should be repeated, with language extension, for every interface language set up for ALEPH. The table is used for:

- defining which heading list (Z01) is used for the update of a full record display formats 001 and 002 in OPAC
- record from authority control
- defining the filing procedure for sort keys based on a field (e.g. Z101 table, p-ret-21 (sort of retrieved BIB records))
- defining the non-filing indicator for the "non_filing" filing procedure
- setting the order of fields for the sort_docx procedure

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Col. 2 Internal codes or tags to be used for cataloging. Tags as defined in this table can be added, deleted, and changed as necessary. Each field code (tag) must be unique in the D table. Changes to field codes (tags) are reflected in any subsequent additions of records to the database, but do not affect existing records.

Col. 3 Filing procedure for sorting records as defined in (tab_filing). Used when building filing key for sort keys (Z101) and for the
batch services p-ret-21 (sorts retrieved docs) and p-manage-27 (update sort index). If changes are made to this column, p-manage-27 should be rerun.

If nothing is entered defaults to 99.

<table>
<thead>
<tr>
<th>Col.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Do not use (always 00)</td>
</tr>
<tr>
<td>5</td>
<td>Do not use (always 0000)</td>
</tr>
<tr>
<td>6</td>
<td>Non-filing indicator- 1/2</td>
</tr>
<tr>
<td></td>
<td>Indicates which indicator to look at for non-filing characters. Only used by p-ret-01 and p-ret-21 (col. 9 – (non filing indicator). tab11_acc is used for defining non-filing elements in indexed headings.)</td>
</tr>
<tr>
<td>7</td>
<td>The code of the headings list that is used for automatic correction of the document record field. If the heading which matches the document field is a &quot;see&quot; cross reference, and the heading is set to &quot;UPDATE Y&quot;, the field will be updated to the correct form.</td>
</tr>
<tr>
<td>8</td>
<td>Input/output code. Repeat definitions in col. 2.</td>
</tr>
<tr>
<td>9</td>
<td>Field name. Used in format 002 for display and printing. Always prefix the text with &quot;L&quot;.</td>
</tr>
</tbody>
</table>

Example of the table:

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>FMT</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>LDR</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>001</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>003</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>005</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>006</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>007</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>008</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>010</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>013</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>015</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>D</td>
<td>016</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>100</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>100</td>
<td>LME-Personal Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>110</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>110</td>
<td>LME-Corporate Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>111</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>111</td>
<td>LME-Meeting Name</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>130</td>
<td>11</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>1 TIT</td>
<td>130</td>
<td>LME-Uniform Heading</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>210</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>210</td>
<td>LAbbreviated Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>222</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>222</td>
<td>LKey Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>240</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>240</td>
<td>LJUniform Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>242</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>242</td>
<td>LTitle Trans.by CatAgency</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>243</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>243</td>
<td>LColl. Uniform Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>245</td>
<td>11</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>2 TIT</td>
<td>245</td>
<td>LTitle</td>
<td></td>
</tr>
</tbody>
</table>
tab02

Location of the table: tab directory of the library

Purpose of the table: Text to use for setting non-filing indicator value

This table defines initial text for the `fix_doc_non_filing_ind` routine. The routine is used for setting the value of the non-filing indicator for the relevant fields, by checking the initial text of a field.

It is activated in the Cataloging module via Fix Record, and the `tab_fix` routines (which are defined in the `tab_fix` table).

If, for example, the cataloger enters:

```
245   a The book of the dead
```

The `tab_fix` routine will add as the 2\textsuperscript{nd} indicator the number 4 (suppression of the word `the` plus one space).

```
245 4  a The book of the dead
```

The `tab_fix` routine is dependent on two elements:
\begin{itemize}
  \item definition of which indicator to use for stopwords in indexing (defined in `tab01.lng`)
  \item presence of a language code in fixed field 008 pos. 35-37
\end{itemize}

Structure of the table:
\begin{itemize}
  \item col. 1 alpha
  \item col. 2 language code
  \item col. 3 word to be ignored - Initial text to count for setting indicator value
\end{itemize}

Example of the table:

```
L eng the
L eng a
L eng an
L fre el
L fre il
L fre la
L spa lo
L spa un
L ger das
L ger Der
L ger die
L spa gli
```

tab04

Location of the table: tab directory of the library

Purpose of the table: Conversion of field codes between formats
Tab04 allows for converting one set of cataloging tags to another. It is used for tab_fix and for expand_doc_bib_tab04.

**tab_fix**
Different conversion routines can be defined and linked to the corresponding fix program in the tab_fix table. A conversion set can be used when importing records from databases with different cataloging systems (e.g. from a UNIMARC type of library to a MARC 21 library).

**expand_doc_bib_tab04**
This expand is primarily intended for the Z39_SERVER instance in tab_expand, in order to translate alphabetic tags to numeric values (e.g. LOC to 952), because the Z39 protocol does not recognize non-numeric tags.

Note that tags not defined in this table will be deleted from the record when activating the tab_fix or expand_doc routine.

**Structure of the table:**

- **Col. 1** Fix procedure.
- **Col. 2** Document field code of library from which record is copied. Use # for none or one character default field code of source library.
- **Col. 3** sub-field codes of the field of the library from which record is copied (up to 10, or blank for entire field)
- **Col. 4** document field code of local library to which record is copied
- **Col. 5** indication to strip sub-field codes when copying
  - Y=strip,
  - N=do not strip
- **Col. 6** sub-field codes to be used instead of original ones
- **Col. 7** text to be appended to contents of field

In example below the table is defined to translate UNIMARC into MARC21

**Example of the table:**

```
<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>FMT</td>
<td>FMT</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>LDR</td>
<td>LDR</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>001</td>
<td>001</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>005</td>
<td>005</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>010##</td>
<td>abdz</td>
<td>020##</td>
<td>N</td>
<td>abc</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>011##</td>
<td></td>
<td>022##</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>012##</td>
<td></td>
<td>024##</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>020##</td>
<td></td>
<td>015##</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>021##</td>
<td></td>
<td>017##</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>022##</td>
<td>bza</td>
<td>086##</td>
<td>N</td>
<td>az2</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>040##</td>
<td></td>
<td>030##</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>100##</td>
<td></td>
<td>008##</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
tab05.<lng>

Location of the table: tab directory of the BIB library

Purpose of the table: Captions for links between records

This table defines captions for the links between records, when the links are defined using the LKR field with subfield 'r'.

In the LKR field, the MARC tag that defines the reason for linking two records can be registered in subfield 'r'. Using the content of subfield 'r', tab05 defines the caption to display in the OPAC before $$n (up link) and $$m (down link).

This is a language sensitive table.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Col. 2</th>
<th>Col. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MARC tag – can be indicator sensitive</td>
<td>Caption for $$n (up link)</td>
<td>Caption for $$m (down link)</td>
</tr>
</tbody>
</table>

Note that the ‘tag’ NONE should be defined for linking fields where no $$r has been defined.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!!------!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!</td>
<td>Click to view related record</td>
<td>Main series:</td>
<td>Subseries of:</td>
</tr>
<tr>
<td></td>
<td>760</td>
<td></td>
<td>Has subseries:</td>
</tr>
<tr>
<td></td>
<td>762</td>
<td></td>
<td>Translation of:</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td>Other edition available:</td>
</tr>
<tr>
<td></td>
<td>775</td>
<td></td>
<td>Available in other form:</td>
</tr>
<tr>
<td></td>
<td>776</td>
<td></td>
<td>Issued with:</td>
</tr>
<tr>
<td></td>
<td>777</td>
<td></td>
<td>Continues</td>
</tr>
<tr>
<td></td>
<td>78000</td>
<td></td>
<td>Continues in part</td>
</tr>
<tr>
<td></td>
<td>78001</td>
<td></td>
<td>Supersedes</td>
</tr>
<tr>
<td></td>
<td>78002</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

tab06

Location of the table: tab directory of the BIB library

Purpose of the table: Defines non-high lighted words for full record display in the WEB OPAC.

Words listed in this table will not be highlighted in the full doc (edit_doc_999.<lng>) display in the WEB OPAC.

Structure of the table:
Col. 1  ALPHA
Col. 2  Word to be ignored

Example of table:

| ! 2 |
| !-!!!!!!!!! |
| L of |
| L the |
| L a |
| L an |

**tab07**

Location of the table: tab directory of the BIB library

Purpose of the table: Defines the links between Authority Headings

This table defines the hierarchical links between Authority Headings for building Z103 link records. It is valid for an AUT type of library only.

**NOTE:** tab07 is limited to 40 lines.

Building Z103 links between AUT records is **required** for multilingual AUT libraries. When cataloging in a multilingual AUT library it suffices to add 5XX references in one language only. Using this table, the system builds Z103 links, to "add" the 5XX in all its language manifestations. The additional 5XX fields are not actually present in the AUT record, but when the record is displayed, they are included (through the Z103 links).

Libraries can **opt** to use tab07 and Z103 links in order to build reciprocal 5XX references. According to table setup, if there is a 5XX in one record, the Z103 can create the reciprocal reference in the other record. In this way, if "broader term" references are entered, the "narrower term" references will be automatically generated.

For example:
LCGSH authority records include 550 fields for Broader terms only ($$wg). If the library wants to show references in the opposite direction (to Narrower terms), this can be done using this table.

Structure of the table:

| Col. 1  | field |
| Col. 2  | sub-field to strip or include when matching |
| Col. 3  | access list used for searching for the heading |
| Col. 4  | sub-field which contains link type |
| Col. 5  | link type code |
| Col. 6  | type of ALEPH link to build |
| Col. 7  | reciprocal link |
Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>550## axyz568</td>
<td>GEN</td>
<td>w a</td>
<td>ET</td>
<td>LT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>550## axyz568</td>
<td>GEN</td>
<td>w b</td>
<td>LT</td>
<td>ET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>551## axy568</td>
<td>GEN</td>
<td>w h</td>
<td>NT</td>
<td>BT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>551## axy568</td>
<td>GEN</td>
<td>w g</td>
<td>BT</td>
<td>NT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500## adqt568</td>
<td>GEN</td>
<td>w f</td>
<td>MC</td>
<td>MC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab11_acc**

Location of the table: tab directory of the library

Purpose of the table: Assignment of fields to headings indexes

The **tab11_acc** table is used to assign fields to headings indexes.

The table is limited to 1500 lines.

Structure of the table:

- **col. 1** Field tag. # (wildcard) can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (e.g., 245## for 2451, 2452, 24501).
- **col. 2** TYP or FMT code. This column can be used to index fields by the format of the record. The program first looks for a match on the TYP field (created by the expand_doc_type program) and then for a match on the FMT field. For example, if the column contains CF then the system will look through all TYP and FMT fields to see if there is a match on CF. If there is a match then the field will be indexed.
- **col. 3** Sub-field to filter. It is used together with column 4. For example if this column contains a 2 and column 4 contains **usm50**, then only records with **usm50** in sub-field 2 will have this field indexed.
- **col. 4** Sub-field contents filter. It is used together with column 3. For example:
  - if column 3 contains a 2 and this column contains **usm50**, then only records with **usm50** in sub-field 2 will have this field indexed.

If the contents is prefixed by a minus (-), the content is considered negative.

For example, if column 3 contains a 2 and this column contains **usm50**, then only records that **do not have** **usm50** in sub-field 2 will have this field indexed.

**NOTE!!** filter text must be entered in lower case (irrespective of
case in the record itself). Use * to indicate truncated text; use # to indicate single wild card for text match

col. 5  Code of the headings index file

col. 6  Sub-field(s) of the field for indexing. Blank indicates the entire field. Use minus (-) sign followed by sub-fields to define the sub-fields to be stripped or list the sub-fields to be indexed.

col. 7  Group. Values are: A, B or C. This column may be used to define a particular group of fields that will be used to create a new headings index. This option cannot be used for an index that already exists. Note that this option is available only when the program is run from the Unix command line. It is not available from the Headings index batch service.

The following is an example of the way in which the program should be run for fields that belong to group B:

    from ../aleph/proc
csh -f p_manage_02
USM01,0,0000000000,999999999,B,1,N,00,

col. 8  Non-filing indicator for non_filing procedure as defined in the tab_filing table.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>050##</td>
<td>LCC</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>090##</td>
<td>LCC</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>060##</td>
<td>NLM</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>082##</td>
<td>DDC</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>086##</td>
<td>SUD</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>086##</td>
<td>SUD</td>
<td>z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100##</td>
<td>AUT</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100##</td>
<td>9</td>
<td>chi</td>
<td>AUTC</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100##</td>
<td>9</td>
<td>jpn</td>
<td>AUTJ</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>100##</td>
<td>9</td>
<td>kor</td>
<td>AUTK</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU100</td>
<td></td>
<td>AWT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AT100</td>
<td></td>
<td>AWT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>110##</td>
<td></td>
<td>AUT</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>110##</td>
<td>9</td>
<td>chi</td>
<td>AUTC</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>110##</td>
<td>9</td>
<td>jpn</td>
<td>AUTJ</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>110##</td>
<td>9</td>
<td>kor</td>
<td>AUTK</td>
<td>-e468</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab11_aut**

Location of the table: tab directory of the library

Purpose of the table: Link back from the AUT record
The tab11_aut table is used to defines the headings files that the system uses to create hypertext links to FIND and BROWSE from the authority record. This allows the user to navigate the bibliographic database using the authority record fields.

Structure of the table:

- **col. 1**: Authority record field. Note that if an authority term is linked to more than one headings file (personal names, 100, can be used for both authors and subjects), several lines can be created for the field. The system runs the search from the current headings file. # can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (e.g., 100# for 1000, 1001, 1003).

- **col 2**: Not in use
- **col 3**: Not in use
- **col 4**: Bibliographic headings file on which Browse/Find is activated
- **col 5**: Sub-fields to include/strip from authority record field for Search. The system looks for the exact text of the field in the headings file. Blank indicates entire field. Use the minus (-) sign followed by sub-fields to define sub-fields to be stripped or list the sub-fields to be indexed.

- **col 6**: Not in use
- **col 7**: Not in use

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>100##</td>
<td>AUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100##</td>
<td>AWT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100##</td>
<td>SUB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100##</td>
<td>SUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100##</td>
<td>TIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110##</td>
<td>AUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110##</td>
<td>AWT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110##</td>
<td>SUB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150##</td>
<td>SUB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150##</td>
<td>SUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150##</td>
<td>SUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150##</td>
<td>MLT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151##</td>
<td>SUB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151##</td>
<td>SUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151##</td>
<td>SUM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151##</td>
<td>MLT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400##</td>
<td>AUT</td>
<td>-wi5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**tab11_ind**

Location of the table: tab directory of the library

Purpose of the table: Assignment of fields to direct indexes

The tab11 table is used to assign fields to Direct indexes. Direct indexes enable the user to retrieve a specific record. A direct index is suited to unique or almost unique identifiers of the record, and as such serves as a quick access to a record.

Direct indexes are also used by record loader and union view tables, in order to find possible matching records (candidates).

The table is limited to 500 lines.

Structure of the table

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Field tag. # (wildcard) can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (e.g., 245## for 2451, 2452, 24501).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Not in use.</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Sub-field to filter. It is used together with column 4. For example if this column contains a 2 and column 4 contains usm50, then only records with usm50 in sub-field 2 will have this field indexed.</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Sub-field contents filter. It is used together with column 3. For example: if column 3 contains a 2 and this column contains usm50, then only records with usm50 in sub-field 2 will have this field indexed. If the contents is prefixed by a minus (-), the content is considered negative. For example, if column 3 contains a 2 and this column contains -usm50, then only records that do not have usm50 in sub-field 2 will have this field indexed.</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Code of the headings index file</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Sub-field(s) of the field for indexing. Blank indicates the entire field. Use minus (-) sign followed by sub-fields to define the sub-fields to be stripped or list the sub-fields to be indexed.</td>
</tr>
<tr>
<td>Col. 7</td>
<td>Not in use</td>
</tr>
<tr>
<td>Col. 8</td>
<td>Non-filing indicator for non_filing procedure as defined in the tab_filing table.</td>
</tr>
</tbody>
</table>

NOTE!! filter text must be entered in lower case (irrespective of case in the record itself). Use * to indicate truncated text; use # to indicate single wild card for text match

Example of the table:
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>010##</td>
<td>010</td>
<td>-bz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>050##</td>
<td>TST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010##</td>
<td>STIDN</td>
<td>-bz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>020</td>
<td>ISBN</td>
<td>az</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>020</td>
<td>STIDN</td>
<td>az</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>022</td>
<td>ISSN</td>
<td>ayz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>022</td>
<td>STIDN</td>
<td>ayz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>028##</td>
<td>028</td>
<td>-b6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>028##</td>
<td>STIDN</td>
<td>-b6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>024##</td>
<td>024</td>
<td>-26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab11_word**

**Location of the table: tab directory of the library**

**Purpose of the table: Assignment of fields to word indexes**

The tab11_word table is used to assign fields to word indexes. Note that for the "general words file" the library should use W-001 WRD, and there should not be large gaps in the numbers of the specific word groups.

The table is limited to 10000 lines.

**Structure of the table:**

- **Col. 1** Field tag. Document record field code. # can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (e.g., 245## for 2451, 2452, 24501)
- **Col. 2** TYP or FMT code. This column can be used to index fields by the format of the record. The program first looks for a match on the TYP field (created by the expand_doc_type program) and then for a match on the FMT field. For example, if the column contains CF then the system will look through all TYP and FMT fields to see if there is a match on CF. If there is a match then the field will be indexed.
- **Col. 3** Sub-field filter, used together with column 4. For example, if this column contains a '2' and column 4 contains 'usm50', then only records with usm50 in sub-field 2 will have this field indexed.
- **Col. 4** Sub-field contents filter. It is used together with column 3. For example, if column 3 contains a '2' and this column contains 'usm50', then only fields with usm50 in sub-field 2 will be indexed. Note the following:
  - the filter text must be entered in lower case (irrespective of case in the record itself)
• the filter text must not include special characters (only 7 bit Latin)
• the filter text does not undergo filing (only unicode_case) and for this reason it should contain the exact string
• use * to indicate truncated text; use # to indicate a single wild card for text match;

Col. 5 Sub-field(s) of the field for indexing:
• sub-field(s) of the field (blank indicates entire field). OR
• minus (-) sign followed by sub-fields to be stripped OR
• F followed by fixed field position (counted from base 00, taking sub-field code into account), followed by hyphen (-) and the no. of characters.

Col. 6 Word building procedure - see tab_word_breaking for more details

Col. 7-8 Not in use

Col. 9-18 Word index code. Up to 10 codes can be assigned

Note the following:
When word proximity (and adjacency) is taken into account, the text string being indexed is used. If the text string is a selection of subfields, this will cause some problems with adjacency.
For example, if you are indexing field 246, which might have subfields a, b, c, d etc., but you have defined indexing as:

```
11 W 246## abnp 03 WTI
```

This leaves out subfields c,d,e,f,g,h,i; and the last word of $$b will be considered adjacent to the first word of $$n.

There might be instances where this is desirable. But, if you do not want this to happen, you should set indexing to separate strings, e.g.

```
11 W 246## ab 03 WTI
11 W 246## np 03 WTI
```

Example of the table:
**tab14**

**Location of the table: tab directory of the ADM library**

**Purpose of the table: Item reshelving time**

Tab14 allows the library to define the time it takes for a returned item to get back on the shelf. The definition can be on the level of the branch library, collection and item status. When an item is returned, the due date column will display reshelving for the time defined in this table.

**Structure of the table:**
- col. 1  sub-library (use #### to signify all sub-libraries)
- col. 2  collection (use #### to signify all collections)
- col. 3  item status (use # to denote all items)
- col. 4  item process status (use ## to denote all items)
- col. 5  reshelving time (HHMM)

**Example of the table:**

```
! 1  2  3  4  5
!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!
!!!!!!!!!!
008  F07-04  01 A  WRD WYR
008  F35-03  01 A  WRD WLN
SBL   01    WSL
SYS   90    SYS
LOC## -o  03    WRD WLC
041## abdefg 41 A  WRD WLN
043##               01    WRD WGA
100## -6   01    WRD WAU
110## -6   01    WRD WAU
111## -6   01    WRD WAU WTI
130## -6   03    WRD WTI WUT
210## a    03    WRD WTI
222## a    03    WRD WTI
240## -6   03    WRD WTI WUT
242## abnp  03    WRD
243## -6   03    WRD WTI
245## c     03    WAU
245## -c6  03    WTI
```
**tab15.<lng>**

Location of the table: tab directory of the ADM library

Purpose of the table: Defines the item statuses and various parameters connected to these statuses.

Structure of the table:

Col. 1  Group ID for group of definitions as defined in the tab_sub_library.<lng>. Note that there is a limitation of 1000 lines for the drop-down list. ### (wildcard) cannot be used in this column.

col. 2  item status. ## is used to denote all item statuses

col. 3  process status - ## is used to denote all process statuses

col. 4  L

col. 5  description

col. 6  can be loaned Y/N

col. 7  can be renewed Y/N

col. 8  Availability check for hold request:
  
  - **Y** - item can be requested. In addition, item availability is checked when hold request is placed. If a copy is available, and patron does not have permission for requesting available item, message displays.
  
  - **N** – item cannot be requested
  
  - **C** - item availability is not checked when hold request is placed, because hold requests are always allowed (e.g. closed stack).
  
  - **T** - item availability is not checked when hold request is placed, because hold requests are always allowed (e.g. short-term loan, lost, on order, etc.). However, for p-cir-12 (print request slips) and ue_06 (online printing of request slips), item is not considered "requestable" and does not print.

In general, note that apart from the availability check in tab15.<lng>, the checks that are made when a hold request is made is controlled by tab_hold_request.

col. 9  Availability check for photo requests;
  
  - **Y** – Photocopy request can be placed
  
  - **N** - photocopy request cannot be placed for the item.
  
  - **T** - item availability is not checked when hold request is placed, because photocopy requests are always allowed
(e.g. short-term loan, lost, on order, etc.). However, for p-cir-22 (print photocopy request slips) and ue_06 (online printing of request slips), item is not considered "requestable" and does not print.

In general, note that apart from the availability check in tab15.<lng>, the checks that are made when a photocopy request is made is controlled by tab_photo_request.

col. 10 Display item in Web OPAC Y/N. Even if set to N – will always display in GUI SEARCH.
col. 11 If this item is requested in the OPAC, the patron will be given this specific item, not another copy (Y/N).
col. 12 limit hold requests to the number of available copies - there can only be a single request per item (Y/N).
col. 13 allow recall of item (Y/N). Note that tab16 cols.19 and 20 define the no. of days for recall notice and the minimum guaranteed loan period. Recall notices are generated by the Recall Notices service (p-cir-13)
col. 14 allow rush recall of item (Y/N).
col. 15 Reloaning limit: The value denotes the number of hours during which a patron is not allowed to reloan the same copy of an item. The count is from the time he returned it. The values "00" or " " (blanks) should be used for unrestricted reloaning.

With regard to cols. 13 and 14 – recall/rush recall: note the variable in www_server.conf setenv www_z37_recall_type, which defines the default recall type when placing a hold request in the WEB OPAC.
(The recall type is 01, 02 or 03, where 01 is regular, 02 is rush and 03 is no recall.)

Example of the table:
Location of the table: tab directory of the ADM library

Purpose of the table: Due dates, fines & limits

Special note: When a change is made to tab16 in order for the changes to take immediate effect - the due dates table, (Z301) should be dropped and recreated (UTIL A/17/1) Note, however, that if the table is not dropped and recreated in this way, the changes automatically take effect the next day.

This table allows for various definitions per patron/item status – for e.g.

- computation of due dates
- grace period
- fine rate
- maximum number of items that can be loaned to a patron
- maximum number of hold requests
- fine method, etc.

The computation of the due dates for each combination of copy/patron status is performed by the system daily, according to the rules defined in tab16. The result of the computation displays in the Due Dates option in the Circulation GUI. The operator can change the dates if necessary.

Fine rates. The library can opt to set fines for selected statuses only (e.g. professors do not have a fee rate, students with patron status nn have a higher fee rate than students with patron status xx.)
Note the switch in tab100, (ZERO-FINE-HANDLING) which defines whether late returns should be registered in the Cash Management record, even if there is no fine for the late return.

**Note:**
The groups in tab16 (16A, 16B, etc.) serve only to reduce the size of the table when there is a library policy which is the same for more than one sub-library. The group itself has no functionality as a group. All checks are made on the sub-library level. All checks of limits are for the items of the same sub-library as the item in hand. For example, If there is a 99 line for "total limit", then this is the **total limit for each of the sub-libraries individually**, and not for the sub-libraries as a group.

The maximum number of lines per sublibrary or group is 500.

**Item status 99**
Defines limits for the total number of loans/holds. There is no item status check.
You can use ## in col.4 (patron status) to denote all patron statuses or enter a specific patron status in col.4 (patron status) if you want to define the global loan/hold limit per patron status.
The total check is for every sublibrary that relates to the group ID. It is possible to define a total check for the entire ADM library.
Define a separate line with a different group ID for the ADM library. Enter this group ID in column 7 in tab_sub_library.<lng> in the line that defines the ADM library.
This must be the LAST line in every group ID.

Hours are rounded to a single hour, days are rounded to a single day. The default is 2.

If no parameters are defined for cols. 15-18, then the regular date/hour parameters are used (cols. 5-6, 8-9)

**Structure of the table:**

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Group ID for group of definitions as defined in tab_sub_library.&lt;lng&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Item status. 2 numeric digits. (## denotes any/all status(es).</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Process status (## for all)</td>
</tr>
</tbody>
</table>

**Note:** The item process status is not taken into consideration when creating due dates and due hours (cols. 6, 9, 16 and 18) but is taken into consideration for all other columns. Due dates (Z301) will only be built from tab16 lines which have "##" in the item process status column.

<table>
<thead>
<tr>
<th>Col. 4</th>
<th>Patron status. ## denotes all patron statuses. Enter a specific patron status if you want to define the global loan/hold limit per patron status.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 5</td>
<td>Date operator for interpretation of the next field:</td>
</tr>
<tr>
<td></td>
<td>• A The date in col. 6 is an absolute date to be assigned as is.</td>
</tr>
<tr>
<td></td>
<td>• + The date in col. 6 is a relative time to be added to the current date.</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Date parameter - DDDDDDDD or YYYYMMDD</td>
</tr>
</tbody>
</table>
This date is treated according to the operator in col. 5:

If + in col. 5 then no. of days to add. Only the last three digits are relevant – e.g. if you enter 1950 days, the system will compute 950 days. Therefore, the maximum number of days that can be defined is 999.

If A in col. 5 then actual date – e.g. 20101231.

**col. 7** Grace days – i.e. number of days within which late return is "forgiven". Note however, that if an item is returned after the grace period, the fine rate will be for the total number of overdue days/hours, taking the grace period into account.

**col. 8** Hour operator for interpretation of the next field:
- A - The hour in col. 9 is the absolute hour to be assigned as is.
- + - The hour in col. 9 is the relative hour to be added to the current hour. This is especially useful for short term loans.

**NOTE:** If there is a + in this column the dates field (col. 6) should be zeroes.

**col. 9** HHMM
Hour and minutes for due date. This date is treated according to the flag in col. 8

**col. 10** Grace time - i.e. the hours and minutes within which a late return is “forgiven”.

**col. 11** Fine multiple, i.e. the factor by which the number of days overdue is multiplied to compute the fee owing for items returned late or overdue (3 digits + 2 decimal).

**NOTE:** the sum refers to a day. In order for it to refer to an hour, multiply the amount for an hour by 24 and register the resulting amount in this column.

Note that in /alephe/aleph_start it is possible to set the multiplication factor and rounding (Y/N).

Example of fine definition in /alephe/aleph_start (UTIL Y/14):

```
Setenv fine_factor_fix 000
Setenv fine_rounding N
```

**col. 12** Maximum number of items for patron/item matrix – 3 digits
Remember that item status 99 defines total number of loans for defined patron status (no item status check). It is not possible to set to unlimited number of loans. If you have a patron status that you want to set up to allow an unlimited number of loans, set “check loan” (col.7) in tab31 to N (z305_loan_check=N). (for e.g. for institutional patrons, like ILL departments.)

**col. 13** Maximum number of hold request – 3 digits.
The maximum number of hold requests a patron is allowed to have of a specific item status or of all item statuses when relating to the 99 line
col. 14  fine method

- 0 - no fines charged
- A - open minutes
- 1 - open hours
- 2  open days
- B - total minutes
- 3 - total hours
- 4 - total days
- 5 - overlapping block date
- 6 - cumulative block date
- C - include recall - behaves like type 1 (open-hours)
- D - include recall - behaves like type 2 (open-days)
- E - include recall - behaves like type 3 (total-hours)
- 7 - include recall - behaves like type 4 (total days)
- R - fixed 1st day - behaves like type 2 (open-days)
- F - total days by intervals (using tab_late_return)
- 8 - overlapping block date +fine calculated according to total days
- 9 - cumulative block date +fine calculated according to total days
- Y - special for Denmark
- Z - special for Denmark

Note the following:

**Fine method 5 - overlapping block date**

A patron who returns one item late is blocked from borrowing items or renewing loans (this depends on the setup of tab_check_circ) for a period equal to the number of days the item was overdue.

A patron who returns two or more items late is blocked from borrowing items or renewing loans for a period equal to the highest number of days an item was overdue.

For example, a patron who returns two items late, one 5 days overdue and the other 10 days overdue is blocked from borrowing items for 10 days.

**Fine method 6 - cumulative block date**

A patron who returns one item late is blocked from borrowing items or renewing loans (this depends on the setup of tab_check_circ) for a period equal to the number of days the item was overdue.

A patron who returns two or more items late is blocked from borrowing items or renewing loans for a period equal to the sum of the number of days each item was overdue.

For example, a patron who returns two items late, one 5 days overdue and the other 10 days overdue is blocked from borrowing items for 15 days.
**Fine method R - First day fixed, then open days**

The fine for the first overdue day is a fixed sum of 2.00 (hardcoded). No other fines are added for the first day. From the second day onward the calculation is like fine method 2 (open days).

**NOTE:** hours are rounded to a single hour; days are rounded to a single day. The default set by the system is fine method 2.

col. 15  Date operator for loan of items with holds (+ or A as in col. 5)
col. 16  Date parameter for loan of items with holds (+ or A as in col. 6)
col. 17  Hour operator for loan of items with holds (+ or A)
col. 18  Hour parameter for loan of items with holds (+ or A)

**NOTE:** if no parameters are defined for cols. 15-18, then the regular date/hour parameters are used (cols. 5-6, 8-9)

col. 19  Number of days from recall notice to new due date for recalled item
col. 20  Number of days for minimum guaranteed loan period (when recalled)

col. 21  Maximum no of renewals. Values: 00-09 or blanks. Renewals can be limited up to 9; leave blank to define that renewals are unlimited.

To limit renewals by a time period rather than by a number, leave this column blank, and define the time period in column 22.

col. 22  Maximum renewal period
D days/W weeks/M months
nnn - number of units

example:
D007 - renew up to a maximum of 7 days
W004 - renew up to 4 weeks

Col. 23. Maximum fine (up to 2 decimals); the field should be left-aligned
Col. 24 Minimum fine. Fines below this amount will be treated as zero amount fines. The field should be left-aligned
Col. 25 Fixed Fine Addition. Fixed amount to be added to any fine. The field should be left-aligned.

Example of the table:
### tab17

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Library open hours

Tab17 defines the days and times that the library is open. It serves 2 purposes:
- sets due date to next open date
- skips "closed" days when computing fines, depending on tab16 col. 14 (fine method) setup

**A library does not have to utilize tab17 for fines.** If it does, fines will be calculated based on the open hours of the library and not on a 24 hour day (as per definition of col. 13 in tab16).

Note that tab16 column 11 still defines the daily fine based on a 24 hour day and tab17 should be defined with this in mind.

Tab17 is also used to calculate due dates. Due dates will not be assigned days that are listed as C (closed) in tab17.

Tab17 is read from the top down, this means that exceptions should appear first, followed by the schedule of a typical week. The valid values for 'Day of Week' column (col. 3) are either numbers to specify the exact day, or spaces that stand for any day.
Structure of the table:

Col. 1 Group ID for group of definitions as defined in tab_sub_library.<lng> (col. 8). A Z30_DEPOSITORY_ID code can be used in this column to set a remote storage’s opening hours disregarding the fact that a Z30_DEPOSITORY_ID code is not a sublibrary code.

col. 2 date YYYYMMDD. Use # as wild card

col. 3 day of week (00 Sun - 06 Sat)

col. 4 O(pen)/C(losed)

col. 5 open hour

col. 6 closing hour

Example of the table:

| 17A | ####1230 | C      |
| 17A | ####1231 | C      |
| 17A | ####0101 | C      |
| 17A | ####0102 | C      |
| 17A | ######## | 01 O 0900 2000 |
| 17A | ######## | 02 O 0900 2000 |
| 17A | ######## | 03 O 0900 2000 |
| 17A | ######## | 04 O 0900 2000 |
| 17A | ######## | 05 O 0900 2000 |
| 17A | ######## | 06 C    |
| 17A | ######## | 00 C    |

... 

tab18.<lng>

Location of the table: tab directory of the ADM library

Purpose of the table: Cash transaction types

A table which defines cash control options. It is branch library/patron status sensitive. Note that for some lines the value in col. 4 is Y or N (meaning that the system should or should not register a cash transaction), or a ratio. The actual amount is set in other tables. In the same case for some lines the value in col.5 is not relevant because the actual amount is set in other tables. In these cases col.5 should be left empty in order not to create confusion, except for specific cases mentioned below.

- **0003,0014** - late return fee, dependent on tab16
- **040,0041,0042** - lost material, dependent on tab34 If tab34 is not present or if there is no value in tab34, the default amount is taken from tab18.<lng>, col.5. In any case the GUI operator can change the value for 0041 (material replacement cost) when registering an item as lost. The amounts for 0040 (handling cost) and for 0042 (notice cost) are not influenced by a manual change of the value for 0041 by the GUI operator
• 0015 - ILL material arrival, dependent on tab_ill_charge
• 0050-0055 - ratio, actual amount or additional daily amount charged for late return of recalled/rush recalled loan. The parameter OVERDUE-RECALL-RATIO in tab100 defines whether the number is a ratio, a set amount or a daily amount added to the fine as calculated in tab16.

**General note:** Note that the possibility to calculate a fine will depend on the existence of the loan history (Z36H), which will be used to locate the exact cash transactions related to the return.

List of transactions and their description:

<table>
<thead>
<tr>
<th>Transaction No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>General</td>
</tr>
<tr>
<td>0001</td>
<td>Photo request</td>
</tr>
<tr>
<td>002</td>
<td>Hold request</td>
</tr>
<tr>
<td>0003</td>
<td>Late return - fine is defined in tab16</td>
</tr>
<tr>
<td>0004</td>
<td>Loan</td>
</tr>
<tr>
<td>0005</td>
<td>Renewal</td>
</tr>
<tr>
<td>0006</td>
<td>Photocopy request processing</td>
</tr>
<tr>
<td>A3-6</td>
<td>Photocopy processing (A3 page size) – example</td>
</tr>
<tr>
<td>A4-6</td>
<td>Photocopy processing (A4 page size) – example</td>
</tr>
<tr>
<td>B2-6</td>
<td>Photocopy processing (B2 page size) – example</td>
</tr>
<tr>
<td>0007</td>
<td>Photocopy request not filled</td>
</tr>
<tr>
<td>0008</td>
<td>Photocopy request home delivery</td>
</tr>
<tr>
<td>0009</td>
<td>Photocopy request pickup</td>
</tr>
<tr>
<td>0010</td>
<td>Claim return</td>
</tr>
<tr>
<td>0011</td>
<td>ILL request</td>
</tr>
<tr>
<td>0013</td>
<td>Routing list</td>
</tr>
<tr>
<td>0014</td>
<td>Late return of routed issue</td>
</tr>
<tr>
<td>0015</td>
<td>ILL material arrival</td>
</tr>
<tr>
<td>0016</td>
<td>Library charge for incoming ILL request</td>
</tr>
<tr>
<td>0017</td>
<td>Printing patron's label card (bor-label-card)</td>
</tr>
<tr>
<td>0021</td>
<td>Local patron registration</td>
</tr>
<tr>
<td>0022</td>
<td>Local patron renewal</td>
</tr>
<tr>
<td>0023</td>
<td>New Patron</td>
</tr>
<tr>
<td>Charges 24-28 are activated when the hold is filled using &quot;Print letter - hold request filled&quot; from the Requests option in the Main Menu. See also charges 1024, 1026, 1027, 1028</td>
<td></td>
</tr>
<tr>
<td>0024</td>
<td>Hold request filled - tab37 process option type 01 (hold shelf)</td>
</tr>
<tr>
<td>0025</td>
<td>Hold request not filled</td>
</tr>
<tr>
<td>0026</td>
<td>Hold request filled - tab37 process option type 02 (home delivery)</td>
</tr>
<tr>
<td>0027</td>
<td>Hold request filled - tab37 process option type 03 (mailbox)</td>
</tr>
<tr>
<td>0028</td>
<td>Hold request filled - tab37 process option type 04</td>
</tr>
<tr>
<td>Transaction No.</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>(reading room)</td>
<td>(reading room)</td>
</tr>
<tr>
<td>0040</td>
<td>Lost material – Handling amount defined in tab34, col. 7</td>
</tr>
<tr>
<td>0041</td>
<td>Lost material – Replacement amount defined in tab34, col. 8</td>
</tr>
<tr>
<td>0042</td>
<td>Lost material – Notice - amount defined in tab34, col. 6</td>
</tr>
<tr>
<td>0050</td>
<td>Recall late return fine</td>
</tr>
<tr>
<td></td>
<td>If TAB-100-OVERDUE-RECALL-RATIO is &quot;Y&quot;, then the fine amount in tab16 is multiplied by this number.</td>
</tr>
<tr>
<td></td>
<td>If TAB-100-OVERDUE-RECALL-RATIO is &quot;N&quot;, then this number is the actual fine amount.</td>
</tr>
<tr>
<td></td>
<td>If TAB-100-OVERDUE-RECALL-RATIO is &quot;B&quot;, then this number is added to the fine amount in tab16.</td>
</tr>
<tr>
<td>0051</td>
<td>Rush Recall late return fine</td>
</tr>
<tr>
<td></td>
<td>See tab100 switch as above</td>
</tr>
<tr>
<td>0052</td>
<td>Recall fine limit ratio</td>
</tr>
<tr>
<td></td>
<td>See tab100 switch as above</td>
</tr>
<tr>
<td>0053</td>
<td>Rush Recall fine limit ratio</td>
</tr>
<tr>
<td></td>
<td>See tab100 switch as above</td>
</tr>
<tr>
<td>0054</td>
<td>Recall lost letter (ratio, actual amount or additional daily amount, see the explanation for 0050) added to the notice charge defined (Col.7) in tab34.</td>
</tr>
<tr>
<td>0055</td>
<td>Rush Recall lost letter (ratio, actual amount or additional see the explanation of 0050) added to the notice charge defined (Col. 8) in tab34</td>
</tr>
<tr>
<td>0080</td>
<td>1st warning – Overdue</td>
</tr>
<tr>
<td>0081</td>
<td>2nd warning – Overdue</td>
</tr>
<tr>
<td>0082</td>
<td>3rd warning – Overdue</td>
</tr>
<tr>
<td>0083</td>
<td>4th warning – Overdue</td>
</tr>
<tr>
<td>0090</td>
<td>Overdue summary</td>
</tr>
<tr>
<td>1000</td>
<td>Payment</td>
</tr>
</tbody>
</table>

Charges 1024, 1026, 1027, 1028 are activated when the hold is filled when the item is returned and the "Letter" button on list of holds is activated. See also charges 0024, 0026, 0027, 0028

<p>| 1024 | Hold request filled - tab37 process option type 01 (hold shelf) |
| 1025 | Hold request not filled |
| 1026 | Hold request filled - tab37 process option type 02 (home delivery) |
| 1027 | Hold request filled - tab37 process option type 03 (mailbox) |
| 1028 | Hold request filled - tab37 process option type 04 (reading room) |</p>
<table>
<thead>
<tr>
<th>Transaction No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9000</td>
<td>Deposit</td>
</tr>
<tr>
<td>9001</td>
<td>Return deposit</td>
</tr>
<tr>
<td>9996</td>
<td>Local charge for disruption</td>
</tr>
<tr>
<td>9997</td>
<td>Damaged material</td>
</tr>
<tr>
<td>9998</td>
<td>Donation</td>
</tr>
<tr>
<td>9999</td>
<td>Payment</td>
</tr>
</tbody>
</table>

Note that system activated transactions are assigned numbers from 0000 to 8999 while manually activated transactions are assigned numbers from 9000-9999.

Structure of the table:
- Col.1  Transaction number – note that these numbers are system defined and cannot be modified.
- Col.2  Sub-library (use ##### to signify all sub-libraries)
- Col.3  Patron status (use ## to signify all statuses)
- Col.4  Y/N –
  - Y – activate cash control
  - N - do not activate. In this case cash transactions of this type will be registered with the value 0.00 without regard to tab18.<lng> col.5 or the value set in the relevant table.
- Col.5  rate
- Col.6  VAT rate: left-aligned. This number is the rate of the VAT that will be added to the sum that is set in column 5. For example, if a transaction has the number 10.00 in column 5 and 15.00 in column 8, the net sum will be 10.00, the VAT sum will be 1.50 (15 percent of 10) and the sum of the transaction will be 11.50. If the column is left empty, no VAT is assumed and the net value of the transaction will equal the total sum. Note that transactions which are calculated based on tab16 (such as 0003) or tab34 (such as 0040-0042) also have their VAT rates set in this table.
- Col.7  Charge for e-mail. Refers to the following cash transactions only: 0015,0024,0026,0027,0028,0042,0080,0081,0082,0090,1024,1026,1027,1028
  - Y = Cash transactions are always charged, even if the letter is sent by e-mail
  - N = If an e-mail-address is present in the patron's valid address record, no cash transaction is charged.

Please note, that in this case the line 'form-print-method EMAIL S' has to be present in the matching translation file and column 4 in print.ini has to be set to "M" to ensure that notifications are sent via e-mail if possible.

Free of charge e-mail messages:
A library that uses certain cash transactions only in order to charge postal fees does not want to charge the patron if the letter
is sent by e-mail. Set this column to N in order not to charge for postal fees.

Col. 8  description of function (up to 40 characters)

Note that in col. 1 it is possible to define a photocopy request fee differentiation according to paper type, e.g.: if paper type is A4 then the entry should be defined with the key A4-6 (A4 superimposed on 0006 key)
If paper type is B2 then define the as B2-6
If no special entry is defined defaults to 006.

Note that it is possible to define cash transactions for hold/photo requests based on the accompanying letter, different send-actions and/or or pickup locations. For example:

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!-!!!!-!!-!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!....!!!!!!!!!!!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000 ###### ## Y</td>
<td>General</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001 ###### ## Y 1.00</td>
<td>Photocopy request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002 ###### ## N</td>
<td>Hold request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003 ###### ## Y</td>
<td>Late return</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004 RLINC ## Y 5.50</td>
<td>Loan RLINC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004 UEDUC ## Y 3.00</td>
<td>Loan UEDUC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004 ###### ## Y 2.00</td>
<td>Loan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0005 ###### ## Y 0.50</td>
<td>Renewal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0006 ###### ## Y 4.50</td>
<td>Photocopy processing (standard)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that a catch-all line MUST be defined – as line 0000 in the example below.

Example of the table:

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>!1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>!!!!-!!!!-!!-!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!....!!!!!!!!!!!!-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000 ###### ## Y</td>
<td>General</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001 ###### ## Y 1.00</td>
<td>Photocopy request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002 ###### ## N</td>
<td>Hold request</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0003 ###### ## Y</td>
<td>Late return</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004 RLINC ## Y 5.50</td>
<td>Loan RLINC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004 UEDUC ## Y 3.00</td>
<td>Loan UEDUC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0004 ###### ## Y 2.00</td>
<td>Loan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab20**

Location of the table: tab directory of the BIB library

Purpose of the table: Links from BIB Headings to AUT Records
This table defines the headings (including the see references) that will be built in the ACC table of the bibliographic base, based on the fields of the authority record that "matches" the BIB ACC heading. The headings added might be the result of cross reference information, or of headings in other languages in a multilingual authority database.

The table can be up to 2000 lines long.

Note that the definition of the relationship of the library's ACC headings table to an authority library is set in tab_aut. This table defines:

• the authority database that is "linked" to a headings (ACC) table,
• and the headings list in the authority database that is used for searching for the authority record.

• Tab20 defines
• the code of the source BIB ACC heading list (i.e. the list from which ue_08 (UTIL E/8) picked up a "new" heading),
• and the code of the target BIB ACC heading list (i.e. the list on which new headings are opened)

When procedure UTIL E/8 is running, for all "-NEW-" Z01 (headings records), the system searches for the authority record that matches it.

This is done by searching the authority library's headings list, to find the appropriate record. The match between BIB heading and AUT record depends on

• the setup of the library's tab_aut table, which defines (for each BIB HEADINGS list) the AUT database(s) to search for a matching record for the list,
• and the "heading use" appropriateness) position in the 008 field in the AUT library (for MARC21 AUT databases).
• There are two types of lines in this table,
• the first line defining how to find a matching AUT record,
• and the second line defining the AUT fields that create cross-references in the BIB HEADINGS list.

When the record is found, headings are added to the library's headings (ACC) table, based on the definitions in tab20.

NOTE:
If the AUT library uses the fix_doc_preferred_usm fix_doc procedure, ALEPH automatically copies the 1xx field in an authority record to COR field when the 1xx field is updated.
In order to have this correction also trigger correction of the bibliographic library's heading and document records, field code "COR" should be added to tab20, formatted in the same way as actual "see" references (4xx).

<table>
<thead>
<tr>
<th></th>
<th>AUT</th>
<th>AUT</th>
<th>1</th>
<th>100##</th>
<th>-wi</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>COR##</td>
<td>-wi</td>
<td>0</td>
<td>SEEFP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that the fix procedure fix_doc_preferred_usm is valid also for UNIMARC headings.
Multilingual Authorities Links:
When setting up multilingual links, if the library wants separate browse lists for each language, the setup should be:

- BIB ACC browse list for ALL the languages together,
- plus individual lists for each language (e.g. SUB for all subjects, SUBE for English subjects, SUBF for French subjects, etc.).
- filter of the AUT records by language sub-field.
- The example following shows the setup for multilingual SUBJECTS, where
  - SUB is the BIB library's general subjects list, and includes subjects in all languages.
  - SUBG includes subjects in German form,
  - SUBE includes subjects in English form, and
  - SUBF includes subjects in French form.

"SUB" is searched on the AUT library's GEN headings; when a matching record is found, it is linked back to the BIB SUB, SUBG, SUBE and SUBF headings lists. The AUT record has multiple 150 fields, with $$9 differentiating between the English, German and French forms of the subject.
In some instances the subject is the same in all forms, in which case the 150 field does not have $$9, which is indicated here as 9 -.

<table>
<thead>
<tr>
<th>!</th>
<th>-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
<tr>
<td>2</td>
<td>450##</td>
</tr>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
<tr>
<td>2</td>
<td>450##</td>
</tr>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
<tr>
<td>2</td>
<td>450##</td>
</tr>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
<tr>
<td>2</td>
<td>450##</td>
</tr>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
<tr>
<td>2</td>
<td>450##</td>
</tr>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
<tr>
<td>2</td>
<td>450##</td>
</tr>
<tr>
<td>1</td>
<td>SUB</td>
</tr>
</tbody>
</table>

Structure of the table:
Col. 1 Line type (1 or 2);
Col. 2 Source. Used for line type 1 only. Code of the heading list in the BIB library whose records are used for finding an AUT record.
Col. 3 Target. Used for line type 1 only. Code of the heading list in the BIB library whose records are updated with a link to the AUT record.
Col. 4 Field tag. For line type 1: AUTHority record field tag to check for match against the BIB.
HEADING record.
For line type 2:
AUThority record field tag for building cross reference.

Col. 5 Subfield filter. Together with the next column, the subfield and
text that must be present in the AUT record field, in order for the
AUT record field be considered a match for the BIB heading.

Col. 6 Subfield text. Together with the previous column, this is the text
(in the subfield defined in the previous column) that must be
present in the AUT record's 4XX field, in order that the field be
used for enriching the BIB library's HEADINGS list. In MARC
21, sub-field $w$, 4th position (that is, $wxxx4) contains a code
that enables the generation or suppression of a cross-reference
from 4XX or 5XX fields. If the fourth position of sub-field $w$
contains a blank or contains an 'n' (or the sub-field does not
exist), the intention is that cross references will display. If the
fourth position of sub-field $w$ contains a different character, the
intention is that the cross-reference is suppressed. This MARC21
principle is applied in this column of this tab20 table.

Col. 7 Subfields to take or strip from the authority record field as a basis
for the match with the bibliographic heading.

Col. 8 Indicator. Used for line type 1 only.
Indicator (first or second) to use for stripping initial non-filing
text.

Col. 9 SEEF. Used for line type 2 only.

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!-!!!!-!-!!!!!!!-!!!-!!!!!!!-!!!-!!!!!!!-!!!-!!!!!!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>AUT</td>
<td>AUT</td>
<td>100##</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>400## w !!!n</td>
<td>-wi5</td>
<td>0 SEEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>AUT</td>
<td>AUT</td>
<td>100##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>430## w !!!n</td>
<td>-wi5</td>
<td>0 SEEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TIT</td>
<td>TIT</td>
<td>130##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>430## w !!!n</td>
<td>-wi5</td>
<td>2 SEEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab22**

Location of the table: tab directory of the **BIB** library

Purpose of the table: Z13 fields definition
tab22 defines from where to take data for the "fixed fields" of the Z13 (short-doc) record. Z13 is used for SQL reports, "quick" filter of set in OPAC, and various instances of print and display of bibliographic information in the ADM environment.

Some instances are:
WEB OPAC: BASKET and PATRON (ILL, loans, holds, SDI)
Brief display when browsing by system number
CIRC list of loans
The browse options in the CATALOG GUI client.

Some of the Z13 codes are system-defined. In addition, five additional codes can be set by the library.

For system codes YEAR is 4 positions, CALL-NO-KEY is 80 positions, and all other fields that are taken from a document record field are up to 100 positions.
For library-defined codes, up to 500 characters are taken from document record fields.
System codes must be based on field tags, whereas library-defined codes can be based on field tags, or on "paragraphs" defined in edit_paragraph.

The Z13 record automatically includes OPEN-DATE and UPDATE-DATE.
If you do not want to set a bibliographic record field for one of the lines in this table, leave cols. 3-13 blank.
The system-defined Z13 fields are:
- YEAR = year
- CALL-NO = call number
- CALL-NO-K = call number key (not currently implemented)
- AUTHOR = author
- TITLE = title
- IMPRINT = imprint
- ISBN-ISSN = ISBN/ISSN

The library-defined Z13 fields are:
- USER-DEF-1 = user defined 1
- USER-DEF-2 = user defined 2
- USER-DEF-3 = user defined 3
- USER-DEF-4 = user defined 4
- USER-DEF-5 = user defined 5

Each group of columns (tag + sub-field + position) is used to define an alternative field, if the tag in the previous group of columns is not present. The system takes the first of the alternative fields that it finds when creating the Z13 record.

Use tag 'SYS' for the BIB record's system number, if you want to map it to Z13.

Virtual fields, created by tab_expand_join, can be used, in which case expand_doc_join must be listed under CREATE-Z13 in tab_expand

When Col. 2 is set to 1, the data is taken from bib record's tag + sub-field + position.
When Col. 2 is set to 2, the data is taken from the bibliographic record, using edit_paragraph. In this case, the paragraph number must be entered in column 3, 6, 9, etc.

The system takes the first of the alternative fields that it finds when creating the Z13 record.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Z13 tag:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>Function code:</td>
</tr>
<tr>
<td></td>
<td>• 1=data taken bib record's tag + sub-field + position</td>
</tr>
<tr>
<td></td>
<td>• 2=data taken from the bib, using edit_paragraph.</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Field tag + indicators</td>
</tr>
<tr>
<td>Col. 4</td>
<td>sub-field</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Defines the position from which to take data from a fixed field. For example, if column 1 is YEAR 1, the year might be taken from a fixed field. In this case, define the position in the fixed field from which to commence taking 4 positions, counting from base 01. If the fixed field has a sub-field code, add 3 to the starting position in order to take it ($$x) into account. e.g.</td>
</tr>
<tr>
<td></td>
<td>• 0008 to define the 8th position of the 008 field in MARC21,</td>
</tr>
<tr>
<td></td>
<td>• 13 to define the 9th position of the 100 field in UNIMARC.</td>
</tr>
<tr>
<td>Col. 6</td>
<td>alternative tag if the first tag is not present</td>
</tr>
<tr>
<td>Col. 7</td>
<td>sub-field</td>
</tr>
<tr>
<td>Col. 8</td>
<td>starting position</td>
</tr>
<tr>
<td>Col. 9-11</td>
<td>as cols. 6-8</td>
</tr>
<tr>
<td>Col. 12-14</td>
<td>as cols. 6-8</td>
</tr>
<tr>
<td>Col. 15-17</td>
<td>as cols. 6-8</td>
</tr>
</tbody>
</table>

Example of table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR</td>
<td>1</td>
<td>008##</td>
<td>0008</td>
<td>260##</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL-NO</td>
<td>1</td>
<td>050##</td>
<td>a</td>
<td>LOC##</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALL-NO-K</td>
<td>1</td>
<td>LOC##</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTHOR</td>
<td>1</td>
<td>1###</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TITLE</td>
<td>1</td>
<td>245##</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMPROINT</td>
<td>1</td>
<td>260##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISBN-ISSN</td>
<td>1</td>
<td>020##</td>
<td>022##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER-DEF-1</td>
<td>2</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER-DEF-2</td>
<td>2</td>
<td>009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**tab24**

Location of the table: tab directory of the ADM library

Purpose of the table: Delivery times from remote storage

This table defines the maximum amount of time from the current time that is required for delivery from the remote storage, and the set times that items sent from the Remote storage arrive at the library.

Note that the expected delivery time from tab24 is checked against tab17, to determine that both the Remote storage and the sublibrary are open at the relevant time. If neither is open, the expected delivery time is set to the next time and the first tab24 available time. If the end-of-day is reached, the next time is the first time as defined for the next day. This new time is re-checked against tab17, and so on.

Structure of the table:

- **col 1** Sublibrary of the item
- **col. 2** Delay from request time to delivery time (in minutes)
- **col. 3-8** Time of day to expect material

Example of the table:

<table>
<thead>
<tr>
<th>Item</th>
<th>Sublibrary</th>
<th>Delay</th>
<th>Time of Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOX</td>
<td>050</td>
<td>1000</td>
<td>1500</td>
</tr>
<tr>
<td>UARCV</td>
<td>055</td>
<td>1005</td>
<td>1505</td>
</tr>
<tr>
<td>URLEC</td>
<td>060</td>
<td>1010</td>
<td>1510</td>
</tr>
<tr>
<td>URPTN</td>
<td>065</td>
<td>1015</td>
<td>1515</td>
</tr>
<tr>
<td>UUPTN</td>
<td>070</td>
<td>1020</td>
<td>1520</td>
</tr>
<tr>
<td>UEDUC</td>
<td>075</td>
<td>1025</td>
<td>1525</td>
</tr>
<tr>
<td>RLINC</td>
<td>080</td>
<td>1030</td>
<td>1530</td>
</tr>
<tr>
<td>UHLTH</td>
<td>085</td>
<td>1035</td>
<td>1535</td>
</tr>
<tr>
<td>ULAW</td>
<td>090</td>
<td>1040</td>
<td>1540</td>
</tr>
<tr>
<td>ULINC</td>
<td>095</td>
<td>1045</td>
<td>1545</td>
</tr>
<tr>
<td>UMUSI</td>
<td>100</td>
<td>1050</td>
<td>1550</td>
</tr>
</tbody>
</table>

...
Material types that begin with ISSXX have special functionality. Items marked ISSXX will sort according to cols. 2 and 3 of tab_z30_sort. All other items sort according to cols. 4 and 5 of tab_z30_sort.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>ALPHA</td>
</tr>
<tr>
<td>Col. 3</td>
<td>Name</td>
</tr>
<tr>
<td>Col. 4</td>
<td>Magnetic media definition. N/Y. This is used for identification of magnetic media in order to stop magnetization or demagnetization when using 3M equipment.</td>
</tr>
</tbody>
</table>

**Note:** In order to manage magnetization, a section [MagneticMedia966] has been added to the file circ.ini in the online clients. The only line in this section is Port=nn. When set to 0 this option is turned off. This should be set to the serial port the 966 module is attached to (if there is such a module).

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>!</th>
<th>!</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOK</td>
<td>L</td>
<td>Book</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ISSUE</td>
<td>L</td>
<td>Serial</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>ISSBD</td>
<td>L</td>
<td>Bound Volume</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>VIDEO</td>
<td>L</td>
<td>Video Cassette</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>AUDIO</td>
<td>L</td>
<td>Audio Cassette</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>DIGIT</td>
<td>L</td>
<td>Digital Form</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>MANUS</td>
<td>L</td>
<td>Manuscript</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Material type ISSBD for bound issues. The material-type is changed from ISSUE to ISSBD after the binding process. In all the usual functions (SORT, DISPLAY etc.) bound issue will be treated as a regular issue.

**tab27**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Hold requests send method configuration

This table - tab27 - defines an aspect of hold and ILL requests:
- whether item is assigned to "hold shelf" or immediately loaned when it becomes available

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>sub-library of the item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 2</td>
<td>item status</td>
</tr>
</tbody>
</table>
| Col. 3 | Pickup location. Location to which item will be sent (either to be
picked up by the patron, or actually sent to the patron)

Col. 4 process option. These are the actions that will result from operator clicking "Letter" on the Holds Requests display (when returning item,) or performing Print letter - Hold request filled in GUI CIRC.

- **01** generates a pickup letter for the patron and a hold shelf slip for the item. The loan will be performed in a regular manner when the item is picked up. This should be used for the normal library "hold shelf" functionality.
- **02** generates a delivery slip for the item, and automatically loans the item to the patron. This can be used for home delivery.
- **03** generates a pickup letter for the patron and a delivery slip for the item, and automatically loans the item to the patron. This can be used if the library puts the requested material in the patron's mailbox, and also wants to inform the patron that the material is now available.
- **04** reading room: performs option 02 and creates a Z310 record for Reading Room control.
- **05** for items received through ILL: loan to the pickup sublibrary, taking the pickup sublibrary from Z40. If there is no pickup sublibrary, loans to the Home Library of the requesting patron. This requires that the sublibrary code is registered as a patron with a Z303 and a Z305 record.

Example of the table:

<table>
<thead>
<tr>
<th>!1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!−!!−!!!!!−!!</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### 73 ##### 03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### 72 ##### 02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### ## UARCV 04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### ## ILLDT 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### ## URLEC 04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>URPTN 70 URPTN 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>URPTN 71 URPTN 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### ## URPTN 04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UUPTN 75 UUPTN 05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>##### ## UUPTN 01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UEDUC 75 UEDUC 05</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab30**

Location of the table: tab directory of the ADM library

Purpose of the table: Patron registration renewal

Tab30 allows the library to define defaults for two elements with regard to expiry of patron registration:
• How many days before the expiry date should the system warn the operator of an approaching expiry date (in CIRC). If a patron borrows or renews an item within the defined period, the system gives a warning “expiry date approaching” together with the expiry date. This warning is only displayed if check_circ_1_c_b for LOAN or RENEW is defined in tab_check_circ. Note that dependent on the switch 18 BOR-EXPIRY-DUE-DATE in tab100 the due date of the item can be set so that it does not fall beyond the expiry date.

• The default renewal period. This can be seen on-line when the renew button (on the Update Local Patrons screen) is activated.

Structure of the table:
Col. 1     patron status
Col. 2     sub-library (use # for wildcard)
Col. 3     registration renewal warning (number of days before expiry)
Col. 4     registration renewal period: years
Col. 5     registration renewal period: months

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-!</td>
<td>-!</td>
<td>-!</td>
<td>-!</td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>UUPTN</td>
<td>014</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>14</td>
<td>UUPTN</td>
<td>014</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>15</td>
<td>UUPTN</td>
<td>014</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>16</td>
<td>UUPTN</td>
<td>014</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>#####</td>
<td>028</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>#####</td>
<td>028</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>02</td>
<td>#####</td>
<td>028</td>
<td>01</td>
<td>00</td>
</tr>
<tr>
<td>08</td>
<td>#####</td>
<td>028</td>
<td>01</td>
<td>00</td>
</tr>
</tbody>
</table>

**tab31**

Location of the table: tab directory of the ADM library

Purpose of the table: Patron statuses

This table defines default privileges and restrictions for the patron based on the patron status. When a new patron is created online the system will use these defaults for creating patron privileges and other definitions. Status MUST be defined here in order to be valid in the system. The translation of the codes to patron groups (e.g. 01=under graduate) should also be registered under the BOR-STATUS in either pc_tab_exp_field.<lng> or pc_tab_exp_field_extended.<lng> so that they will appear in the patron status pull-down menu.

The Default Hold Priority values should also be registered under HOLD-PRIORITY in pc_tab_exp_field.<lng> so they will appear in the hold priority drop-down menu.

Structure of the table:
col. 1 sub-library
col. 2 patron status (01-99)
col. 3 loan permission (Y/N)
col. 4 photo permission (Y/N)
col. 5 override permission: can an override on a loan transaction be invoked or is the loan blocked
col. 6 multiple hold permission: can the patron place multiple holds on the same record?
col. 7 check loan: should the system check if the patron can loan the item?
col. 8 hold permission (Y/N): can the patron place hold requests?
col. 9 renew permission: can the patron renew items in the OPAC?
col. 10 ignore late return: should items returned late be registered (for fining)?
col. 11 photocopy charge
  • C-charge
  • F-free
col. 12 expiry date operator
  • +=add to current date
  • A=actual date
col. 13 expiry date operator type (when col. 12 has ‘+’)
  • D=day, M=month, Y=year
col. 14 expiry date parameter
  • if col. 12 has ‘4’; no. of days to add
  • if col. 12 has ‘A’ actual date
col. 15 cash overspend limit - last two digits are decimal
col. 16 request hold for item on shelf (Y/N)
col. 17 loan display – Y/N
  allows for the display of patron information in the WEB OPAC for an item loaned out – e.g. a department library that has permanent loan items.
col. 18 reading room permission
col. 19 Default Hold priority

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>14</th>
<th>15</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td>!!!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ALEPH 10 N N N N Y Y N N C + D 000000028 0000000000 N Y N 30
ALEPH # # N N N N Y Y N N C A 20041231 0000000000 N Y N 30
##### 01 Y Y Y N Y Y Y Y N C A 20041001 0000000000 N N Y 30
##### 02 Y Y Y N Y Y Y Y N C A 20041001 0000000000 N N Y 30
##### 03 Y Y Y N Y Y Y Y F A 20041001 0000000000 Y N Y 20
##### 04 Y Y N N Y N N N N C A 20041001 0000000000 N N N 30
##### 05 Y Y Y N Y Y N N C A 20041001 0000000000 N N Y 30
tab32

Location of the table: tab directory of the ADM library

Purpose of the table: Overdue letter intervals

Structure of the table:

- col. 1: sub library (##### to signify all sub-libraries)
- col. 2: item status (## for all)
- col. 3: patron status (## for all)
- col. 4: current letter number in the loan transaction record (Z36). If the letter number is '00', no previous letters have been sent, and therefore this is the first notice. If the letter number is '01', this is the second notice, and so on.
- col. 5: Number of days since:
  - 4 = 00 - number of days since due date overdue
  - 4 > 00 - number of days since last overdue letter
  - 4 = 09 - number of days since due date for items which are "claimed returned". This column is used when p-cir-50 (Print Overdue and Lost Billing Notices) is run for "Bills for lost material that was claimed to have been returned".
- col. 6: Number of days for requested items: if
  - 4 = 00 - number of days since due date overdue
  - 4 > 00 - number of days since last overdue letter
  - 4 = 09 - number of days since due date for items which are "claimed returned". This column is used when p-cir-50 (Print Overdue and Lost Billing Notices) is run for "Bills for lost material that was claimed to have been returned"
- col. 7: action:
  - O - overdue letter; can incur overdue letter charge, using column 8 and tab18
  - L - lost letter, can incur lost charges, using lines 0040, 0041, 0042, 0054 and 0055 in tab18, and tab34. The tab18 lines do not have to be defined in column 8, they are automatically assumed.
- col. 8: transaction cost - line no. in tab18

Note: The system reads the table from the top down. Item status "##" is used to denote "all" item statuses.

Example of the table:
Location of the table: tab directory of the BIB and ADM libraries

Purpose of the table: Defines the format of the BIB-ADM record that displays in the navigation window in the various GUI modules.

This table should be present in each BIB and ADM library. The table in the BIB library defines the fields from the BIB record. The table in the ADM library defines the fields from the ADM record. Note that the order of the fields in the record, and not in the table, determines the display order of the fields.

**NOTE** that the "D" lines (D in col. 4) of the BIB library's /tab/edit_field.<lng> table, set the formatting of the BIB field.

Structure of the table:
- Col. 1  tag
- Col. 2  ALPHA
- Col. 3  caption

Example of the table in the ADM library

```
! 1 2 3

!!!!!!-!-!!!-!!-!!!-!!-!!-!!-!!-!!!-!!-!!-

##### 11 ## 00 000 000 O 0080
##### 11 ## 01 000 000 O 0081
##### 11 ## 02 000 000 O 0082
##### 11 ## 03 000 000 L
##### 11 ## 09 021 021 L

!*

##### 31 ## 00 005 000 O 0080
##### 31 ## 01 007 000 O 0081
##### 31 ## 02 014 000 O 0082
##### 31 ## 03 014 014 L
##### 31 ## 09 021 021 L

!*

##### 12 ## 00 005 000 O 0080
##### 12 ## 01 007 000 O 0081
##### 12 ## 02 014 000 O 0082
##### 12 ## 03 021 021 L
##### 12 ## 09 021 021 L
```

Example of the table in the ADM library

```
FMT  L FormatADM
ACQ## L Acquisition note
Z68## L Order note
```
Example of the table in the BIB library

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

020## L ISBN
022## L ISSN
100## L Author
245## L Title
250## L Edition
260## L Imprint
300## L Collation
310## L Frequency
4#### L Series
70### L Added Author
74### L Added Title

**tab34**

**Location of the table: tab directory of the ADM library**

**Purpose of the table: Lost material costs**

Tab34 is an optional table that computes the charges to be added to the patron's cash record in case an item is changed to lost, (either through the LOST function in the Circulation module, or as a result of batch services p-cir-50, -51 and -52, which use tab32 to declare an item as lost).

Note that if the library does not define default charges in this table, the system will not be able to compute a default charge.

It is possible for the user to manually input the replacement cost. This is configurable from circ.ini:

```
[Lost]
ManualReplacement=Y/N
```

The charges are defined for each item-patron combination. There are three types of charges:

- notice,
- replacement handling
- price.

When a lost item is found, the charges for handling and price are waived.
Structure of the table:

Col. 1 sub library – use # for wildcard
Col. 2 item status
Col. 3 patron status
Col. 4 call number. For all call numbers, use only one # in the first position
Col. 5 collection
Col. 6 notice cost. This amount will not be credited to the patron if the item is returned. Transaction 0042 in tab18 must be set to Y.
Col. 7 handling cost. This amount will be credited to the patron if the item is returned. Transaction 0040 in tab18 must be set to Y.
Col. 8 fixed price. This amount will be credited to the patron if the item is returned. Transaction 0041 in tab18 must be set to Y.
You may also indicate that Z30-PRICE will be consulted prior to this column by using the prefix “/” (e.g. I/420.00). This applies only if Z30-PRICE is a decimal number other than zero before the first space in the field.
Col. 9 MARC price (placeholder for Danish implementation)
Col. 10 patron factor (Not yet implemented)

Example of the table:

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

!!!!!!-!!!-!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!-!!!!!-

!!!!!!-!!!!!!!!!!!-!!!!!!-!!!!!!>

UARCV 70 ## #
RARE 1.50 5.00 400.00

UARCV 71 ## #
RARE 1.50 5.00 450.00

UMUSI 70 ## #
RARE 1.50 5.00 350.00

**tab35**

Location of the table: tab directory of the ADM library

Purpose of the table: Location numbers and other information for EDI orders

Every sub-library may have different EDI ID numbers at different vendors. In addition, different sub-libraries can have one account at the same vendor.
Tab35 allows for the defining of such numbers for each branch library. In addition the table allows for the registration of each sub-library’s VAT number.
Structure of the table:

Col. 1  sub library or order unit
Col. 2  Vendor code
Col. 3  Customer’s EDI code
Col. 4  Customer’s EDI code type
  • 31B for US SAN,
  • 14 for EAN-13
  • 91 for id assigned by supplier
  • 92 for id assigned by customer.
Col. 5  Not in use
Col. 6  Not in use
Col. 7  VAT number
Col. 8  Email address for reporting errors in order processing

Example of the table:

<table>
<thead>
<tr>
<th>Sub-library</th>
<th>Vendor Code</th>
<th>EDI Code</th>
<th>EDI Code Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UARCV   HAR</td>
<td>IOWA-SER</td>
<td>091</td>
<td>IL</td>
</tr>
<tr>
<td>111 222 331</td>
<td><a href="mailto:yifat.lulav@exlibris.co.il">yifat.lulav@exlibris.co.il</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORDU1 HAR</td>
<td>IOWA-SER</td>
<td>091</td>
<td>IL</td>
</tr>
<tr>
<td>111 222 331</td>
<td><a href="mailto:yifat.lulav@exlibris.co.il">yifat.lulav@exlibris.co.il</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UARCV SWETS</td>
<td>IOWA-SER</td>
<td>091</td>
<td>IL</td>
</tr>
<tr>
<td>111 222 331</td>
<td><a href="mailto:yossi.loss@exlibris.co.il">yossi.loss@exlibris.co.il</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that if EDI communication is active for a vendor, the vendor’s EDI code MUST be defined in the "vendor EDI code" field and the code type MUST be defined in "vendor EDI type" field. Both fields are part of the online ACQ vendor form.

**tab36**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** automatic opening of monograph-type items in Acquisitions

The system will automatically open items for monograph orders as per the number registered in the number of units field in the order record if

• CREATE-ITM-FORM-ORDER-M in tab100 is set to Y.
• the checkbox “create item records” on the monograph order form is flagged.

Tab36 allows for defining the default values (sub-library, item status, material type, collection, location and location type) for such automatic item creation.

Structure of the table:

Col. 1  Sub-library (use # for wild card)
Col. 2  Item status
Col. 3  Material type
Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UUPTN</td>
<td>70</td>
<td>BOOK</td>
<td>GEN</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UARCV</td>
<td>70</td>
<td>BOOK</td>
<td>GEN</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UELEC</td>
<td>11</td>
<td>DIGIT</td>
<td>GEN</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UGDOC</td>
<td>70</td>
<td>BOOK</td>
<td>GEN</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#####</td>
<td>70</td>
<td>BOOK</td>
<td>GEN</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab37**

Location of the table: tab directory of the ADM library

Purpose of the table: Pick up list configuration for hold requests

This table - tab37 - defines one aspect of hold and ILL requests:
- drop-down list of pickup and deliver locations for hold requests (dependent on item and patron status, and on sub-library of item).

NOTE the following:
If the item sublibrary, item status, item process status, patron status and pickup location are not defined in this table, the item cannot be requested.

If you are using a specific item status in col.2 (i.e. not ##), make sure that the specific lines come before the general lines, since the table is read from top to bottom, and the first match is always taken. The same holds true for col.3 (item process status) and col.4 (patron status).

All item statuses are matched on ## in the first line, and the second line will never be used.

**Example of correct setup:**

```
! ULINC 14 ## ## # UEDUC
! ULINC ## ## # UEDUC
!
```

**Example of incorrect setup:**

```
! ULINC ## ## ## # UEDUC
! ULINC 14 ## ## # UEDUC
```

If you have a combination of one specific item status with all patron statuses and a combination of all item statuses with a specific patron status, define these lines first.

For example:
You want all items with item status 14 except those requested by a patron with patron status 05 to be ready for pickup at UEDUC, and all items with patron status 05 to be ready for pickup at UEDUC, and all items requested by a patron with patron status 05 to be ready for pickup at UHLTH. In order to achieve this specify lines for the combination of the two specific statuses:

Example:

<table>
<thead>
<tr>
<th>ULINC 14 ## 05 # UHLTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULINC 14 ## ## # UEDUC</td>
</tr>
<tr>
<td>ULINC ## ## 05 # UHLTH</td>
</tr>
<tr>
<td>ULINC ## ## ## # ULINC</td>
</tr>
</tbody>
</table>

You can define up to 1000 different pickup locations for each combination. Up to ten different locations can be entered on each line. If there is more than one line per combination (to allow for more than ten pickup locations), leave columns 1-5 empty in the other lines.

For example:

<table>
<thead>
<tr>
<th>UHLTH ## ## ## # XXX01 XXX02 XXX03 XXX04 XXX05 XXX06 XXX07 XXX08 XXX09 XXX10</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXX11</td>
</tr>
<tr>
<td>XXX12</td>
</tr>
<tr>
<td>XXX13</td>
</tr>
<tr>
<td>XXX14</td>
</tr>
<tr>
<td>XXX15</td>
</tr>
<tr>
<td>XXX16</td>
</tr>
<tr>
<td>XXX17</td>
</tr>
<tr>
<td>XXX18</td>
</tr>
<tr>
<td>XXX19</td>
</tr>
<tr>
<td>XXX20</td>
</tr>
<tr>
<td>XXX21</td>
</tr>
<tr>
<td>XXX22</td>
</tr>
<tr>
<td>XXX23</td>
</tr>
</tbody>
</table>

**Note** the virtual item status **IL** which may be used for setting the pickup location list in the WEB-OPAC ILL request form. To enable the pickup location list to be sensitive to the ILL Patron status, the patron ILL sublibrary should be defined in Col.1, use **IL** as item status in Col.2 and insert the ILL patron status in Col.4.

**Display of the pickup sublibrary drop-down list**

Only the most specific line matching sublibrary/item-status/item-process-status is taken for the dropdown list (i.e. although ## lines would match they are not included in the dropdown list if a more specific line was found).

**Note the following:** Column 5 (Availability of item) can be used to filter out a sub-library from the drop-down list of pickup locations.

If column 5 is set to "N", the pickup location will display only if the item is not available (i.e. item is on loan or on hold shelf).

There are two correct ways to use this:

by exclusion:

<table>
<thead>
<tr>
<th>UHLTH ## ## ## N HOME</th>
</tr>
</thead>
<tbody>
<tr>
<td>UHLTH ## ## # UHLTH UEDUC</td>
</tr>
</tbody>
</table>

by inclusion:

<table>
<thead>
<tr>
<th>UHLTH ## ## ## Y UHLTH UEDUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>UHLTH ## ## ## N UHLTH UEDUC HOME</td>
</tr>
</tbody>
</table>

**Structure of the table:**

Col. 1 Sublibrary of the item
Col. 2  Status of item being requested
Col. 3  Process Status of item being requested
Col. 4  Status of the patron
Col. 5  Availability of item.
   If "Y" this line only applies to available items (i.e. on the shelf).
   If "N" this line only applies to unavailable items (i.e. on loan, hold waiting to be picked up or in transit).
   Use # to indicate that the line is insensitive to availability of the item
Col. 6-15
   Pickup location. Location to which item will be sent (either to be picked up by the patron, or actually sent to the patron).

Example of the table:

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!!!-!!-!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!-!!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

##### IL ## 03 # HOME ULINC UMUSI USCI UUPTN
##### ## ## 11 # ILLDT ULINC UMUSI USCI UUPTN
UARCV ## ## ## # UARCV UEDUC ULAW ULINC UMUSI USCI UUPTN RLINC
ULINC 42 DP 02 # UMUSI
ULINC 42 DP ## # ULINC UUPTN UEDUC UHLTH ULAW
ULINC 73 ## ## # BOX HOME ULINC UEDUC UHLTH ULAW UMUSI USCI UUPTN RLINC
ULINC ## ## ## # ULINC UEDUC UHLTH ULAW UMUSI USCI UUPTN HOME RLINC

**NOTE** that in tab_sub_library.<lng>, it is possible to define different *library types*:
1 - full sub library (patron and items)
2 - patron sub library only (no items)
3 - special
4 - reading room sub library (1 + special request handling)
5 – ordering unit

For the purpose of tab37, it is possible, for example, to define different *mailboxes* which need to also be defined in tab_sub_library:

Example from tab_sub_library.<lng>:

```
library - type
BOX1  3  USM50 L Pickup in Forum
```

Where *Box1* is the “library”, and the library type is 3=special

**tab37_campus_filter**

Location of the table: tab directory of the ADM library
Purpose of the table: Hold requests filter table
This table is used in conjunction with p-cir-24, the hold request router program. Set this table up to remove holds from consideration that have certain characteristics.

For example, the following line will not allow you to recall a ULAW book from a faculty member (03).

```
ULAW  ## ## 03 N N
```

The table is read from the top down, so put specific lines at the top and the more general lines at the bottom.

**Structure of the table:**

- Col. 1: Sublibrary of the requested item
- Col. 2: Status of the requested item
- Col. 3: Processing status of the requested item
- Col. 4: Patron status of the person who is requesting the item
- Col. 5: Patron status of the person who currently has the item out
- Col. 6: Allow hold (Y/N)
- Col. 7: Allow recall (Y/N)

**Example of the table:**

```
!1  2  3  4  5  6  7
!!!!-!-!-!-!-!-!-!
##### ## ## ## ## Y N
UARCV ## ## ## ## Y N
UELEC ## ## ## ## Y N
ULAW ## ## 03 N N
ULAW ## ## 02 N N
UGDOC ## ## ## N N
##### ## 11 16 Y N
```

**tab37_campus_preferred**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Preferred supplier list for hold requests

This table is used in conjunction with the holds request router job, p-cir-24, and should be used to determine the preferred supplier list for items being requested.

Column 1 defines the pickup location for the hold. So, for example, if the patron wants to pick up an item near his home at campus A (CMPA) the following line indicates that the system should try to supply this from CMPA, then CMPA2, etc, and through the list.

```
CMPA  CMPA  CMPA2 CMPA3 CMPB1
```
If no item is available from the local list, the system will try to supply it from the
global list:

##### CMPA CMPB CMPC CMPD CMPE CMPF CMPG CMPH
##### CMPI CMPJ

Not that the above lines are concatenated.

Structure of the table:

| Col 1 | Pickup sublibrary of the hold group |
| Col 2-10 | Preferred provider for the request |

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ILL01</td>
<td>UELEC</td>
<td>UARCV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!EXL:2222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ILL02</td>
<td>ULAW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>######</td>
<td>URLEC</td>
<td>UEDUD</td>
<td>UHLTH</td>
<td>ULINC</td>
<td>URD</td>
<td>RLINK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ULAW</td>
<td>ULAW</td>
<td>ULINC</td>
<td>UHLTH</td>
<td>USCI</td>
<td>UED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USCI</td>
<td>UHLTH</td>
<td>USCI</td>
<td>UELEC</td>
<td>ULINC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UED</td>
<td>UED</td>
<td>UUPTN</td>
<td>ULINC</td>
<td>USCI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>######</td>
<td>UARCV</td>
<td>ULAW</td>
<td>UELEC</td>
<td>UED</td>
<td>UGDOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab38**

Location of the table: tab directory of the **ADM** library

Purpose of the table: configuration for photocopy requests

Tab38 defines the configuration for photocopy requests. This configuration includes:

- target library for photocopy request
- item status
- patron status
- pickup library
- delivery method

**NOTE:** if no combination is defined for an item/patron, the photocopy request will
not be allowed.

Note too that **HOME** is equal to **delivery**: a photocopy letter only will be printed
All other sub-library codes are pickup. A photocopy letter and a pickup notice are
printed. For this reason it is not possible to define HOME and a sub-library code in
the same line.

Note too that it is possible to charge a different fee for home delivery (cash
transaction 0008 – tab18.<lng>).

Structure of the table:
Col. 1 source sub-library
Col. 2 source item status
Col. 3 patron status
Cols. 4-13 pickup location

Example of the table:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UGDOC</td>
<td>##</td>
<td>##</td>
<td>UGDOC</td>
</tr>
<tr>
<td>UEDUC</td>
<td>##</td>
<td>##</td>
<td>UEDUC</td>
</tr>
<tr>
<td>UHLTH</td>
<td>##</td>
<td>##</td>
<td>UHLTH</td>
</tr>
<tr>
<td>ULAW</td>
<td>##</td>
<td>##</td>
<td>ULAW</td>
</tr>
<tr>
<td>ULINC</td>
<td>##</td>
<td>##</td>
<td>ULINC</td>
</tr>
<tr>
<td>ULINC</td>
<td>##</td>
<td>##</td>
<td>HOME</td>
</tr>
<tr>
<td>UMUSI</td>
<td>##</td>
<td>##</td>
<td>UMUSI</td>
</tr>
</tbody>
</table>

**NOTE** that in tab_sub_library.<lng>, it is possible to define different library types:
- 1 - full sub library (patron and items)
- 2 - patron sub library only (no items)
- 3 - special
- 4 - reading room sub library (1 + special request handling)
- 5 – order units

For the purpose of tab38, it is possible, for example, to define different mailboxes which need to also be defined in tab_sub_library.<lng>:

e.g. from tab_sub_library.<lng>:

```plaintext
library - type
BOX1 3 USM50 L Pickup in Forum
```

Where *Box1* is the “library”, and the library type is 3=special

**tab39**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Hold request daemon configuration

The tab39 table is used in conjunction with the hold request daemon (ue_06 – UTIL E/6). For this purpose tab39 specifies several parameters for processing hold requests. One of these parameters is a suffix or extension that is attached to each print file that is produced as a result of hold requests coming from the Circulation module and the Web OPAC. The suffix is identified by the print daemon which directs it to a specified printer.

**Structure of the table:**
- Col. 1 item sub-library (##### for all sub libraries)
Col. 2  item status (## for all statuses)
Col. 3  Item process status (## for all item statuses);
Col. 4  collection (##### for all collections)
Col. 5  Call number range. This is the lower limit of the range. Leave empty if you want to limit the range only on the upper limit. Note: if this field is filled in and the field in column 6 is empty, the request daemon ue_06 will only retrieve the item(s) with this exact call number
Col. 6  Call number range. This is the upper limit of the range. Leave empty if you want to limit the range only on the lower limit.
Col. 7  Print hold wait letters?
This switch determines whether "hold request not filled" letters (hold-wait-letter.xsl) will be printed or not.
Col. 8  How many hold slips?
This switch determines how many hold request slips (hold-request-slip.xsl) will be printed.
Col. 9  Format number of hold slip template (hold-request-slip.xsl)
Col. 10 Print ID. Hold slip Print ID; that is, the extension that allows the Print Daemon to identify the file [Note: Print ID must be entered in lowercase.]
Col. 11 Format number of "hold request not filled" letter template (hold-request-wait.xsl)
Col. 12 Print ID. Hold request not filled Print ID; that is, the extension that allows the Print Daemon to identify the file [Note: Print ID must be entered in lowercase.]
Col. 13 Request type
Filter daemon according to the type of request :
- R   - Rush request
- N   - Normal request
- blank - all requests

Example of the table:

<table>
<thead>
<tr>
<th>Col. 2</th>
<th>Col. 3</th>
<th>Col. 4</th>
<th>Col. 5 Call Number Range</th>
<th>Col. 7 Print Hold Wait Letters?</th>
<th>Col. 8 How Many Hold Slips?</th>
<th>Col. 9 Format Hold Slip Template</th>
<th>Col. 10 Print ID Hold Slip</th>
<th>Col. 11 Format Hold Request Not Filled Letter Template</th>
<th>Col. 12 Print ID Hold Request Not Filled</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td></td>
<td></td>
<td>00 depo.slip 01 depo.wait</td>
<td>Y</td>
<td>1</td>
<td>0</td>
<td>00 depo.slip 01 depo.wait</td>
<td>Y 1 00 holdslip 00 holdwait</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>00 holdslip 00 holdwait</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab40.<lng>**

Location of the table: tab directory of the ADM library
Purpose of the table: Collection codes and names

This table defines the expansion of the collection code as entered in the Z-30-COLLECTION FIELD in items, to the expanded form to be displayed in the OPAC.

Structure of the table:

Col. 1  Collection code as entered in the Z30-COLLECTION field of the item record.
Col. 2  Sub-library (may use #)
Col. 3  ALPHA
Col. 4  Collection name as will display in the OPAC (maximum 80 characters)

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>!!!-!!!!-!!!-!!!!!!!!!!!!!!!!!!!!!!!&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABST ULINC L Abstracts</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASIAN ULINC L Asian Collection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AV ULINC L Audio-Visual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEN ULINC L General</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRANT ULINC L Grant Collection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEE ULINC L Lee Collection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SER ULINC L Serials/Microforms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MICR ULINC L Microforms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REF ULINC L Reference</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESV ULINC L Reserves</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab41**

Location of the table: tab directory of the ADM library

Purpose of the table: Photocopy request daemon configuration

The tab41 table is used in conjunction with the photocopy request daemon (ue_06 – UTIL E/6). For this purpose tab41 specifies several parameters for processing photocopy requests. One of these parameters is a suffix or extension that is attached to each print file that is produced as a result of hold requests coming from the Circulation module and the Web OPAC. The suffix is identified by the print daemon which directs it to a specified printer.

Structure of the table:

Col. 1  item sub-library (##### for all sub libraries)
Col. 2  item status (## for all statuses)
Col. 3  collection (##### for all collections)
Col. 4  Call number range. This is the lower limit of the range. Leave empty if you want to limit the range only on the upper limit. Note: if this field is filled in and the field in column 5 is empty, the request daemon ue_06 will only retrieve the item(s) with this exact call number
Col. 5 Call number range. This is the upper limit of the range. Leave empty if you want to limit the range only on the lower limit.

Col. 6 Print photocopy wait letters? (Y/N) determines whether "photocopy request not filled" letters (photo-wait-letter.xsl) will be printed or not.

Col. 7 Print photocopy slips? (Y/N) determines whether photocopy request slips (photo-request-slip.xsl) will be printed or not.

Col. 8 Number of slips - how many copies of each photocopy request slip will be printed.

Col. 9 Format number of photocopy slip template (photo-request-slip.xsl)

Col. 10 Photocopy slip Print ID - the extension that allows the Print Daemon to identify the file [Note: Print ID must be entered in lowercase.]

Col. 11 Format number of "photocopy request not filled" letter template (photo-request-wait.xsl)

Col. 12 Photocopy request not filled Print ID – the extension that allows the Print Daemon to identify the file [Note: Print ID must be entered in lowercase.]

Col. 13 Format number of "photocopy request filled" letter template (photo-request-letter-p.xsl or photo-request-letter-d.xsl)

Col. 14 Photocopy request filled Print ID – the extension that allows the Print Daemon to identify the file [Note: Print ID must be entered in lowercase.]

Example of the table:

```
##### ## #####
YY10 0 photoslip 00 photowait 00 photofilled
```

tab42

Location of the table: tab directory of the ADM library

Purpose of the table: Automatic update of the processing status field in the item record.

This table controls automatic update of the processing status field in the item record. Automatic update is triggered by various actions or changes in the Acquisitions, Serials and Items modules, and dependent on sub-library and action. If the action trigger is listed in this table, the item processing status will be updated as defined.

**ACQUISITIONS**

Changing the order status acts as the trigger. The order statuses are:

- **NEW** - Newly created order.
- **WP** - Waiting for processing.
- **PS** - Processing started.
- **WB** - Waiting for budget confirmation.
- **QSV** - Query before sending order.
- **CNB** - Cancelled, no budget.
- **DNB** - Delayed, no budget.
- **RSV** - Ready to send to vendor.
- **SV** - Order sent to vendor.
- **SV+** - This is not an order status. It is the "action" when material arrival is registered.
- **VC** - Vendor cancellation of order.
- **LC** - Library cancellation of order.
- **CLS** - Closing of order.

**SERIALS**
- **OP** - Item opened
- **CL** - Item claimed
- **AR** - Item arrival registered
- **NP** - Item not published (this is not a trigger, it is a manually entered processing status.
- **NO** - Item opened, when Z16 is not active
- **UAR** - Unarrive item

**ITEMS**
- **BD** - Binding (occurs when issues are processed for binding)
- **PL** - Print label utility
- **DP** - Remote Storage functionality

**Note:** The process status of the new bound issue, will be taken automatically from tab42, section **ITEM** and status **BD**. If the line doesn’t exist in tab42, then the process status will be taken from the manual input of the user.

**Structure of the table:**

Col. 1 Module
- **ACQ** = Acquisitions
- **SERIAL**= Serials
- **ITEM**= Items

Col. 2 Sub-library (Use # for wild card)

Col. 3 Action code that will trigger the change of the item processing status. Use **DP** as the trigger for an **ITEM** when you want the processing status updated because the item has a remote storage-id.

Col. 4 The processing status code. Enter blanks to remove the processing status code. The processing status codes are not system controlled, but the following has functionality in the system:
- **BD** (bound item - does not display in list of items)

Col 5 Previous Item processing status:
Only applicable when column 1. is **ACQ**. The current Item
Process Status is checked, and the process status will be changed only if there is a match.
This can be used in order to ensure that a Process Status entered manually will be retained.
Use "##" to indicate that the current Item Process Status is irrelevant, and the change is to be made in all cases.

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>ACQ</td>
<td>ULINC NEW OR ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQ</td>
<td>ULINC RSV OR ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQ</td>
<td>ULINC SV OI ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQ</td>
<td>ULINC CLS ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQ</td>
<td>##### NEW OI ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQ</td>
<td>##### WP OI ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACQ</td>
<td>##### PS OI ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SERIAL</td>
<td>##### UAR NA ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SERIAL</td>
<td>##### NP NP ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SERIAL</td>
<td>##### OC NP ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM</td>
<td>##### PL ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM</td>
<td>##### BD SB ##</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tab43**

Location of the table: tab directory of the **ADM** library

Purpose of the table: Advanced booking schedule

Tab43 is an expanded table for defining advanced booking parameters. The table allows for the definition of start day and hour, as well as end day and hour.
The table allows for defining "closed" times, independent of tab17 (library hours).

It is possible to define days where the slots are different from the schedule as a whole. To do so, use the E line type to define Exact day.

The definition can be of a slot different from the one already set for the schedule period.

For example:

**The regular slot is:**

```
P ###### 60 20040101 01 0900 01 1100;
```

The P stands for settings of a period and will include the date until which the definition is valid.

**The settings for the irregular slot are:**

```
E ###### 60 20030930 00 0900 00 1700
```
- The E indicates that the following line is a definition of an exact date and not an end of period as in lines of type P.
- The 00 in an E type of line, indicates that the definition is for the date listed in column 4.
- 00 in a P type of line still stands for Sunday.

The slot in this example starts on 09:00 and ends at 17:00 on the same day.

It can also be defined to end the next day in the following way:

```
E ###### 60 20030930 00 0900 01 0900;
```

In this case the slot starts at 09:00 on the current day and ends at 09:00 one day after.

Another option is to define a date that is included in the schedule period and on which no slots will be open.
To do so, leave columns 5-8 empty in the following way:

```
E ###### 60 20030930
```

This indicates that on September 30th, there are no slots available for item status 60.
To define two close days in a row, set a line for each day.

It is possible to use ## instead of the year, month or day listed in column 4 of the E type of lines.
For example:

```
E ###### 60 2003##01
```

Indicates that there are no open time slots for the 1st of every month on 2003.
```
E ###### 60 ####0930
```

Indicates that there are no open time slots on September 30th of each year.
It is NOT possible to use ## as part of the date in the P type of lines.

Structure of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Line Type:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• P - Slots for period date</td>
</tr>
<tr>
<td></td>
<td>• E - Slots for exact date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Col. 2</th>
<th>Sub-library (use #### for wildcards)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. 3</td>
<td>Item Status</td>
</tr>
<tr>
<td>Col. 4</td>
<td>End Period Date</td>
</tr>
<tr>
<td>Col. 5</td>
<td>Beginning of time slot - Day (00=Sunday, 06= Saturday)</td>
</tr>
<tr>
<td>Col. 6</td>
<td>Beginning of time slot - Hour</td>
</tr>
<tr>
<td>Col. 7</td>
<td>End of time slot (00-10) - Day (number of days that have elapsed since Sunday)</td>
</tr>
<tr>
<td>Col. 8</td>
<td>End of time slot - Hour</td>
</tr>
</tbody>
</table>

**NOTE:** Every row should end with a semi-colon (;). See following example.

Note that the number of advanced booking days that will display in the web OPAC (i.e. the number of days ahead that a person may book an item), is defined in www_server.conf – setenv www_item_schedule.

Example of the table:
tab44

Location of the table: tab directory of the ADM library

Purpose of the table: Defines the period that requested items will be held on the hold shelf

Tab44 allows for defining the period for which requested items will be kept on the hold shelf. A differentiation may be made depending on whether the hold is triggered by the return of the item, or triggered by the Patron Request utility "Print Letter - Hold Request Filled". Many libraries use the utility for closed stack management and want a shorter hold period for items in closed stacks than for other items. Two date options can therefore be defined:

Date 1- **Return**: hold request status is changed to "S" through the return screen  
Date 2- **Utility**: hold request status is changed to "S" through requests/print letter hold request filled

Structure of the table:

| Col. 1 | Sub-library                          |
| Col. 2 | Item Status                           |
| Col. 3 | Patron status (use # for wild card)   |
| Col. 4 | Date operator -1  
|       | (D=day, M=month, W=week)            |
|       | Date 1 - Item is put on hold using the Return functionality |
| Col. 5 | Date parameter -1  
|       | Number of days/weeks/months to add   |
| Col. 6 | Date operator - 2  
|       | (D=day, M=month, W=week)            |
|       | Date 2 - Item is put on hold using the Print letter-Hold request filled functionality |
| Col. 7 | Date parameter -2                  |
Number of days/weeks/months to add

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>11</td>
<td>#</td>
<td>#</td>
<td>D</td>
<td>004</td>
<td>D</td>
<td>004</td>
</tr>
<tr>
<td>21</td>
<td>#</td>
<td>#</td>
<td>D</td>
<td>006</td>
<td>D</td>
<td>006</td>
</tr>
<tr>
<td>31</td>
<td>#</td>
<td>#</td>
<td>D</td>
<td>007</td>
<td>D</td>
<td>007</td>
</tr>
<tr>
<td>41</td>
<td>#</td>
<td>#</td>
<td>D</td>
<td>007</td>
<td>D</td>
<td>007</td>
</tr>
<tr>
<td>12</td>
<td>#</td>
<td>#</td>
<td>D</td>
<td>008</td>
<td>D</td>
<td>008</td>
</tr>
<tr>
<td>22</td>
<td>#</td>
<td>#</td>
<td>D</td>
<td>010</td>
<td>D</td>
<td>010</td>
</tr>
<tr>
<td>32</td>
<td>#</td>
<td>#</td>
<td>W</td>
<td>002</td>
<td>W</td>
<td>002</td>
</tr>
<tr>
<td>42</td>
<td>#</td>
<td>#</td>
<td>W</td>
<td>002</td>
<td>W</td>
<td>002</td>
</tr>
</tbody>
</table>

... 

**tab45.<lng>**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** Order transaction types for the order log

The *order log*, is part of the on-line Acquisitions. It is dependent on definitions defined in tab45. An entry is made in the log when an order is created, when the order status or item status is created or changed, when the next claim date changes, or when the user manually adds his own log notes.

Definitions in tab45 sets whether the transaction creates an order log record or not, and defines the description for the action (which is written in the order log text field).

**Note:** It is recommended to set Col.3 (Order Log can/cannot be entered manually) to Y only for the following transactions:

- 00 - General
- 95 - Subscription history note
- 98 - Reply from vendor (no claim)
- 99 - Note to vendor

If Y is set for other transaction types, the log record is created, the text that is entered is written on the Oracle table, but it does not display correctly in the GUI Order Log window.

**Structure of the table:**

- col. 1 Transaction number
- col. 2 Y/N
  - Y=activate logger
  - N=do not activate
- col. 3 Order Log
  - Y = Order Log can be entered manually
  - N = Order log cannot be entered manually
- col. 4 ALPHA plus function description
Example of the table:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>LGeneral note</td>
</tr>
<tr>
<td>01</td>
<td>Y</td>
<td>N</td>
<td>LOrder status</td>
</tr>
<tr>
<td>02</td>
<td>Y</td>
<td>N</td>
<td>LNext claim date</td>
</tr>
<tr>
<td>03</td>
<td>Y</td>
<td>N</td>
<td>LOrder created</td>
</tr>
<tr>
<td>04</td>
<td>Y</td>
<td>N</td>
<td>LItem(s) created</td>
</tr>
<tr>
<td>05</td>
<td>Y</td>
<td>N</td>
<td>LItem process status</td>
</tr>
<tr>
<td>06</td>
<td>Y</td>
<td>N</td>
<td>LEstimated price</td>
</tr>
<tr>
<td>08</td>
<td>Y</td>
<td>N</td>
<td>LArrival</td>
</tr>
<tr>
<td>09</td>
<td>Y</td>
<td>N</td>
<td>LBudget</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
<td>N</td>
<td>LInvoice</td>
</tr>
<tr>
<td>11</td>
<td>Y</td>
<td>N</td>
<td>LNo budget</td>
</tr>
</tbody>
</table>

**tab46.<lng>**

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** ILL request log transaction types

Tab46 defines whether
- a transaction creates an ILL request log record,
- the transaction is system generated, or can be entered manually.

The ILL request log is part of the on-line ILL module. It depends on definitions defined in tab46. For example, an entry can be made
- when an ILL Outgoing request (Z40) is created,
- when a supplier request is created (Z41),
- when Incoming ILL request is created (Z46)
- or when the user manually adds his own log notes.

**Note:** It is recommended to set Col. 3 (Request Log can/cannot be entered manually) to Y only for the following three transactions:
- 00 - General
- 20 - Message to Supplier
- 21 - Supplier Response

If Y is set for other transaction types, the log record is created, the text that is entered is written on the Oracle table, but it does not display correctly in the GUI Request Log window.

**Structure of the table:**

<table>
<thead>
<tr>
<th>col. 1</th>
<th>Transaction number</th>
</tr>
</thead>
</table>
col. 2  Active/non active log
- Y=activate logger
- N=do not activate

col. 3  Manually log
- Y = ILL Log can be entered manually
- N = ILL log can not be entered manually

col. 4  ALPHA +function description

Example of the table:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>00</td>
<td>Y</td>
<td>Y</td>
<td>LGeneral</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>Y</td>
<td>N</td>
<td>LSsupplier Request Status Changed</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>Y</td>
<td>N</td>
<td>LILL request created</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>Y</td>
<td>N</td>
<td>LSsupplier request created</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Y</td>
<td>N</td>
<td>LVouchers allocation</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>Y</td>
<td>N</td>
<td>LVouchers de-allocation</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>Y</td>
<td>N</td>
<td>LLetter was sent to the patron</td>
<td></td>
</tr>
</tbody>
</table>

**tab47.<lng>**

*Location of the table: tab directory of the ADM library*

*Purpose of the table: ILL supplier material type and copyright agreement*

This table allows for the coded definition of materials requested by the library. The coded definitions can be supplier specific. In addition, definitions regarding copyright agreement are also defined.

Structure of the table:

- **col. 1**  Supplier code – use # as a wild card
- **col. 2**  Copyright agreement needed – Y/N
- **col. 3**  Block request – Y/N
  - The request will be blocked if there is no copyright agreement.
- **col. 4**  ILL material type (up to 16 characters)
- **col. 5**  ILL material description

**Note the media type L-NORETURN.**

It is treated by the system in the same way as any other loan type (L-) except that the expected return date will be always set to 31-Dec-2099.

This allows for the creation of a Loan (Z36) and Item (Z30) for material that does not need to be returned to the ILL supplier. In those cases where it is important for the library to create an item and a loan record, this option will be useful.

Example of the table:
### tab48.<lng>

**Location of the table:** tab directory of the ADM library

**Purpose of the table:** General invoice payment status

This table allows for a more flexible way of defining the Z77-P-STATUS in relation to the invoice approval process, checks etc.

1. **Payment status**
   It defines all statuses for Z77-P-STATUS. It also defines the default payment status when the general invoice is created.

2. **Freeze on invoice**
   Defines in which statuses the invoice is frozen. This means that all operations except change in status are not allowed.

3. **Permissions**
   The table also defines which user_function routine should be used over and above the standard checks for different statuses. To accommodate this, the following changes have been made to the table user_function.<lng>.

**ACQ/INVOICE-HEAD-P-S-R**
Update "Ready to be paid" general invoice and/or its line invoices.

**ACQ/INVOICE-HEAD-P-S-Y**
Update "Payment authorization given" general invoice and/or its line invoices.

**ACQ/INVOICE-HEAD-P-S-P**
Update "Paid" general invoice and/or its line invoices.

This replaces the old line:
**ACQ/INVOICE-HEAD-UNPAID**
Update "PAID" general invoice and/or its line invoices.

**Structure of the table:**
- Col. 1 Invoice payment status.

Note that a line with status "P" (Paid) **MUST** exist in this table.
All other statuses can be library defined.

Col. 2 Default
Defines if the status of Col.1 will be the default value for a new general invoice
- Y = Default
- N = No default
When creating a new invoice, the system opens the invoice with the payment status as blanks. Refresh/Update replaces it with the assigned default. For a single line item invoice, the default value is taken automatically. There is no Z77-P-STATUS in the single line invoice form window.

**Note:** Only one line in this table should have the value "Y";

Col. 3 Freeze invoice
Freeze all related invoice records for this payment status. Records cannot be updated.
- Y = Freeze
- N = Do not freeze

Col. 4 user_function authorization to use for this status, in addition to the standard checks. If left blank, no additional check is made, and authorization for ACQ Update Invoice Payment Status suffices. The code in this line does not have to be the same as the in column 1. It is used only to identify the user authorization function line in the user_function table. Therefore, multiple invoice payment statuses can use the same authorization.

Col. 5 Payment status text

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Not ready to be paid</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>N</td>
<td>N</td>
<td>INVOICE-HEAD-P-S-R</td>
<td>Ready to be paid</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>INVOICE-HEAD-P-S-Y</td>
<td>Payment authorization given</td>
</tr>
<tr>
<td>P</td>
<td>N</td>
<td>Y</td>
<td>INVOICE-HEAD-P-S-P</td>
<td>Paid</td>
</tr>
<tr>
<td>U</td>
<td>N</td>
<td>N</td>
<td>INVOICE-HEAD-P-S-Y</td>
<td>User invoice</td>
</tr>
</tbody>
</table>

**tab100**

Location of the table: tab directory of the library

Purpose of the table: Switch settings for different aspects of the system

A new configuration table, tab100, replaces tab10.

- Tab100 will provide a single method for functional configuration.
• It is a parameter table like tab10, but is build in a textual way, i.e. “var=value”. This is similar to the GUI ini files. The maximum possible length of its variables is 99 characters.
• It has an internal mechanism for assignment of default values and validation of values. The automatically-built header file specifies the default value and the permitted values.
• It has hierarchy for library vs. system wide variables and server type vs. general variables.

It is possible to create tab100 or tab100_<server_type> under $alephe_tab or $data_tab of a library. For example, there can be an extension _pc or _www, in order to specify different values for pc_server and for www_server. If values are common, they can be entered in the tab100 that does not have extension.

When a program calls tab100 of a certain library/server type, the following is the reading order of tab100:

1) initialization with default values from TAB100.source (always)
2) reading $alephe_tab/tab100
3) reading $alephe_tab/tab100_<server_type>
4) reading $data_tab/tab100
5) reading $data_tab/tab100_<server_type>

At each stage (from 2 to 5), if tab100 exists, the values defined in it override the values from previous readings, e.g.
• if in $alephe_tab/tab100, ADDRESS-ZIP-STYLE=2,
• and in $data_tab/tab100 of USM50, ADDRESS-ZIP-STYLE=3,
• the final value of ADDRESS-ZIP-STYLE will be "3".

The names of the available variables, their types (text or number), their maximum length, their default value and permitted values are defined in ../alephm/source/copy/TAB100.source, and CANNOT be changed by the user.

Lines beginning with an asterisk ("*”) contain category names, which are not part of tab100; they are used only for display purposes when ./alephm/source/copy/TAB100 and the tab100 header are created automatically out of TAB100.source.

The names of the variables are in the main the same as in tab10 (but without the prefix "TAB10-“).

• A tab100 variable name can contain no more than 23 characters - (Column 1 of TAB100.source).
• Column 2, Variable Maximum Length, is a two-digit number from 01 to 99 for textual variables. If the variable is a number, column 2 is set to "N ".
  All numbers in tab100 can be up to 9 digits
• Column 3, Permitted in tab100 of library - Y/N, indicates whether the variable can be defined in tab100 located under the $data_tab directory of a library.
• Column 4, Permitted in tab100_<server_type> - Y/N, indicates whether the variable can be defined in tab100_<server_type> (e.g. tab100_pc) located under $alephe_tab, or tab100_<server_type> located under the
$data_tab directory of a library. This column has been set to "N" for all variables transferred from tab10.

- Column 5, Default Value, is maximum 20 characters long; it is a string or a number with which the variable is initialized whenever tab100 is loaded, BEFORE reading the settings of tab100 under $alephe_tab or $data_tab.
- Column 6, Permitted Values, is a space-delimited list of all possible values for the variable; for numeric variables, a range of numbers is provided, e.g. 10-90.

The values of tab100 can be displayed via UTIL G/4/100. This option is particularly useful for two reasons:

1) ALL tab100 variables (as defined in TAB100.source) are shown together with their values, regardless of whether they have been set in the specific tab100; unset variables are given a default value.
2) Full validation is performed on all tab100 values before display;
   The checks are:
   a) the tab100 line contains a "=" sign (e.g. CREATE-Z30H=Y); a line without a "=" sign is ignored.
   b) the tab100 variable is allowed for the specific tab100 that is being loaded e.g. variables which are not allowed for tab100 of $data_tab must NOT be set there, only in tab100 of $alephe_tab.
   c) the value of a variable does not exceed its maximum length.
   d) the value of a variable receives one of the permitted values only (if defined).
   e) numeric variables indeed receive a numeric value, which is also in the permitted range (if defined).

Whenever an invalid value is encountered, an error message is displayed and the value is replaced with the predefined default value, e.g.

Note that an asterisk * identifies lines that have limited special use.

**ACQUISITION**

**CHECK-INVOICE-CURRENCY**
Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = currency on invoice must match one of the vendor currencies. At least one currency must be entered in the vendor record.

**CHECK-ORDER-BUDGET**
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.
Y = for "P" (purchase) orders, does not allow 0.00 in the estimated price of order, when the order is sent, and requires a budget encumbrance. Performs budget checks (validity and balance) and changes the order status to DNB if errors are detected.
N = "P" order is allowed 0.00 in estimated price, and budget assignment is optional. If a budget is assigned, when the order is sent -- budget checks are performed, errors are reported, but the order status is not changed to DNB, and the order is sent. If no budget was assigned to the order, an invoice line item can be registered without a budget.

CHECK-ORDER-ISBN-ISSN
Values: Y N  Default: N
Type: Text; Max Length: 01
Y = ISBN/ISSN entered on the Acquisition order is checked for validity

CREATE-ITM-FORM-ORDER-M
Values: Y N  Default: Y
Type: Text; Max Length: 01
Y = checkbox for automatic item creation (on tab 3 of order form) is set to "checked".

OVER-EXP-INCLUDE-ENC
Values: Y N  Default: Y
Type: Text; Max Length: 01
Y = Max over expenditure of a budget is checked in relation to the encumbrance. The system checks the max over expenditure against the free balance. The free balance is the total allocation - (paid + unpaid invoices + encumbrances).
N = Max over expenditure of a budget is not checked in relation to the encumbrance but in relation to the expenditure. The system checks the max over expenditure against the actual balance of the budget. The actual balance is the total allocation - paid + unpaid invoices).

TWO-LEVEL-VENDOR
Values: Y N  Default: N
Type: Text; Max Length: 01
Y = The system uses two levels of vendor.
N = Only one level of vendor is managed.

USE-OBJECT-CODE
Values: Y N  Default: N
Type: Text; Max Length: 01
Y = When registering the Invoice, there is no check against the budget's object code and filling the Invoice's object code is not mandatory. Object code may be used for reporting only.
Y = Object code must be assigned to the Invoice and must match the assigned budget's object codes.

**USE-ORDER-UNIT**
Values: Y N  Default: N  
Type: Text; Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

This switch determines in which method the system works: Ordering units or sublibraries. It concerns the Budget's (Z76) order unit, Vendor's (Z70) order unit, Acq order's (Z68) order unit and user-passwords (Z66).

N = each sublibrary is an order unit.  
Y = ordering units are defined in tab_sub_library.<lng>.lng using sublibrary type 5.

**VENDOR-SHARING**
Values: 0 1  Default: 0  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

0 = for sites that do not have multi-ADM, or are not sharing a single vendor table in a multi-ADM environment. This is the default if this parameter is not defined.  
1 = for sites that are sharing a single vendor table in a multi-ADM environment. The Parameter must be "Y" in order to filter the vendor list to display only vendors that have either a relevant sublevel vendor record, or a relevant sublibrary/order unit defined.

**CATALOG**

**CREATE-852-HOL**
Values: Y N  Default: N  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

* Applicable only to the HOL library (USM60).  
Y = automatic generation of 852 subfields from call number fields in the BIB record (099, 098, 090, 092, 096, 050, 055, 060, 070, 082, 086)

**CREATE-Z00H**
Values: Y N  Default: N  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

Y = transfer a deleted BIB record to a history file. This is for statistical purposes only, and does not imply that there is capability to restore.

**CREATE-Z00R**
Values: Y N  Default: N  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.
Y = create a Z00R record for each Z00 record. Suitable for BIB, HOL and AUT libraries, but not for an ADM library.

CREATE-Z106
Values: Y N  Default: N
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

This switch determines whether a Z106 record is created automatically each time a cataloging record is created or updated.
N = record updates will not automatically generate Z106 records. In this case, the Z106 records can be created by running the Create/Update Z106 Table for "CAT" Field (p-manage-19) service available from the Catalog Maintenance Procedures option of the Services menu in the Cataloging module.
Y = each time a record is created or updated a Z106 record will be created.

DOC-BLANK-CHAR
Values: ^  Default: ^
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Defines what sign is used to denote a blank in MARC21 fixed fields. This should not be confused with the fill character |.

FORCE-USE-Z07
Values: Y N  Default: N
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Y = A Z07 record will be created in the library (e.g. ADM or HOL) although the document being updated does NOT belong to it (e.g. a Z07 will be created in an ADM library when a record in a BIB library to which it is linked, is updated).

MARC-TYPE
Values: 1 2 3 4  Default: 1
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Defines that type of MARC record
(1=USMARC, 2=UNIMARC, 3=DANMARC, 4=MAB)

OWN-FILTER
Values: Y N  Default: N
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Y = The display filter based on tab_own is activated (only for HOL or BIB)
N = The filter is not active. (Default)
**RLIN-INPUT-DIR**
Values: Default: rlin_input_dir  
Type: Text; Max Length: 99  
atab100 of library: Yes; tab100_<server_type>: No.

**RLIN-OUTPUT-DIR**
Values: Default: rlin_output_dir  
Type: Text; Max Length: 99  
atab100 of library: Yes; tab100_<server_type>: No.

**USE-ACC-TEXT**
Values: Y N  Default: N  
Type: Text; Max Length: 01  
atab100 of library: Yes; tab100_<server_type>: No.

Y = using CTRL+F3/F4 in cataloging the system copies the content of the chosen GEN heading into the BIB.  
N = using CTRL+F3/F4 in cataloging the system takes the preferred term (MARC = 1XX; MAB = TMP01) from the AUT record and copies it to the BIB.  
Default = N; this switch should only be defined for AUT libraries

**Z01-TAG-SENSITIVE**
Values: Y N  Default: N  
Type: Text; Max Length: 01  
atab100 of library: No; tab100_<server_type>: No.

**CIRCULATION**

**AVAILABILITY-ROUTINE**
Values: 1 2  Default: 1  
Type: Text; Max Length: 01  
atab100 of library: Yes; tab100_<server_type>: No.

* 1 = item is non-available if it is on loan, or on hold, or has request (Z37-SOURCE is "C" or "O"). This is the default if the switch is left blank.  
  2 = (site specific) item is non-available if it is on loan, or on hold, or has request (Z37-SOURCE "O" only).

**BARCODE-DELETE-SPACES**
Values: Y N  Default: Y  
Type: Text; Max Length: 01  
atab100 of library: Yes; tab100_<server_type>: No.

* Y = deletes spaces when barcode data is transferred to the system. This refers to the item barcode only.

**BOR-EXPIRY-DUE-DATE**
Values: Y N  Default: N  
Type: Text; Max Length: 01  
atab100 of library: Yes; tab100_<server_type>: No.
Y = sets the due date for loan and renew to the local patron expiration date if the local patron expiration date is earlier than the computed due date.

**BOR-STATUS-FOR-RENEWAL**
Values: L C  Default: L
Type: Text; Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

L = when the item is renewed, the patron status at the time of the original loan is used. This is the default if the switch is left blank.
C = when the item is renewed, the patron's current status (at the time of renewal) is used, and the Z36-BOR-STATUS is updated accordingly.

**CATALOGER-IP**
Values: Y N  Default: N
Type: Text; Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

This switch determines whether to display the Cataloger's staff name or the Cataloger's IP in GUI fields where the Cataloger's identity appears.
Y = Z36-LOAN-CATALOGER-NAME, Z36-RETURN-CATALOGER-NAME and Z36-RENEW-CATALOGER-NAME will display the cataloger IP address.
N = Z36-LOAN-CATALOGER-NAME, Z36-RETURN-CATALOGER-NAME and Z36-RENEW-CATALOGER-NAME will display the cataloger name.

**CHANGE-DUE-HOUR**
Values: Y N  Default: N
Type: Text; Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Y = for due-hour for loan, renew and change-date, use tab17 OPEN HOURS to re-calculate the due date, so that the due hour will always fall within library open hours
N = for due-hour for loan, renew and change-date, use tab16 parameters, ignore tab17 hours (e.g. if tab16 is set to 24:00, the due hour will be 24:00, even if closing time is 19:30)

**CHECK-BARCODE**
Values: Y N  Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

* N = disables (the default) check for barcode on the item record, thereby allowing item records without barcodes. If library opts to disable barcode check for items, the z30_id1 entry in the file_list must be adjusted accordingly, since it assumes that the barcode is an index, and will not allow a non-unique index key ("blank" is an index key).

**CHECK-BOR-DEFAULTS**
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = when updating the Z305 (local patron record) in the GUI CIRC client, the values set in the record are checked against the values set in tab31 for the patron status. If there is a discrepancy, an overridable warning displays.

CHECK-UNIQUE-NAME-BIRTH
Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = in a patron Z303, record the combination of name and birth date must be unique. Umlaut letters are considered to be identical to their non-umlaut value for this purpose. The non-umlaut value depends on the translation of the umlaut in the Unicode table.

CREATE-Z30H
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = create Z30H (ITEM HISTORY) record when item is updated or deleted, if /tab/tab_item_history.lng is set for creating history record. There can be multiple Z30H records for a single item; a Z30H record can be re-instated as current record.

CREATE-Z36H
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = transfer completed loan records to a history file

CREATE-Z37H
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = transfer completed hold records to a history file

DEFAULT-BOR-ID
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = create a default field value, using last-bor-id counter from z52 (UTIL G/2).

DEFAULT-BOR-ID-1
Values: Y N Default: Y
Type: Text; Max Length: 01
Y = create a default field value, using last-bor-id-1 counter from z52 (UTIL G/2)

DEFAULT-BOR-VERIFY-1
Values:Y N Default: Y
Y = create a default field value, using last-bor-verify-1 counter from z52 (UTIL G/2)

DEFAULT-BOR-VERIFY-2
Values:Y N Default: Y
Y = create a default field value, using last-bor-verify-2 counter from z52 (UTIL G/2)

ENCRYPT-Z308
Values:Y N Default: Y
Y = indicates that the patron identifying numbers are encrypted

HOLD-REQ-PROCESS-STATUS
Values:Y N Default: N
Y = a hold request includes the process status of the copy chosen for hold, and only copies with same process status will satisfy the request.

HOLD-REQUEST-COLLECTION
Values:Y N Default: Y
Y = a hold request includes the collection of the copy chosen for hold, and only copies with the same collection will satisfy the request.

HOLD-REQUEST-ITM-STATUS
Values:Y N Default: Y
Y = a hold request includes the item status of the copy chosen for hold, and only copies with same item status will satisfy the request.

ILL-ITEM-SUB-LIBRARY
Values:1 2 Default: 1
1 = (default). Always use patron home library if populated, otherwise use pickup location
2 = use pickup location if its a valid "item" sub library, otherwise use patron home library

**LOST-LOAN-CREDIT-METHOD**
Values: 1 2 Default: 1
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

1 = when a lost loan is returned (i.e. an item which has a Z36 record with status "L"), all associated cash transactions except "waived" are credited. This means "C"losed, "T"ransferred or "O"pen transactions are credited.
2 = when a lost loan is returned, all "T"ransferred or "O"pen transactions are credited. Closed transactions are not credited automatically!

**OFFLINE-CASH-TRANS**
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = Offline Circulation transactions will record cash transactions.
N = Offline Circulation transactions will not record cash transactions.

**OVERDUE-LETTER-NO**
Values: 0 1 2 3 4 Default: 0
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Defines "overdue" for tab_check_circ 1_d block. A loan is overdue if the due date has passed and "n" or more notices have been sent.

**OVERDUE-LETTER-STYLE**
Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

N = standard overdue notice, without special text based on letter number
Y = overdue notice text is sensitive to letter number

**OVERDUE-RECALL-RATIO**
Values: Y N B Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

This switch defines whether lines 0050-0055 in tab18.<lng> (fines relating to recalled items) are to be treated as a ratio or as an actual amount
Y or blank = lines are ratio
N = lines are actual amount
B = Both - lines are actual amount but they are added to the tab16 amount

**PHOTO-REQUEST-STYLE**
Values: H - Default: -
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

H = after the photocopy request slip is printed the request is deleted. The system does not check for availability of the item.
blank = after the photocopy request is printed the system sets the status of the request to 'A' and the print status to 'P'.

**PHOTO-SLIP-STYLE**
Values: 1 2 3 Default: 1
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

For printing photocopy request slips (cir-22). If a suitable item is available, a pickup slip is printed. Otherwise, a wait letter is printed. This switch relates to the list of items that are included in the print slip.
1 = print all "like" item records in the slip
2 = print only available "like" item records in the slip
3 = print the first of the available "like" items in the slip

**RECALL-METHOD**
Values: 1 2 3 Default: 1
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

"1": Let renewals update the z36_due_date and let recalls update it, if the recall-due-date is earlier. Use the z36_original_due_date as the "due-date-before-recall". (Let renewals update it.)
"2": Let renewals update the z36_due_date. Do not actually update the z36_due_date to the recall-due-date. But, if the latter is earlier than the z36_due_date, use the latter as the "effective due date". Leave the z36_original_due_date as the actual, original due date. (Do not let renewals update it).
"3": Same as "2", except always use the recall-due-date as the effective due-date - even if it's later

**REFUND-RATE**
Values: Default: 100 100 0
Type: Text; Max Length: 50
tab100 of library: Yes; tab100_<server_type>: No.

When an item that has been declared lost is returned by the patron, the charges that were made when the item was declared lost can be refunded. This variable can be used to set the percentage of each charge that will be refunded.
Possible values are three groups of digits, each group up to three digits long, up to
value "100".

- The first group sets the percentage that will be refunded for handling costs
  (transaction number 40),
- the second group the replacement cost (transaction number 41)
- and the third group the notice cost (transaction number 42).

A value of 0 means that no refund will be made, and a value of 100 means that full
refund will be made. When one of the rates is set to zero, the flag ZERO-FINE-
HANDLING is consulted whether to generate zero-sum credit transaction.

**RENEW-DURING-LOAN**
Values: Y N  Default: Y
Type: Text; Max Length: 01

tab100 of library: Yes; tab100_<server_type>: No.

Y = when a loan is performed on a loaned item, if the patron ID for the loan is the
same as the patron ID in the loan record, the loan is treated as a renewal, and switch
RETURN-DURING-LOAN, is ignored. When set to "Y", the system continues to
work in the same manner that it worked from versions 12.2 through 12.4.
N = when a loan is performed on a loaned item, only RETURN-DURING-LOAN is
in effect.

**RETURN-DURING-LOAN**
Values: 0 1 2  Default: 0
Type: Text; Max Length: 01

tab100 of library: Yes; tab100_<server_type>: No.

0 = when an item is being loaned, and the system detects that the item is currently on
loan, a message displays and the item must first be returned
1 = when an item is being loaned, and the system detects that the item is currently on
loan, the item is automatically returned. No cash for overdue and no check for hold
requests.
2 = when an item is being loaned, and the system detects that the item is currently on
loan, the item is automatically returned and a cash transaction is created if the item
was returned late.

**SHOW-USR-VERIFICATION**
Values: Y N  Default: N
Type: Text; Max Length: 01

tab100 of library: Yes; tab100_<server_type>: No.

Y = display patron verification fields when updating the Z303 (patron) in the GUI
CIRC and ADMIN clients.
N = mask out the view of patron verification fields when updating the Z303 (patron)
in the GUI CIRC and ADMIN clients.

**SUB-LIBRARY-DIVISION**
Values: Y N  Default: N
Type: Text; Max Length: 01

tab100 of library: Yes; tab100_<server_type>: No.
Y = a hold/photo request includes the sublibrary of the copy chosen for hold/photo, and only copies with the same sublibrary will satisfy the request.

**TIMEZONE-ADJ**

Values: 0 + - Default: 0
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

This switch is used to set the "current time" of a library in relation to the server time, where + adds one hour and - subtracts one hour. This adjustment affects circulation-related activities in an ADM library.

**UPDATE-RECALL-DATE**

Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

This switch determines if, when an item is loaned, and there is a matching outstanding hold request, the loaned item should immediately be considered "recalled", or whether a loaned item becomes "recalled" only through the batch recall service (cir-13). Y = item loan can trigger "recall" (Z36-recall-date and z36-recall-due-date are automatically set at time of loan if there is a matching "O" request) N = recall of a loaned item (update of Z36-recall-date and z36-recall-due-date) is triggered only by p-cir-13.

**USER-ADDR-PERMISSION**

Values: 0 1 2 3 Default: 0
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

0 = No Check - Can update user Addresses
1 = Can update its Addresses only if it has any Z305 records.
2 = Can update its Addresses only if it has Z305 records other than ALEPH.
3 = The staff user can update Addresses only if he is connected to a library that includes the patron's home library. If the USER-ADDR-PERMISSION flag is set to 3, the home library is a mandatory field when creating a new global patron.

**USER-IDS-PERMISSION**

Values: 0 1 2 3 4 Default: 0
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

0 = No Check - Can view/update user ID’s
1 = Can view/update user ID’s only if the user has any Z305 records.
2 = Can view/update user ID’s only if the user has Z305 records other than ALEPH.
3 = The staff user can view and update the patron's IDs only if he is connected to a library that includes the patron's home library. If the USER-IDS-PERMISSION flag is set to 3, the home library is a mandatory field when creating a new global patron.
4 = Can view/update user IDS only if there is a Z305 record for the user that matches the operator's sublibrary/library permission . This option is relevant for installations
that share a common Z305 table, permission is allowed because the user has
registered at the library.

USER-PERMISSION
Values: 0 3  Default: 0
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

0 = update of Z303 is based on standard permissions (i.e. operator has been assigned
privileges for update Z303
3 = update of Z303 is allowed only if the Z303-HOME-LIBRARY matches the
connected library, or is included in the list of sublibraries of the connected library. If
set to 3, Z303-HOME-LIBRARY is a mandatory field when creating a new patron.

USER-SHARING
Values: Y N  Default: N
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Y = The library participates in a shared user environment; patrons created by this
library are NOT identified in the user record (Z303_USER_LIBRARY is blank), and
will display in the Patron List when a staff operator is connected to ANY library that
is defined as "Y"; patrons that are identified as "belonging" to a library
(Z303_USER_LIBRARY=an ADM library code) are NOT included in the Patron
List.
N = The library does not participate in a shared user environment; patrons created by
this library are identified by the ADM library code in the patron record
(Z303_USER_LIBRARY), and will display in the Patron List only when a staff
operator is connected to the ADM library.

ZERO-FINE-HANDLING
Values: Y N  Default: N
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

Y = register a cash transaction for a late return even though the amount is 0.00.

GENERAL

AUT-TYPE
Values: S M T  Default:
Type: Text;  Max Length: 01
tab100 of library: Yes;  tab100_<server_type>: No.

S = regular AUT database
M = multilingual AUT database
T = Thesaurus

DOC-TYPE-ADM
Values: Y N  Default: N
Y = indicates that the library is of type "ADM"

DOC-TYPE-AUT
Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = indicates that the library is of type "AUT"

DOC-TYPE-BIB
Values: Y N Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = indicates that the library is of type "BIB"

DOC-TYPE-HOL
Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = indicates that the library is of type "HOL"

DOC-TYPE-ILL
Values: Y N Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = indicates that the library is of type "ILL"

ITEMS

INVENTORY-NUMBER
Values: Z52 Z311 Z313 Default: Z52
Type: Text; Max Length: 04
tab100 of library: Yes; tab100_<server_type>: No.

Z52 - Z30-INVENTORY-NUMBER can have any value; it can be assigned by using Z52 sequences, defined by UTIL G/2. To do so, a question mark (?) followed by Z52-sequence> must be entered in the GUI field "Inventory Number" (e.g. ?inventory-number).

Z311 - Z30-INVENTORY-NUMBER is always assigned by using the Oracle table Z311 and the ALEPH table /usm50/tab/tab_z311.

Z313 - Z30-INVENTORY-NUMBER is always assigned by locating a relevant Z313 record; A mere question mark (?), or a question mark followed by <Series> MUST be entered in the GUI field "Inventory Number" for the relevant Z313 record to be found and allocated to Z30-INVENTORY-NUMBER.
UPDATE-Z30-COPY
Values: Y N  Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = Z16-COPY-SEQUENCE is copied to Z30-COPY when an item record (Z30) is opened from a serials schedule record (Z17)

X852-ITEM-OVERRIDE
Values: Y N  Default: Y
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = item location subfields (sublibrary, collection, call no.) are updated from the 852 field of the linked HOL record. The line
setenv correct_852_subfields
in aleph_start controls which 852 subfields update the Z30 (item) location fields. "Y" must be set in both the xxx50 and xxx60 libraries.

Z30-PRICE-FROM-ORDER
Values: Y N  Default: N
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

Y = the item price field is controlled by and automatically updated from the acquisition price (estimated and invoiced).

OPAC

ACC-COUNT-STYLE
Values: S R  Default: S
Type: Text; Max Length: 01
tab100 of library: Yes; tab100_<server_type>: No.

This switch determines how the counter of number of related BIB records displays in a browse list, when the heading is a "see" reference.
R = counts both the BIB records linked to this heading and also the BIB records linked to the heading pointed to.
Normally, there will be no records under the "see" heading, since it refers to another heading. However, this can occur in libraries that do not automatically update BIB records because of AUTHority record "see" references (these libraries use UPD=N in the AUTHority record).
S = count only the BIB records that are linked to this "see" heading. In libraries that do not use UPD=N, and do automatic correction from cross references, this will normally be zero.

AUTO-TRUNCATE-Z01-FIND
Values: Y N  Default: N
Y = when a FIND search is performed on an access (browse) list, the search term is automatically set as truncated

**BASE-DEMO-LIB**

Values: 1 2  Default: 1  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

1 = the system's demo data base is USMXX  
2 = the system's demo data base is MABXX  
This parameter is used for tables synchronization report (UTIL H/1), and for UTIL G/2 values.

**ITMG-DISPLAY**

Values: Y N  Default: N  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

In Web OPAC, for ITM1, ITM4 and ITM5:
Y = display the sublibrary in edit_doc_999 only if there are Z30 (item) records for the library.  
N = display the sublibrary in edit_doc_999 if there are either Z30 (item) or HOLdings record for the library

**UNION-LIBRARY**

Values: 0 1 2  Default: 0  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

0 = BIB database does not use "union" features  
1 = Union View Catalog: BIB database uses Z120 to display equivalent BIB records as one record. The single record display is linked to the multiple related ADM records for items/holdings display  
2 = Union Catalog: BIB database can use Z120 to display equivalent BIB records as one record. Holdings information is imbedded in the BIB record, and there is no item display.

**X852-HOLDING**

Values: Y N  Default: N  
Type: Text;  Max Length: 01  
tab100 of library: Yes;  tab100_<server_type>: No.

Y = whether to use a special program that presents a list of holdings taken from 852 tag. For union catalogs.

Example of the table in the BIB library

```
CENTRAL-LIBRARY=N
```
Example of the table in the AUT library

CREATE-852-HOL=T
MARC-TYPE=1
ADDRESS-ZIP-STYLE=A
AUT-TYPE=S
DOC-TYPE-BIB=N
DOC-TYPE-HOL=N
DOC-TYPE-ADM=N
DOC-TYPE-AUT=Y

Example of the table in the ADM library

HOLD-REQUEST-ITM-STATUS=Y
HOLD-REQUEST-COLLECTION=Y
CENTRAL-LIBRARY=N
CREATE-852-HOL=Y
CREATE-Z36H=Y
CREATE-Z37H=Y
CREATE-Z30H=Y
CHECK-INVOICE-CURRENCY=N
X852-ITEM-OVERRIDE=Y
HOLD-REQ-PROCESS-STATUS=N
RETURN-DURING-LOAN=0
CHECK-ORDER-BUDGET=Y

tag_text.dat

Location of the table: pc_tab/catalog directory of the library

Purpose of the table: Pre-defined text options for sub-fields

Structure of table:
  col. 1   tag
col. 2 indicators. Use # as wildcard.
col. 3 ALPHA - should always be L
col. 4 sub-field. Use # to indicate any sub-field.
col. 5 Value // Description

Note that if changes are made to this table, UTIL M/7 must be activated in order for the changes to be reflected in the on-line catalog.

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>655 ## L a Biographies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>655 ## L a Bird's eye views</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>655 ## L a Cartoons</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>655 ## L a Catechisms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LKR ## L a PAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LKR ## L a ITM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STA ## L a SUPPRESSED</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STA ## L a DELETED</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that to activate the option in the on-line cataloging enter F8 (or defined hot key for the list), and the beginning text of the element.

For example for the 655 tag:

- “b” + tab will present the user with a list of the entries beginning with B (there is more than one entry beginning with the letter ’b’).
- When entering Q, however, as there is only one entry with the letter Q (case sensitive), the text will automatically be placed in the field.

Note that the library can define any kind of text list for field fill in – as per the example of the STA field

**tagonnew.dat**

Location of the table: pc_tab/catalog directory of the library

Purpose of the table: Default fields for a new record

This table defines which are the default fields when “Create New Record” is activated in the on-line Cataloging Module.

Structure of the table:

Col. 1 ALPHA + field code

Example of the table:

```
! 1
```
unicode_to_filing_01

Location of the table: alephe/unicode directory

Purpose of the table: Unicode character conversion equivalencies for sorting

This table defines equivalencies for characters, for the purpose of filing (sorting). All characters naturally sort by their unicode value. In order to force a different sort order, an equivalency for sorting can be set. The equivalency can be up to 5 characters.

For example:
- in order to sort an accented lower case \textit{a} (e.g. Latin small letter a with acute - â) as a regular capital \textit{A}, set the equivalency of the accented character (00E1) to 0041 (the hexadecimal value of \textit{A}).

Another example:
- in order to sort \textit{ü} as \textit{ue}, set the equivalency of \textit{u-umlaut} (00FC) to \textit{U + E} (0055 + 0045).

Remember that some of the procedures in the library's tab\_filing table also change characters (e.g. compress, compress\_blank, expand\_num, etc.)

When a browse query is parsed, the data is translated using the unicode_to_filing table, not tab\_filing. Therefore, the two tables should be coordinated. In actual fact,
- "to\_blank" in tab\_filing is equivalent to setting a character to 0020 in unicode\_to\_filing
- "compress" in tab\_filing is equivalent to setting a character to 0000 in unicode\_to\_filing,

and it suffices to set their value in the unicode\_to\_filing table. However, for purposes of clarity, the following characters have been defined both in tab\_filing (procedures 11, 11, 12, 99), and in unicode\_to\_filing:
- compress (set to 0000): ";" "<>%
- to\_blank (set to 0020): ./--;{>[]?^\_\n
The library may have special filing requirements for non-textual information (e.g. UDC, DDC, call numbers), in which case a different unicode\_to\_filing table should be set, in which the value of special characters such as those listed in "compress" and "to\_blank" should be retained.

Also note that tab\_filing can include "to\_lower" which uses the unicode\_case table to transfer all upper case characters to lower

Note:
- All characters are expressed in hexadecimal values.
- Characters which are not listed in this table, and characters which are listed in col.1, but do not have a value in col.2, are dropped from the filing key. In other words, all filing characters must be defined in the filing table.
• The last column in the table is a note field, giving a description of the character.
• There can be multiple `unicode_to_filing_nn_source` tables (where `nn` is replaced by two digits). The relevant table is called up by the FILING-KEY-`nn` line in the tab_character_conversion_line table. The library's tab_filing table points to the FILING-KEY-`nn` line.

In this sample file, `unicode_to_filing_01`, there is character conversion for
• Basic Latin,
• Latin-1 Supplement + Latin Extended-A (0021 through 017F),
• Cyrillic,
• Greek,
• superscript,
• subscript,
• some other individual characters.

We have not provided character conversion for other characters and languages (e.g. Danish, Czech).

In general, special characters in these languages have been left with their "natural" value.

In the 0021 through 017F range,
• some special characters have been translated to 0000 (e.g. PERCENT SIGN),
• accented characters have been translated to the equivalent non-accented character,
• and a few characters have been translated to two characters (e.g. 00C6 (AE) to A + E).

Structure of the table:
Col. 1 Character as stored in the database, in its hexadecimal value
Col. 2 Equivalent – i.e. character to which character will be changed
Col. 3-6 Equivalents – i.e. 2nd – 5th characters in string to which characters will be changed
Col. 7 Description of the character. Must begin with #

Example of the table:

<table>
<thead>
<tr>
<th>Character</th>
<th>Equivalent</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0020 0020</td>
<td>#SPACE</td>
<td></td>
</tr>
<tr>
<td>0021 0000</td>
<td>#EXCLAMATION MARK</td>
<td></td>
</tr>
<tr>
<td>0022 0000</td>
<td>#QUOTATION MARK</td>
<td></td>
</tr>
<tr>
<td>0023 0023</td>
<td>#NUMBER SIGN</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C0 0041</td>
<td>#LATIN CAPITAL LETTER A WITH GRAVE</td>
<td></td>
</tr>
<tr>
<td>00C1 0041</td>
<td>#LATIN CAPITAL LETTER A WITH ACUTE</td>
<td></td>
</tr>
<tr>
<td>00C2 0041</td>
<td>#LATIN CAPITAL LETTER A WITH CIRCUMFLEX</td>
<td></td>
</tr>
<tr>
<td>00C3 0041</td>
<td>#LATIN CAPITAL LETTER A WITH TILDE</td>
<td></td>
</tr>
<tr>
<td>00C4 0041</td>
<td>#LATIN CAPITAL LETTER A WITH DIAERESIS</td>
<td></td>
</tr>
</tbody>
</table>
unicode_to_word_gen

Location of the table: alephe/unicode directory

Purpose of the table: List of Unicode character conversion equivalencies, to be used when creating records in the WORDS file

This table defines equivalencies for characters, for the purpose of creating words in the words file. It is used in conjunction with the library's tab_word_breaking table. All characters naturally retain their unicode value, and are stored in the system in UTF encoding. In order to translate one character into another character (e.g. translating an accented "e" to "e"), you can set an equivalency. The equivalency can be up to 5 characters.

For example,
- to translate a lower case "a" with acute accent to regular "a", set the equivalency of the accented character (00E1) to 0061 (the hexadecimal value of a).
- to set an umlauted "u" as "ue", set the equivalency of u-umlaut (00FC) to "u" + "e" (0075 + 0065).

Notes:
- All characters are expressed in hexamdecimal values.
- The last column in the table is a note field, giving a description of the character.
- The unicode_to_word_gen table is used both for building words entries and for translating search query input.
- The library's tab_word_breaking table can define different treatment for the same characters. In separate procedures specific characters can be set to compress or to be changed to blank. Characters dealt with in this manner should be left in their natural value, and not translated in this table.
- Note that the system automatically carries out triple posting for hyphens and apostrophes:
  1. as separate words;
  2. as is (with hyphen/apostrophe);
  3. with hyphen/apostrophe compressed. For example, twenty-five is indexed as:
     twentyfive
twenty
five
twenty-five
- Both the hyphen and the apostrophe must be left with their actual value (hyphen - 002D and apostrophe - 0027), and not translated in this table.
- The table does not have to include all characters. It suffices to register the exceptions.

Structure of the table:
- Col. 1 Character as stored in the database, in its hexadecimal value
Col. 2 Character to which character will be changed
Col. 3-6 2nd – 4th characters in the string, to which the character will be changed
Col. 7 Description of the character. Must begin with #

Example of the table:

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Col. 2</th>
<th>Col. 3</th>
<th>Col. 4</th>
<th>Col. 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0041</td>
<td>0061</td>
<td>#LATIN CAPITAL LETTER A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0042</td>
<td>0062</td>
<td>#LATIN CAPITAL LETTER B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0043</td>
<td>0063</td>
<td>#LATIN CAPITAL LETTER C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0044</td>
<td>0064</td>
<td>#LATIN CAPITAL LETTER D</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C0</td>
<td>0061</td>
<td>#LATIN CAPITAL LETTER A WITH GRAVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C1</td>
<td>0061</td>
<td>#LATIN CAPITAL LETTER A WITH ACUTE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C2</td>
<td>0061</td>
<td>#LATIN CAPITAL LETTER A WITH CIRCUMFLEX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C3</td>
<td>0061</td>
<td>#LATIN CAPITAL LETTER A WITH TILDE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00C4</td>
<td>0061</td>
<td>#LATIN CAPITAL LETTER A WITH DIAERESIS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**union_global_param**

Location of the table: alephe/tab directory

Purpose of the table: Union deduplication parameters

Structure of the table:

Col. 1 Library
Col. 2 T = for testing (UTIL/F/1/21)
        B = for batch jobs
Col. 3 Code
Col. 4 Program
Col. 5 Table

Example of the table:

<table>
<thead>
<tr>
<th>!</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>USM90</td>
<td>B</td>
<td>candidate_prog</td>
<td>union_candidate_cdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM90</td>
<td>B</td>
<td>match_prog</td>
<td>union_match_cdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM90</td>
<td>B</td>
<td>preferred_prog</td>
<td>union_preferred_cdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM90</td>
<td>B</td>
<td>normalize_prog</td>
<td>union_normalize_cdl</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USM90</td>
<td>T</td>
<td>candidate_prog</td>
<td>union_candidate_cdl</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**union_preferred**

Location of the table: tab directory of the library
Purpose of the table: Weighting table for determining which of two matching records is the preferred record

This table's functionality can be carried out by a table of any name. The name is not set in the programs, it is set in the third column of the tab_preferred table. It is the table to be used for weighing and giving points to a record in order to set which of two matching documents is the preferred document.

Setting the preferred document is an option for the merge_doc_adv_overlay program, used when merging records.

This program uses the tab_merge_adv_overlay table to decide which fields to retain and which fields to drop when merging records. Preferred record identification is an added parameter in the tab_merge_adv_overlay table.

Structure of the table:

Col. 1 Field tag. Document record field code. # can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (e.g., 245## for 2451, 2452, 24501).

Col. 2 Subfield(s) of the field (blank indicates the entire field).

OR

F followed by fixed field position (counted from base 00, taking the subfield code into account if there is a subfield code), followed by a hyphen (-) and the number of characters.

Col. 3 Operation code - Code of the action to carry out on the data. Available actions are EQUAL, NOT-EQUAL and PRESENT.

Col. 4 Value of the subfield content, or of the data in the section of the fixed field. Multiple values separated by a comma are treated as "or".

Col. 5 Points assigned. Number of points to assign when the line is "true". A negative number of points can be assigned by placing a minus in the first position, followed by 2 digits.

Example of the table:

```
!!!!!-!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!-!!!
LDR F05-01 EQUAL d -10
LDR F17-01 NOT-EQUAL 1,2,3,4,5,7,8,u,z 010
LDR F17-01 EQUAL 1 009
LDR F17-01 EQUAL 2 008
```

user_function

Location of the table: alephe/tab directory

Purpose of the table: User authorization definitions and codes
This is a list of the password authorizations that can be assigned to users, including all functions in the Web Services module (Lines which begin WWW-U, R, G, N, T). A single password can be defined for several libraries. The passwords, however, are defined in one library which must be set up as the ‘pw_library’ in aleph_start.

NOTE: This table should not be modified except for translation purposes. When translating, ONLY the NAMES, should be translated, NOT the codes.

Structure of the table:

Col. 1 Internal code of function. This should NOT be translated.
Col. 2 ALPHA
Col. 3 Name of function (can be translated)
Col. 4 Internal code of sub-function. This should NOT be translated.
Col. 5 ALPHA
Col. 6 Name of sub-function (can be translated)

Example of the table:

|    | 1       | 2   | 3       | 4       | 5   | 6
|----|---------|-----|---------|---------|-----|-----
|    | !!!!!!!! | !-!-! | !!!!!!!! | !-!-!!! | !    |     |
| GLOBAL | L All Functions | GLOBAL | L All sub-functions |
| OVERVIEW-TREE | L Overview Tree | VIEW-TREE | L Vie |
| w Overview tree |
| ACQ | L Acquisitions | GLOBAL | L All subfunctions |
| ACQ | L Acquisitions | INDEX-LIST | L Ord |
| er Search - Order index |
| ACQ | L Acquisitions | LIST | L Ord |
| er Search - New/Cancelled index |
| ACQ | L Acquisitions | ORDER-LIST | L Ord |
| ers - Display list for single record |
| ACQ | L Acquisitions | ORDER-VIEW | L Ord |
| er - View summary information |
| ACQ | L Acquisitions | ORDER-GET | L Ord |
| er - Display input form |

www_f_func

Location of the table: alephe/tab directory

Purpose of the table: Functional codes linked to HTML action commands

This table includes all functional codes that come from the HTML action command, and the correspondence program name and parameters (if needed). It is an internal table, which makes it easier to add programs for particular applications or projects.
Structure of the table:

| Col. 1 | Library code |
| Col. 2 | function code |
| Col. 3 | program name |
| Col. 4 | parameters |

Example of the table:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!...</td>
<td></td>
</tr>
</tbody>
</table>

> ##### FIND-A-0  www_f_find_a_0 
##### FIND-A  www_f_find_a
##### FIND-B-0  www_f_find_b_0 
##### FIND-B  www_f_find_b
##### FIND-B-LIST  www_f_find_b_list 
##### FIND-B-PERMUTE  www_f_find_b_permute

www_f_heading

Location of the table: alephe_root/error_<lng>

Purpose of the table: Messages to the patron in the WEB OPAC

This table lists the messages sent to the patron in the WEB interface.

Note the following:
Availability display in the "Due date" column - heading #350 ("On shelf") is sensitive to the Z30-ITEM-PROCESS-STATUS of the item. In order to take care of this, the following changes have been implemented:

- There is a file containing all headings for an item's circulation status. The file is Salephe_error_eng/circ_status_holding. Codes 8000 - 8012 were removed from Salephe_error_eng/global and put into this table. The holdings display in the web now uses codes 8000 - 8013 in circ_status_holding instead of codes 0350 - 0359 in www_f_heading
- A new heading was created - 8013. If Z30-ITEM-PROCESS-STATUS is blank, the heading #8012 (On Shelf) appears. If it is not, heading #8013 appears with the text taken from tab15.<lng>.

Structure of the table:

| Col. 1 | Message number |
| Col. 2 | ALPHA |
| Col. 3 | Text displayed to patron in WEB OPAC |

Example of the table:

| !!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!... |
0019 L Reenter your password.
0095 L Enter patron ID / barcode.
0100 L Unrecognized operation code. Type ENTER.
0105 L Error in borrower ID/sub-library.
0106 L Note is too long.
0107 L Note is empty.
0108 L Note cannot be added on to current notes.
0109 L Borrower doesn't have Items on Loan.
0110 L Your note has been delivered.
0120 L All items
0122 L not sorted.
0123 L sorted by :
0124 L Sort options :
0132 L Verification fields do not match or empty.
0133 L Verification field is empty.

**Note** that if the patron is allowed to update his/her address, the link for address update from the html page displays. If the patron is not allowed to update address, line 4052 from www_f_heading displays.

### www_f_sort_heading

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** Sort options for WEB OPAC brief formats

The brief display of bibliographic data in the Web OPAC allows for sorting of the different display elements as defined in www_tab_short.<lng>.

WWW_f_sort_heading defines the sort options, and their description. The description of the sort displays on this brief list of records. Note that the sort keys are set up in tab_sort.

The display includes the current sort -- Heading "Sorted by" (0123 in www_f_heading), and optional sorts -- Heading "Sort options" (0124 in www_f_heading).

**Structure of the table:**

- **Col. 1** Primary sort key code.
  - Must match the sort key as defined in the tab_sort table
- **Col. 2** A-Ascending
  - D-Descending
- **Col. 3** Secondary sort key code
  - Must match the sort key as defined in the tab_sort
- **Col. 4** A-Ascending
  - D-Descending
- **Col. 5** Text that displays after "Sorted by"
Example of the table:

| Col. 6 | Text that displays after "Sort options"
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>!1 2 3 4 5 6</td>
</tr>
<tr>
<td></td>
<td>!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!</td>
</tr>
<tr>
<td></td>
<td>02---A01---D Author/Year (descending) Author/Year(d)</td>
</tr>
<tr>
<td></td>
<td>02---A01---A Author/Year (ascending) Author/Year(a)</td>
</tr>
<tr>
<td></td>
<td>01---D02---A Year (descending)/Author Year(d)/Author</td>
</tr>
<tr>
<td></td>
<td>02---A03---A Author/Title Author/Title</td>
</tr>
<tr>
<td></td>
<td>03---A01---D Title/Year (descending) Title/Year(d)</td>
</tr>
<tr>
<td></td>
<td>03---A01---A Title/Year (ascending) Title/Year(a)</td>
</tr>
<tr>
<td></td>
<td>01---D03---A Year (descending)/Title Year(d)/Title</td>
</tr>
</tbody>
</table>

Note that the linked pairs of primary and secondary sort options, should match the sort definitions as defined in www_tab_short:

<table>
<thead>
<tr>
<th>www_f_sort_heading:</th>
</tr>
</thead>
<tbody>
<tr>
<td>02---A01---A Author, then Year ...</td>
</tr>
<tr>
<td>01---D02---A Year, then Author ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>www_tab_short:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 L Author 02---A01---A 00 00 0100 S 9 - 1#### 700##</td>
</tr>
<tr>
<td>3 L Year 01---D02---A 08 04 0000 S 008</td>
</tr>
</tbody>
</table>

**www_f_tab_short_999**

**Location of the table:** tab directory of the BIB library

**Purpose of the table:** Custom brief display format for Web OPAC

This table defines the short format options (not short table view) for the Web OPAC. The selection of the format is through the preferences (option-display/option-display-nobor) screen.

The first column defines the format number. The format number should be between 951-998. 950 is reserved for the full edit_doc_999.<lng> view.

Columns 2-27 define the field codes which will be displayed.

Note: Format 999 must be defined. The system will use it if a format that does not exist in the table is chosen.

<table>
<thead>
<tr>
<th>Col. 1</th>
<th>Format number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cols. 2-27</td>
</tr>
<tr>
<td></td>
<td>Field code</td>
</tr>
</tbody>
</table>

Example of the table:
www_r_heading

Location of the table: alephe_root/error_<lng>

Purpose of the table: Messages to the patron in the Course Reading Module

Structure of the table:
- Col. 1  Message number
- Col. 2  ALPHA
- Col. 3  Text displayed to patron in WEB OPAC.

0002 L Error in Library code! Type again!
0009 L You are not authorized to use this function.
0074 L Mandatory field is missing. Please check your data insertion.
0075 L You inserted "Fee Amount" without filling "Fee Type" field.
0076 L You must insert "Vendor/Provider Code" field.
0077 L You must insert "Vendor/Provider Code" field or "Fee Amount".
0100 L Unrecognized operation code. Type ENTER.
0143 L The requested record does not contain an external link or subfield "u" is empty.
0221 L Error in password or patron is not permitted/classified.
0309 L Error in barcode number.

www_server.conf

Location of the table: alephe_root directory

Purpose of the table: Default values for the Web

General note:
Note that there are some definitions that offer more than one "setenv" option. The non-active option does not have to be commented out – the system will always use the last line defined for that option.

Explanation of some of the elements related to path in the file:
setenv icon_path  
"&server_httpd/www"
It is possible to use an httpd daemon to load icon GIF files on to the ALEPH WWW directories. This will improve the efficiency of the main WWW server.
setenv security_path
"${ALEPH_MOUNT}/a${ALEPH_VERSION}_${ALEPH_COPY} /
/tmp;${TMPDIR};${FILE_TMPDIR}"
In order to be able to use the “save set” option in the WEB OPAC, the security path
must include $TMPDIR and $FILE_TMPDIR:
Only the files under the defined sub-file-systems can be delivered by the server.

Explanation of some of the variables:
setenv allow_create_new_profile  Y
If set to "Y", the system will create a new profile for the patron

setenv create_statistics          N
The z34 records can be used for statistical analysis of WEB server use. Overrides the
default that is set in aleph/proc/www_server; If set to Y the counter “last-z34-
sequence” should be defined in the z52 table (UTIL G/2).

set_default_z110_create_order    A
A variable that controls how documents are added into a set (z110).
The order can be "A" (ascending) or "D" (descending).
The default is "A".
This order is important in non-sorted sets as typically the last document numbers in a
system are the most the recently cataloged records.

setenv direction_support          Y
Text displayed in web pages of the WEB OPAC will be right-aligned and
right-oriented (for Hebrew and Arabic)
Note the following: Add placeholders with numbers greater by 50 than the original, to
web pages for which right-to-left support is required.
For instance, if the title (placeholder $0400 in ./alephe/www_f_eng/short-a-body) has
to be right-aligned and right-oriented, add a placeholder - $5400 - immediately
following the cell's class definition:
• Without right-alignment:
  <td class=td1 width="30%">$0400</td>
• With right-alignment:
  <td class=td1 $5400 width="30%">$0400</td>

setenv due_date_format            1
For details see explanation for the same variable as defined in pc_server_defaults

setenv hol_filter_op              BASE
setenv hol_sort_op                BASE
The variables hol_filter_op and hol_sort_op can determine the filtering and sorting
options of the holdings list on the holding information page.

setenv hold_pickup_from_all_items Y
For more information, refer to this variable under pc_server defaults.

setenv mark_match_words_prefix    "<span class=text3 id=highlight>">
setenv mark_match_words_suffix    "</span>"
Highlight words in full (999) format display (edit_doc_999 format) depending on the search request. The function compares each line with Z05-REQUEST. The two parameters mark_match_words_prefix and mark_match_words_suffix defines what is to be added before and after the matched word. In the example above, the word will be highlighted. In order, for example, to have the matched word appear in bold define as follows:

mark_match_words_prefix "<b>"
mark_match_words_suffix "</b>"

setenv server_default_charset iso-8859-1
If the WEB server detects that the WEB OPAC browser is not utf-8 enabled (i.e. is not Netscape 6+ or Microsoft Internet Explorer 5+) the utf-8 parameter in the html files is "naturalized", and the data is sent back from the server in a standard character set. The character set to be used is defined in this line.

setenv set_prox_limit 01000
Sets the limit on the number of documents that will be checked for adjacency, if the aleph_start ADJACENCY is set to "N".
The default is 00500.

setenv set_word_limit 01000
Limits the number of words that will be "collected" when truncation is used (e.g. find a? will do a find on all words beginning with a). If a find limit is defined, this is the number of z97 record numbers (i.e. distinct words) in a given search. Maximum value is 9999. The system will stop the FIND if the number of documents in the set exceeds the defined limit.

setenv sfx_base_url http://demo.exlibrisgroup.com:9003/demo
setenv sfx_base_url_usm50 http://demo.exlibrisgroup.com:9003/demo
The SFX base URL can be ADM-library sensitive. When constructing the openURL link to SFX, the hierarchy where to find the sfx-base-url is as follows:

- Use a web cookie under the name "BASE-URL". The value of the cookie should contain the host-name and the sfx instance to use in the format of http://host-name:port/instance
- Use an environment variable "sfx_base_url_<ADM_LIBRARY>" (as in the above example)
- Use a global variable in www_server.conf named "sfx_base_url" (as in the above example)

For example,

- if no cookie is present and the definitions in www_server.conf contain:
  setenv sfx_base_url http://demo.exlibrisgroup.com:9003/default
  setenv sfx_base_url_usm50 http://demo.exlibrisgroup.com:9003/usm50
  setenv sfx_base_url_usm51 http://demo.exlibrisgroup.com:9003/usm51
- And if the profile of the patron (or the IP profile or the ALEPH profile) has USM50 as the bor-library, the base-url will always be http://demo.exlibrisgroup.com:9003/usm50.
- A patron from library USM52 will have the default base of http://demo.exlibrisgroup.com:9003/default.
setenv www_base USM01
Default database login

www_bor_registration_ends 20050101
When registering as a new user in the WEB OPAC, if the "address valid to" field is blank, the date defined in this variable sets the value.

setenv www_browse_previous Y/N
- Y - show one line before scan start
- N - no preview. List will display from requested entry. For example:
  Patron requests scan of author "jones":
  where set to N. The list displays from “jones”
  where set to Y. The list displays one entry before “jones”

setenv www_con_lng ENG
Default language of interface

setenv www_course_email_address
The default email address to which course reading material will be sent (from the WEB OPAC) for addition into the course reading module.

Note that for the option for saving the basket contents from the Web OPAC into a server file plus notifying the course-reading administration, it is possible to specify inside the web-page the email of the administration. Only if the value is not an e-mail address, (e.g. the default value of the check-box being checked) , the variable www_course_email_address will be consulted.

This option can be used to build a drop-down menu of e-mails where there are several course reading administrations to choose from.

For example , setting the web-page "basket-course" in this manner :

... <input type=checkbox name="SEND_MAIL" value="tsachm@exlibris-usa.com">  
  Check here in order to inform the Course Reading admini...

will send the notification e-mail to tsachm@exlibris-usa.com , while this setting :

... <input type=checkbox name="SEND_MAIL">  
  Check here in order to inform the Course Reading admini...

will send the e-mail to the value defined in "www_course_email_address".

setenv www_dn_link_display 03
Defines number of down links to display in the full view of the parent record. If there are more down links than the number defined in www_server defaults, the full view of the parent record will offer two options:
  viewing the down links in a separate window
  creating a set of down linked records
setenv www_item_schedule 14
Defines the number of days that should be accessible for advanced booking of short loan items. According to the example, a patron will be able to book an item up to 14 days in advance.

setenv www_lib_info_link N/Y
If set to Y, patron can click on the “About library” hypertext link in order to see more information about a branch library.
Note: does not influence ITM4.

setenv www_parallel_search_base USM01-CUN01
The service - www_f_service_find_m_acc in tab_service (UTIL G/1/m)- which activates find-m on bases which are defined in this variable.
Note that the browse lists in the various databases must be defined with the same code.
In order to ensure that long titles are retrieved, the AUTO-TRUNCATE-Z01-FIND switch in tab100 should be set to "Y".

setenv www_short_max_no_locations "05"
Defines the maximum number of locations that will display in the brief display. Note that the system limit is 2000 characters. If either this or the parameter in www_short_max_no_locations is exceeded, a message displays:

9564 L More Libraries (as defined in www_f_heading)

setenv www_short_max_no_ext_links "2"
Defines the maximum number of external links that will display in the brief display (using code EXTL. See www_tab_short.<lng> for more information). lines is limited either by this variable or to 2000 characters. In both cases a heading is displayed to indicate that there are more lines: heading 9568 (as defined in www_f_heading)

setenv www_sort_field "01---D02---A"
Primary and secondary sort keys – as per tab_sort – for displaying keyword search results

setenv www_sort_field_aut "03---A01---D"
setenv www_sort_field_sub "01---D02---A"
setenv www_sort_field_srs "05---A01---D"
Primary and secondary sort keys - as per tab_sort – for an ACC listing – Author (AUT), Subjects, (SUB) and Series (SRS) as per the above lines..

setenv www_update_session_id "UPDATE_ALEPH"
In order to update the current session default values from the WWW interface and not from the GUI.

When a non-identified patron accesses ALEPH through the WEB OPAC, he is assigned the ALEPH profile values. ALEPH profile values are set in program code, which can be overwritten by a setting up an actual ALEPH profile through the GUI ADMIN module. By program definition, the ALEPH profile cannot be updated.
through WEB OPAC “option” (preferences). However, if the library defines a setenv www_update_session_id value, when a patron signs in with this ID, he is able to change the ALEPH profile through “option” (preference).

The purpose of this environment variable is to provide a tool for changing ALEPH profile values, without having to use the GUI ADMIN interface.

Sort routines for loans (z36), hold requests (z37), photocopy requests (z38):

**Sort order – (A)scending, or (D)escending**

```bash
setenv www_user_z36_sort_routine 00
setenv www_user_z36_sort_order A
```

Possible routines:
- 00 - z36-due-date
- 01 - z36-sub-library z36-status z30-collection z36-due-date

```bash
setenv www_user_z37_sort_routine 00
setenv www_user_z37_sort_order A
```

Possible routines:
- 00 - z37-priority z37-request-date z37-open-date z37-open-hour
- 01 - z30-sub-library z30-item-status z30-collection z37-open-date z37-open-hour
- 02 - z30-sub-library z30-item-status z30-collection z37-status z37-open-date z37-open-hour

```bash
setenv www_user_z38_sort_routine 00
setenv www_user_z38_sort_order A
```

Possible routines:
- 00 - z38-open-date

```bash
setenv www_z36h_max_record_limit 100
```
Define the maximum number of z36h records displayed in the WEB OPAC - patron information - Loan history list

```bash
setenv www_z37_course_request_days 20
```
Define how long before a course starts should a hold request be effective

```bash
setenv www_z37_date_to M002
```
Define the end of the time slot when creating hold request.

The variable definition should include one character that describes the time period (D-day, M-month, Y-year) and a 3 digit value for quantity – e.g. M003

This will create a hold request that starts from current date and ends in three months.

If the variable is not defined, or is defined incorrectly, the default "date to" will be three months.

```bash
setenv www_z37_recall_type 03
```
Define the default recall type for requests placed in the WEB OPAC. If the line is missing from www_server.conf, the system defaults the recall type to 01.
The recall type is 01, 02 or 03, where 01 is regular, 02 is rush and 03 is no recall.

Recall is related to item status in tab15.<lng> , col.13 and 14, according to which an item can be recalled or rush recalled. This relates only to the b-cir-13 recall service.

Not all libraries use the recall feature, and not all libraries want every request to automatically be recalled --- some libraries don’t want recall at all; some libraries want the recall to be in effect only if the person placing the request comes to the CIRC desk and asks for recall.

Therefore, the need for the default variable.
For libraries that want all requests to automatically be recalls, this variable should be set to 01. For libraries that want to use recall, but only for librarian initiated requests, should be set to 03.

**www_tab_short.<lng>**

Location of the table: tab directory of the library

Purpose of the table: Brief display format in the Web OPAC

This table defines which bibliographic information will display in the columns of the short display in table format in the WEB OPAC.

The order of the columns defined in this table determines the order of the display, in conjunction with the short-a-head and short-a-body web files in the web files directory.

These files should have placeholders in sequential order, and should have as many placeholders (from 0300) as the number of columns defined in this table.

**Note** that the field that is defined for a column can be dependent on the bibliographic record format (FMT). The alternative definitions are assigned the same identifying number in column 1 of this table.

Example:

<table>
<thead>
<tr>
<th></th>
<th>L Author</th>
<th>02---A01---A 00 00 0100 S 9 -</th>
<th>BK 1#### 700##</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>SE 260##</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>## 1#### 700##</td>
</tr>
</tbody>
</table>

In the above example the **Author** column in the WWW OPAC will be taken from fields:

- 1#### or 700## if the document is a book,
- from field 260## if the document is a journal,
- and in any other case it will be taken from field 1#### or 700##.

A single column can be set to display more than one field of information, by:

- repeating the column number in col. 1,
• entering the name of the field in col. 3,
• and the field code in col. 12.

For example:

<table>
<thead>
<tr>
<th>No.</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Title</td>
<td>03---A01---A 00 00 0100 S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BK 245## 240##</td>
</tr>
<tr>
<td>2</td>
<td>Type</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>## TYP##</td>
</tr>
</tbody>
</table>

The second column (2) contains two fields
• the Title field
• and the TYP field.

It creates a virtual field TYP indicating the type of record - (Electronic Resource, Web Resource, Map, Kit etc.), based on coding in
• LDR (pos. 06,07),
• 006 (pos. 00), 007 (pos. 00,01) and
• 008 (pos. 23)

The TYP field can be used for indexing and display, depending on the setup of tab-expand. Note that the name "TYP" will not be displayed. Note too that this option will not work with the special types of fields like ITM+ and RANK.

Note: ! There are special "field codes" for displaying and, in some cases, linking to item information:
• **EXTL** - displays all fields that contain $$u (URL), such as 856, 505, 520, 555, 583, 545, 530 etc. (Depends on the tab_buf_z403 table definitions.)
• **ITMG** -displays message 9052 as link to global holdings display
• **ITM+** -displays each sublibrary+collection, with link, and with total no. of items / items on loan
• **ITMH** - link to holdings and items, each sublibrary + collection listed on a separate line. Note that the holdings and items display is NOT sensitive to collection, and displays the data at the sublibrary level. Up to 200 sublibraries can be displayed. **Note:** Will only display if there is a holding (HOL) record
• **LOC+** -like LOC (displays the LOC field, created by expand_doc_bib_loc_disp and using edit_field), + link to relevant holdings.
• **LOC** - displays the LOC field, created by expand_doc_bib_loc_disp and using edit_field
• **PST** - like PSTS, for the PST field created by expand_doc_bib_loc_4_n and expand_doc_sort_x group of programs (refer to ALEPH User Guide/ Indexing section / Expand routines, for more information related to this set of programs).
• **PSTS** - similar to LOC+ (displays the PSTS field, created by expand_doc_bib_psts_disp and using edit_field), + link to relevant holdings.

Note that LOC/PSTS/ITM+/LOC+ are limited either by 2000 characters or by the parameter defined in:

```
setenv "www_short_max_no_locations"
```
in www_server.conf. In both cases a message is displayed to indicate that there are more lines: The headings are 9564 for ITM+ and 9553 for LOC, LOC+, PST, PSTS in alephe/error_lng/www_f_heading
The same applies to EXTL: the number of lines is limited either by the variable: "www_short_max_no_ext_links"
in www_server.conf or to 2000 characters. In both cases a heading is displayed to indicate that there are more lines: alephe/error_lng/www_f_heading - heading 9568.

Structure of the table:
Col.1 Group no.
Col. 2 ALPHA
Col. 3 Name of column (20 characters)
Col. 4 sort key as defined in tab_sort)
   The first 6 positions define the first sort key.
   The following 6 positions define the second sort key.
   Note that A and D stand for Ascending and Descending
Col. 5 For fixed fields starting position in field from which to take information. 00 for entire field
   Note that if you want to display "year", you can identify this by code YR in column 12, if you have set expand_doc_yr in tab_expand section WEB-BRIEF.
Col. 6 For fixed fields, number of characters to take for display. 00 for entire field.
Col. 7 Length of field to display for variable fields. Enter 0000 for fixed fields
Col. 8 Identifier of the line to use from edit_field.<lng> when formatting the data. If the field code is "LOC" (for library location) and this position is "L", there is a link from the library name to the items list.
Col. 9 filter sub-field code
Col. 10 filter text - together with previous column, defines whether the field should be taken for the column, dependent on:
   • presence of a particular sub-field and content,
   • absence of a sub-field (where absence is identified as ‘-‘ in text)
Col. 11 document record format (FMT) code
Col. 12 first field code
Col. 13-16 Alternate field codes.

Note that within the www_tab_short.<lng> table it is possible to define sort options for the table. E.g. in the table example below, clicking on the Title (as defined in column 3), will sort the short display according to the options set in columns 4-7 for that line.

Order of Sub-Libraries with codes ITM and ITM+: 
The first sub-library to display will be the default sub-library, set through the profile record, (Z61-BOR-SEARCH-SUB-LIBRARY).
The order of the rest sub-libraries will be alphabetical.
as set in the table tab_sub_lib_sort

Order of Sub-Libraries with codes LOC, LOC+, PSTS
as set in the table tab_sub_lib_sort

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>!-!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td>!!!!-</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>L Author</td>
<td>02---A01---A 00 00 0100 S 9 -</td>
<td>## 1#### 700##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>L Format</td>
<td>S</td>
<td>## TYP## FMT##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>L Title</td>
<td>03---A01---A 00 00 0100 S</td>
<td>## 245##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>L Year</td>
<td>01---D02---A 00 00 S</td>
<td>## YR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

www_x_func

Location of the table: alephe/tab directory

Purpose of the table: X Server function definitions

Structure of the table:

Col. 1 client version code
Col. 2 function code
Col. 3 program name
Col. 4 parameters

Example of the table:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td>!!!!-</td>
<td>!!!!-</td>
</tr>
<tr>
<td></td>
<td>### BOR-AUTH</td>
<td>www_x_bor_auth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>### BOR-AUTHENTICATE</td>
<td>www_x_bor_auth</td>
<td></td>
</tr>
<tr>
<td></td>
<td>### BOR-BY-KEY</td>
<td>www_x_bor_by_key</td>
<td></td>
</tr>
<tr>
<td></td>
<td>### BOR-COURSE</td>
<td>www_x_bor_course</td>
<td></td>
</tr>
<tr>
<td></td>
<td>### BOR-INFO</td>
<td>www_x_bor_info</td>
<td></td>
</tr>
</tbody>
</table>
## Index

Abbreviations to be used with numbering ... 14  
Abbreviations to full text ......................... 102  
Access definitions to WEB and GUI ............. 98  
Address types for forms ........................... 58  
Addresses of sub-libraries for printouts ....... 208  
Advanced booking schedule ...................... 293  
Advanced routines for the merging of  
document records .................................. 184  
aleph_start ......................................... 74, 75, 116, 258, 317, 326, 332  
alephadm ............................................ 78  
ALEPHADM .......................................... 104  
alephe directory  
alephe_start ....................................... 15  
alephe_startup ..................................... 19  
alephe/error_<lng> ................................ 32  
check_circ .......................................... 21  
pc_tree_e1501 ....................................... 94  
www_f_heading ...................................... 327  
www_r_heading ...................................... 330  
alephe/error_<lng>/check_doc ................... 34  
alephe/error_<lng>/tab  
check_doc .......................................... 24  
alphe/tab directory  
853_chrono ......................................... 14  
853_numbering ....................................... 14  
allowed_languages ................................. 69  
job_list ............................................. 61  
job_list.conf ...................................... 63  
library_relation .................................... 64  
license .............................................. 67  
marc_country_codes ................................ 67  
marc_language_codes ............................... 69  
server_ip_allowed .................................. 98  
tab_alephadm.<lng> ................................ 104  
tab_attr_sub_library .............................. 105  
tab_base.<lng> ...................................... 109  
tab_base_count .................................... 112  
tab_ccl_boolean_operators ....................... 119  
tab_days ............................................ 126  
tab_export ......................................... 140  
tab_ill_unit.<lng> .................................. 167  
tab_io_remote ...................................... 168  
tab_library_group .................................. 175  
tab_location_name<lng> ........................... 177  
tab_mime_type ..................................... 188  
tab_month .......................................... 188  
tab_oclc .......................................... 191  
tab_sub_library.<lng> ............................. 205  
tab_sub_library_address.<lng> .................. 208  
tab_suf_default .................................... 210  
tab_suf_lib ........................................ 211  
tab_version ........................................ 217  
union_global_param ............................... 324  
user_function ...................................... 325  
www_f_func ......................................... 326  
www_x_func ........................................ 340  
alphe/unicode directory  
marc8_xxx_to_unicode ............................ 69  
tab_character_conversion_line .................. 119  
unicode_to_filing_01 ............................. 320  
unicode_to_word_gen ............................. 322  
alphe_root directory  
pc_server_defaults ............................... 71  
www_server.conf ................................... 331  
Allowed and denied tags for users in  
cataloging ......................................... 94  
AUT information on cross references .......... 49  
Authority lists accessed on defined cataloging  
tags .................................................. 97  
Automatic opening of monograph-type items  
in Acquisitions ................................. 282  
Automatic update of the processing status ...... 291  
bib_format .......................................... 53  
Block codes and restrictions ..................... 128  
Blocking loans ..................................... 113  
BNA loader .......................................... 100  
Brief display format in the Web OPAC ......... 336  
Brief format of a record in the Search function  
..................................................... 92  
Brief record structure ............................ 224  
bib_format .......................................... 176  
Building of links between records .......... 227  
Captions for links between records .......... 245  
Cash transaction types ......................... 262  
catalog.ini ......................................... 39  
Cataloging OWN permissions and filter ....... 192  
Cataloging tables  
check_doc .......................................... 23  
check_doc (error messages) ..................... 24  
check_doc.<lng> .................................... 25  
check_doc.<lng> .................................... 26  
check_doc_field_006 ............................. 27  
check_doc_field_007 ............................. 28  
check_doc_field_008 ............................. 29  
check_doc_field_1dr ............................. 30  
check_doc_line .................................... 31  
check_doc_line_contents ......................... 33  
check_doc_mandatory ............................. 34  
check_doc_new_acc ............................... 36  
check_doc_new_acc_aut ......................... 36  
check_doc_tag_text .............................. 37  
check_doc_unique_index ......................... 38  
codes.<lng> ........................................ 38  
fix_doc.<lng> ..................................... 55  
formats.<lng> ..................................... 59  
marc_exp.dat ....................................... 68  
pc_tab_acq_fast_cat.<lng> ...................... 79  
pc_tab_circ_fast_cat.<lng> ...................... 81  
permission.dat ..................................... 94  
scancode.dat ....................................... 97  
tab_fix ............................................. 151  
tab_merge .......................................... 183  
tab_merge_adv_overlay ........................... 184  
tab_merge_overlay ............................... 186  
tab_own ............................................ 192  
tab_preferred ..................................... 195  
tab_tag_text ...................................... 211  
tag_text.dat ....................................... 319
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character conversion tables</td>
<td>69</td>
</tr>
<tr>
<td>Check of pre-defined texts for sub-fields</td>
<td>37</td>
</tr>
<tr>
<td>Check routines</td>
<td>23</td>
</tr>
<tr>
<td>Check routines for user authorization and payment related to digital rights</td>
<td>39</td>
</tr>
<tr>
<td>check_circ</td>
<td>22</td>
</tr>
<tr>
<td>check_doc.&lt;lng&gt;</td>
<td>26</td>
</tr>
<tr>
<td>check_doc_doc</td>
<td>25</td>
</tr>
<tr>
<td>check_doc_mandatory</td>
<td>25</td>
</tr>
<tr>
<td>Checks between tags</td>
<td>26</td>
</tr>
<tr>
<td>checks for hold requests</td>
<td>161</td>
</tr>
<tr>
<td>checks for loans and renewals</td>
<td>122</td>
</tr>
<tr>
<td>Checks on contents of a tag</td>
<td>33</td>
</tr>
<tr>
<td>Checks on forbidden errors in a document record</td>
<td>34</td>
</tr>
<tr>
<td>Checks on MARC21 tag 006</td>
<td>27</td>
</tr>
<tr>
<td>Checks on MARC21 tag 007</td>
<td>28</td>
</tr>
<tr>
<td>Checks on MARC21 tag 008</td>
<td>29</td>
</tr>
<tr>
<td>Checks on tags</td>
<td>31</td>
</tr>
<tr>
<td>Checks on the MARC ldr</td>
<td>30</td>
</tr>
<tr>
<td>Checksum programs</td>
<td>124</td>
</tr>
<tr>
<td>circl.ini</td>
<td>273, 280</td>
</tr>
<tr>
<td>circe_status_holding</td>
<td>327</td>
</tr>
<tr>
<td>Circulation tables</td>
<td></td>
</tr>
<tr>
<td>cash_09_translate</td>
<td>21</td>
</tr>
<tr>
<td>check_circ</td>
<td>21</td>
</tr>
<tr>
<td>check_circ_override</td>
<td>22</td>
</tr>
<tr>
<td>pc_tab_fast_circ</td>
<td>88</td>
</tr>
<tr>
<td>tab_block_circ</td>
<td>113</td>
</tr>
<tr>
<td>tab_bor_address</td>
<td>114</td>
</tr>
<tr>
<td>tab_bor_id.&lt;lng&gt;</td>
<td>116</td>
</tr>
<tr>
<td>tab_check_circ</td>
<td>122</td>
</tr>
<tr>
<td>tab_delinq.&lt;lng&gt;</td>
<td>128</td>
</tr>
<tr>
<td>tab_fast_patron_register</td>
<td>142</td>
</tr>
<tr>
<td>tab_hold_request</td>
<td>161</td>
</tr>
<tr>
<td>tab_hold_request_form</td>
<td>163</td>
</tr>
<tr>
<td>tab_latest</td>
<td>174</td>
</tr>
<tr>
<td>tab_photo_request</td>
<td>194</td>
</tr>
<tr>
<td>tab_sip2_alert</td>
<td>200</td>
</tr>
<tr>
<td>tab_sip2_conf</td>
<td>201</td>
</tr>
<tr>
<td>tab_sip2_translate</td>
<td>202</td>
</tr>
<tr>
<td>tab14</td>
<td>253</td>
</tr>
<tr>
<td>tab15.&lt;lng&gt;</td>
<td>254</td>
</tr>
<tr>
<td>tab16</td>
<td>256</td>
</tr>
<tr>
<td>tab17</td>
<td>261</td>
</tr>
<tr>
<td>tab18.&lt;lng&gt;</td>
<td>262</td>
</tr>
<tr>
<td>tab24</td>
<td>272</td>
</tr>
<tr>
<td>tab27</td>
<td>274</td>
</tr>
<tr>
<td>tab30</td>
<td>275</td>
</tr>
<tr>
<td>tab31</td>
<td>276</td>
</tr>
<tr>
<td>tab32</td>
<td>277</td>
</tr>
<tr>
<td>tab34</td>
<td>280</td>
</tr>
<tr>
<td>tab37</td>
<td>283</td>
</tr>
<tr>
<td>tab37_campus_filter</td>
<td>285</td>
</tr>
<tr>
<td>tab37_campus_preferred</td>
<td>286</td>
</tr>
<tr>
<td>tab38</td>
<td>287</td>
</tr>
<tr>
<td>tab39</td>
<td>288</td>
</tr>
<tr>
<td>tab41</td>
<td>290</td>
</tr>
<tr>
<td>tab43</td>
<td>293</td>
</tr>
<tr>
<td>tab44</td>
<td>295</td>
</tr>
<tr>
<td>Client definitions to access the pc_server</td>
<td>217</td>
</tr>
<tr>
<td>Collection codes and names</td>
<td>290</td>
</tr>
<tr>
<td>columns in GUI screens</td>
<td>82</td>
</tr>
<tr>
<td>Configuration for Advanced Library Statistics (DBS)</td>
<td>127</td>
</tr>
<tr>
<td>Conversion of field codes</td>
<td>244</td>
</tr>
<tr>
<td>Country codes that may be used in MARC records</td>
<td>67</td>
</tr>
<tr>
<td>create links</td>
<td>133</td>
</tr>
<tr>
<td>Creation of a virtual field from two or more fields</td>
<td>136</td>
</tr>
<tr>
<td>Creation of items and holdings</td>
<td>156</td>
</tr>
<tr>
<td>Custom brief display format for Web OPAC</td>
<td>329</td>
</tr>
<tr>
<td>Database definitions</td>
<td>110</td>
</tr>
<tr>
<td>Day codes and abbreviations</td>
<td>126</td>
</tr>
<tr>
<td>Default fields for a new record</td>
<td>320</td>
</tr>
<tr>
<td>Default tag values when creating records</td>
<td>211</td>
</tr>
<tr>
<td>Default values for GUI applications</td>
<td>71</td>
</tr>
<tr>
<td>Default values for item records created using OCLC server and RLIN loader</td>
<td>232</td>
</tr>
<tr>
<td>Default values for p-file-98 (BNA loader)</td>
<td>100</td>
</tr>
<tr>
<td>Default values for p-file-99 (Marcive loader)</td>
<td>101</td>
</tr>
<tr>
<td>Default values for the Web</td>
<td>331</td>
</tr>
<tr>
<td>Default values the YBP loader (p-file-96)</td>
<td>223</td>
</tr>
<tr>
<td>Definition of ILL centers for the External ILL module</td>
<td>167</td>
</tr>
<tr>
<td>Delivery times from remote storage</td>
<td>272</td>
</tr>
<tr>
<td>Direct indexes</td>
<td>250</td>
</tr>
<tr>
<td>Display of bibliographic fields in the Acquisition's Order index</td>
<td>103</td>
</tr>
<tr>
<td>Display order of the items</td>
<td>169</td>
</tr>
<tr>
<td>DisplayTagInfo</td>
<td>39</td>
</tr>
<tr>
<td>Document format definition for GUI/WWW display</td>
<td>42</td>
</tr>
<tr>
<td>Due dates, fines &amp; limits</td>
<td>256</td>
</tr>
<tr>
<td>EDI incoming messages attributes</td>
<td>40</td>
</tr>
<tr>
<td>EDI orders</td>
<td>281</td>
</tr>
<tr>
<td>EDI outgoing messages attributes</td>
<td>41</td>
</tr>
<tr>
<td>edit_doc</td>
<td>53, 91</td>
</tr>
<tr>
<td>edit_doc.&lt;lng&gt;</td>
<td>110</td>
</tr>
<tr>
<td>edit_doc_999,&lt;lng&gt;</td>
<td>49, 51, 110, 117, 137, 139, 197, 204, 228, 246, 318, 330, 332</td>
</tr>
<tr>
<td>edit_doc_999.aut.&lt;aut_lib&gt;;&lt;lng&gt;</td>
<td>49</td>
</tr>
<tr>
<td>edit_doc_999.aut.&lt;aut_library&gt;,&lt;lng&gt;</td>
<td>110</td>
</tr>
<tr>
<td>edit_field</td>
<td>47, 51, 53, 92, 96, 118, 279</td>
</tr>
<tr>
<td>edit_field.&lt;lng&gt;</td>
<td>42, 49</td>
</tr>
<tr>
<td>edit_paragraph</td>
<td>270, 271</td>
</tr>
<tr>
<td>edit_paragraph.&lt;lng&gt;</td>
<td>20, 42, 51</td>
</tr>
<tr>
<td>Editing of fields and headings for OPAC display and print</td>
<td>50</td>
</tr>
<tr>
<td>End-subfield punctuation</td>
<td>209</td>
</tr>
<tr>
<td>Error messages linked to circ tables</td>
<td>21</td>
</tr>
<tr>
<td>Events definitions</td>
<td>129</td>
</tr>
<tr>
<td>Expand programs</td>
<td>131</td>
</tr>
<tr>
<td>expand_doc</td>
<td>53</td>
</tr>
<tr>
<td>expand_doc_bib_loc.[n]</td>
<td>46, 92</td>
</tr>
<tr>
<td>expand_doc_bib_loc.4_n</td>
<td>338</td>
</tr>
</tbody>
</table>
tab_alephadm.<lng> .............................................. 104
load MARCIVE Records
p-file-99 ...................................................... 179
Load YBP Records
p-file-96 ...................................................... 179
Locating a record in other libraries .......... 176
Location names for Z39.50 server .......... 177
Logical bases (record counters) in Web and
GUI browse list ........................................... 112
Lost material costs ...................................... 280
management of patron ID’s ......................... 116
Mapping bibliographic values into the relevant
holdings and/or item values ......................... 159
Mapping values when creating records with
the RLIN loader (UTIL E/3) ......................... 178
Marcive loader .............................................. 101
Match specifications for checking input file
against the database .................................. 181
Match specifications for importing files .... 179
Matching fields for p-serial-52 (import 85XX
records) ...................................................... 99
Matching of BIB library's Headings (ACC)
index to AUThority records ....................... 108
Material types ............................................. 273
Merge routines ............................................. 183
merge_doc_adv_overlay ............................. 196
Messages between libraries ....................... 230
Messages to the patron in the Course Reading
Module ...................................................... 330
Messages to the patron in the WEB OPAC 327
Minimum user level for override ............. 22
Month codes and abbreviations ............... 188
Moving records .......................................... 189
Multilingual authorities links ................. 268
NACO Normalization Rules ....................... 143
Names of months and seasons .................... 14
Navigation Window .................................... 94, 133
Node environment definitions .................... 15
Non-filing indicator ..................................... 243
Non-high lighted words ............................. 246
OCLC Loader ............................................. 179, 181
OCLC server .............................................. 232
OCLC server configuration table ............... 191
Order of HOL records to be expanded into the
BIB record ............................................... 153
Order transaction types for the order log ... 296
Overdue letter intervals ............................ 277
OWN field .................................................... 232
OWN field values ....................................... 130
p-acq-04 ..................................................... 104, 166
Paragraphs for displaying bibliographic
information ................................................. 52
Parsing call numbers for label printing .... 171
Patron address periods ............................... 114
Patron registration renewal ....................... 275
Patron statuses .......................................... 276
pc_server_defaults ................................. 17, 76, 82, 203, 332
pc_tab/catalog directory of the library
codes.<lng> .................................................. 38
fix_doc.<lng> ............................................ 55
formats.<lng> ............................................. 59
marc_exp.dat ............................................. 68
permission.dat ........................................... 94
scancode.dat ............................................. 97
tag_text.dat ............................................. 319
tagonnew.dat .......................................... 320
pc_tab_col.<lng> ........................................ 112
pc_tab_exp_field.<lng> 87, 104, 165, 166, 233,
276
pc_tab_exp_field_extended.<lng> ............ 276
p-cir-08 ..................................................... 122
p-cir-10 ..................................................... 122
p-cir-12 ..................................................... 255
p-cir-13 ..................................................... 255, 314
p-cir-22 ..................................................... 255
p-cir-24 ..................................................... 285, 286
p-cir-50 ..................................................... 278
p-cir-77 ..................................................... 122
tab directory of the library
check_doc........................................... 23
check_doc.<lng>.................................... 25
check_doc_doc................................. 26
check_doc_field_008.......................... 29
check_doc_field_1dr.......................... 30
check_doc_line................................. 31
check_doc_line_contents...................... 33
check_doc_mandatory......................... 34
check_doc_new_acc............................. 36
check_doc_tag_text........................... 37
check_doc_unique_index....................... 38
edit_doc.<lng>.................................. 42
edit_doc_777.................................... 43
edit_doc_999.<lng>.............................. 44
edit_field.<lng>.................................. 50
edit_paragraph.<lng>.............................. 52
expand_doc_bib_z30............................ 54
form_print_method............................. 57
path_convert.................................... 70
pc_tab_short.<lng>.............................. 92
tab_abbrev...................................... 102
tab_alephadm.<lng>............................... 104
tab_events.<lng>.................................. 129
tab_expand....................................... 131
tab_expand_duplicate_field.................. 134
tab_expand_external.......................... 135
tab_expand_extract......................... 136
tab_expand_join............................... 136
tab_expand_join_simple...................... 138
tab_expand_sort............................... 139
tab_expand_split............................. 140
tab_filing....................................... 143
tab_fix............................................ 151
tab_locate....................................... 176
tab_mapping...................................... 178
tab_match........................................ 179
tab_match_acc................................. 181
tab_merge........................................ 183
tab_merge_adv_overlay....................... 184
tab_merge_overlay............................. 186
tab_own.......................................... 192
tab_preferred.................................. 195

tab directory of the BIB and ADM libraries
pc_tab_acq_fast_cat.<lng>.................... 79
pc_tab_circ_fast_cat.<lng>................... 81
tab_z30_sort.................................... 233
tab_z33.<lng>................................... 279

tab directory of the BIB and AUT libraries
pc_tab_sear.<lng>................................ 90

\begin{center}
tab directory of the BIB library
check_doc_field_006............................ 27
check_doc_field_007............................ 28
check_doc_new_acc.aut........................ 36
drm_routines.................................... 39
edit_doc_999_aut.<aut_lib>.<lng>.......... 49
form_description.<lng>........................ 56
pc_tab_cat_conv............................... 80
pc_tab_col.<lng>................................ 82
tab_aut........................................... 108

tab_buf_z403.................................. 117
tab_cat_hidden_fields......................... 118
tab_fix_local_notes.conf..................... 153
tab_fix_z103................................... 154
tab_hol_mapping................................ 160
tab_item_list_order........................... 169
tab_itm_mapping................................ 170

tab_mapping_mng500........................... 179
tab_move_record.............................. 189
tab_move_record.............................. 189

tab_mapping_mng500........................... 179

tab_move_record.............................. 189
tab_service ............................................. 197

tab_sort ................................................. 202

tab_sub_library_address.<lng> .................. 208

tab_subfield_punctuation .......................... 209

tab_tab_short.<lng>.................................. 211

tab_tag_text ............................................ 212

tab_type_config.<lng> .............................. 213

tab_word_breaking .................................... 220

tab_z103 ................................................. 227

tab_z105 ................................................. 230

tab_z105_filter ....................................... 231

tab00.<lng> ........................................... 239

tab01.<lng> ........................................... 240

tab02................................................... 241

tab04................................................... 242

tab100.................................................. 243

tab11_acc .............................................. 247

tab11_aut .............................................. 249

tab11_ind .............................................. 250

tab11_word ............................................. 251

union_preferred ...................................... 324

www_tab_short.<lng> ................................. 336

tab directory of the USR library

ldap.conf ............................................. 63

tab_block_circ ........................................ 113

tab_bor_address ..................................... 114

tab_bor_id.<lng> ..................................... 116

tab_fast_patron_register .......................... 142

lab/import directory of the BIB library
generic_fix ......................................... 59

tab_98_def .......................................... 100

tab_99_def .......................................... 101

tab_hol_item_create ............................... 155

tab_hol_item_map .................................... 159

tab_yankee_def .................................... 223

tab_acq_index ....................................... 166, 167

tabattr_sub_library .......... 122, 123, 235, 237

tab_aut .............................................. 36, 66, 267

tab_base.<lng> .................................... 42, 44, 91, 112

tab_base_count ..................................... 110

tab_block_circ ..................................... 123

tab_bor_id.<lng> ................................... 16

tab_buf_z403 ....................................... 47, 338

tab_character_conversion ....................... 321

tab_character_conversion_line 40, 41, 145, 220

tab_check_circ .................................... 113, 259, 275, 311

tab_checksum ........................................ 126

tab_code_prefix .................................... 125

tab_expand ........................................... 103, 213, 244, 271, 337

tab_filing ........................................... 150, 203, 224, 240, 241, 248, 251, 321

tab_fix .......................... 55, 66, 183, 184, 187, 213, 243, 244

tab_hold_request ................................... 194

tab_hold_request ................................... 255

tab_ill_charge ...................................... 263

tab_ill_in_index .................................... 166, 167

tab_item_history ..................................... 309

tab_item_list_order .............................. 111, 204

tab_late_return ..................................... 259

tab_locate ............................................. 176

tab_match .......................................... 181, 182, 196, 197

tab_merge ........................................... 184, 187, 192, 196

tab_merge_adv_overlay ............................ 325

tab_merge_overlay ................................ 153, 184

tab_oclc .............................................. 232

tab_photo_request .................................... 255

tab_preferred ..................................... 184, 185, 187, 325

tab_sear.<lng> ....................................... 203

tab_service ......................................... 47, 48, 334

tab_sfx .............................................. 199

tab_sort ............................................. 91, 328, 329, 335, 338

tab_sub_lib_sort ................................. 46, 169, 339

tab_sub_library.<lng> .............................. 177, 204, 254, 257, 262, 285, 288, 304

lab_sub_library_address ........................... 58

lab_subfield_punctuation .......................... 98

lab_suf_default .................................... 211

lab_word_breaking .................................. 252, 323

lab_z0101 ............................................. 226

lab_z0101_text ...................................... 225

lab_z103 .............................................. 45

lab_z121 ............................................. 198

lab_z30_sort ....................................... 273

lab_z311 ............................................. 316

lab00.<lng> .......................................... 91, 144, 203

lab01 .................................................. 243

lab01.<lng> .......................................... 137, 139, 144

lab04 .................................................. 153

lab10 .................................................. See lab100

lab100 .......................... 17, 74, 123, 192, 197, 257, 263, 275, 282, 334

lab11_acc ............................................ 136, 137, 139, 185, 197, 242

lab11_aut ............................................. 44

lab11_ind ............................................ 180, 185, 197, 239

lab11_word .......................................... 44, 222

lab15.<lng> ........................................... 123, 161, 194, 206, 327, 336

lab16 .......................... 17, 123, 162, 174, 206, 255, 261, 262, 263, 265, 308

lab17 ................................................. 162, 206, 272, 293, 308

lab18 .................................................. 165, 278, 280

lab18.<lng> ........................................... 287, 311

lab20 .................................................. 49

lab24 .................................................. 162

lab25.<lng> ........................................... 123

lab30 .................................................. 123

lab31 .................................................. 259, 308

lab32 .................................................. 280

lab34 ................................................. 262, 264, 265, 278

lab37 ................................................. 19, 75, 161, 206, 263, 264, 265

lab38 .................................................. 19, 194, 206

tag_text ............................................... 37

Text codes list and definitions for brief record

(Z0101) ................................................. 226

Time and rate for fine method "F" ............ 174

two-level vendor .................................. 216

ue_01 .................................................. 133

ue_06 ............................................... 255, 288, 290

ue_08 ............................................... 267

Unicode character conversion equivalencies

323
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicode character conversion equivalencies for sorting</td>
<td>320</td>
</tr>
<tr>
<td>Unicode conversion tables</td>
<td>119</td>
</tr>
<tr>
<td>unicode_to_filing</td>
<td>321</td>
</tr>
<tr>
<td>unicode_to_word_gen</td>
<td>222</td>
</tr>
<tr>
<td>Union deduplication parameters</td>
<td>324</td>
</tr>
<tr>
<td>union_preferred</td>
<td>184</td>
</tr>
<tr>
<td>User authorization definitions and codes</td>
<td>326</td>
</tr>
<tr>
<td>user_function.&lt;lng&gt;</td>
<td>300</td>
</tr>
<tr>
<td>UTIL A/17/1</td>
<td>256</td>
</tr>
<tr>
<td>UTIL E/11</td>
<td>18</td>
</tr>
<tr>
<td>UTIL E/5</td>
<td>178</td>
</tr>
<tr>
<td>RLIN loader</td>
<td>232</td>
</tr>
<tr>
<td>UTIL E/6</td>
<td>19</td>
</tr>
<tr>
<td>UTIL E/8</td>
<td>16, 109, 267</td>
</tr>
<tr>
<td>UTIL G/2</td>
<td>73, 309, 316, 317</td>
</tr>
<tr>
<td>UTIL H/1</td>
<td>317</td>
</tr>
<tr>
<td>UTIL H/I/10</td>
<td>111</td>
</tr>
<tr>
<td>UTIL M/7</td>
<td>95, 319</td>
</tr>
<tr>
<td>UTIL Y/6</td>
<td>71</td>
</tr>
<tr>
<td>UTIL/F/1/21</td>
<td>324</td>
</tr>
<tr>
<td>Valid tags and aliases for the Cataloging Client</td>
<td></td>
</tr>
<tr>
<td>Validation messages (system driven) for check_doc programs</td>
<td>24</td>
</tr>
<tr>
<td>Validation messages (table driven) for check_doc programs</td>
<td>25</td>
</tr>
<tr>
<td>VAT percentages</td>
<td>215</td>
</tr>
<tr>
<td>WEB OPAC services</td>
<td>197</td>
</tr>
<tr>
<td>web service search engine</td>
<td>231</td>
</tr>
<tr>
<td>Weight factors for determining candidate records</td>
<td>218</td>
</tr>
<tr>
<td>Weighting table</td>
<td>325</td>
</tr>
<tr>
<td>Word breaking procedures</td>
<td>220</td>
</tr>
<tr>
<td>Word indexes</td>
<td>251</td>
</tr>
<tr>
<td>www_f_heading</td>
<td>46, 328, 334, 335, 338</td>
</tr>
<tr>
<td>www_server.conf</td>
<td>16, 74, 75, 77, 197, 256, 295, 338</td>
</tr>
<tr>
<td>www_server_defaults</td>
<td>203</td>
</tr>
<tr>
<td>www_tab_short.&lt;lng&gt;</td>
<td>51, 203, 204, 328, 335, 339</td>
</tr>
<tr>
<td>X Server function definitions</td>
<td>340</td>
</tr>
<tr>
<td>XML_XSL</td>
<td>57</td>
</tr>
<tr>
<td>XYZ</td>
<td>223</td>
</tr>
<tr>
<td>YBP loader</td>
<td>223</td>
</tr>
<tr>
<td>Z01</td>
<td>108, 241, 267</td>
</tr>
<tr>
<td>z103</td>
<td>246</td>
</tr>
<tr>
<td>Z103</td>
<td>246</td>
</tr>
<tr>
<td>Z13 fields definition</td>
<td>270</td>
</tr>
<tr>
<td>Z30-MATERIAL</td>
<td>273</td>
</tr>
<tr>
<td>Z311 table</td>
<td>238</td>
</tr>
</tbody>
</table>