System Librarian's Guide

Version 23

Ex Libris Confidential

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: December 24, 2017

Web address: http://www.exlibrisgroup.com

System Librarian's Guide - Acquisitions

Version 23

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: April 4, 2017

Web address: http://www.exlibrisgroup.com

Table of Contents

1 ORDER INDEX 6

2 PRINT TEMPLATES 7

2.1 Available Templates 7

3 OPEN ANNUAL BUDGETS 8

4 RENEW ORDER ENCUMBRANCES 8

5 TRANSFER REMAINING BALANCE 9

6 DROP-DOWN MENUS 9

7 BUDGET CHECK 12

8 ARRIVAL FORM 13

9 ITEM RECORDS 13

10 ORDER NUMBER COUNTERS 14

11 ISBN/ISSN 15

12 ORDER STATUS AND ITEM PROCESS STATUS 16

13 ORDER LOG 16

13.1 Order/Subscription/Invoice Log - Sorting Routine 16

14 ITEM PRICE FOR MONOGRAPH ORDER 17

15 VENDOR'S CURRENCY 17

16 ACQUISITIONS-RELATED ENTRIES IN TAB100 17

17 ACQUISITIONS TABLES 23

18 ACQUISITIONS VALUES OF PC_SERVER_DEFAULTS 26

19 SETTING THE LOCAL CURRENCY 28

20 COLUMN HEADINGS (PC_TAB_COL.LNG) 28

21 CLIENT SETUP (ACQ.INI) 29

21.1 ACQ.INI SETTINGS 29

22 INVOICE PAYMENT STATUS (TAB48) 33

23 CHECKSUM OF VENDOR ADDITIONAL CODE FORMAT (ABN) 34

24 BUDGET AND VENDOR CODE PREFIX 34

25 ORDER UNITS 35

26 LINE ITEM INVOICE - ALERT MESSAGE 39

27 ORDER BAR SEARCH OPTIONS 40

28 SENDING A LIST OF ORDERS SENT VIA EDI TO A VENDOR INCLUDING MORE THAN ONE SUBLIBRARY 41

29 SETTING UP THE BUDGET LIST 41

30 SETTING UP ACQUISITIONS FOR EDI 41

31 SETTING UP VENDOR SHARING 42

32 ACQUISITIONS VALUES OF ALEPH_START 42

33 LOAD INVOICE AND LINE ITEMS (ACQ-32) 43

33.1 The Batch Service Interface 43

33.2 The Input File 45

33.3 Service Execution – A Successful Update 52

33.4 The Service Output Files 53

33.5 User Privileges 54

33.6 Statistics Report 54

34 VENDOR NAME IN JAPANESE: KANJI AND KATAKANA 54

34.1 Storing Vendor Name in Kanji and Katakana 54

34.2 Sorting Vendor Name in Katakana 56

34.3 Searching for Vendor Name in Kanji and Katakana 56
34.3.1 Searching for Vendor Katakana Name 57
34.3.2 Vendor Keywords Search – Kanji and Katakana 58

35 UPDATING THE BIB RECORD WITH THE DATE OF THE ISSN FORMAL NOTICE 60

35.1 Configuring tab_z105 60

35.2 Configuring aleph_start 61

36 UPDATING THE BIB STA FIELD AFTER ORDER CREATION AND ARRIVAL 61

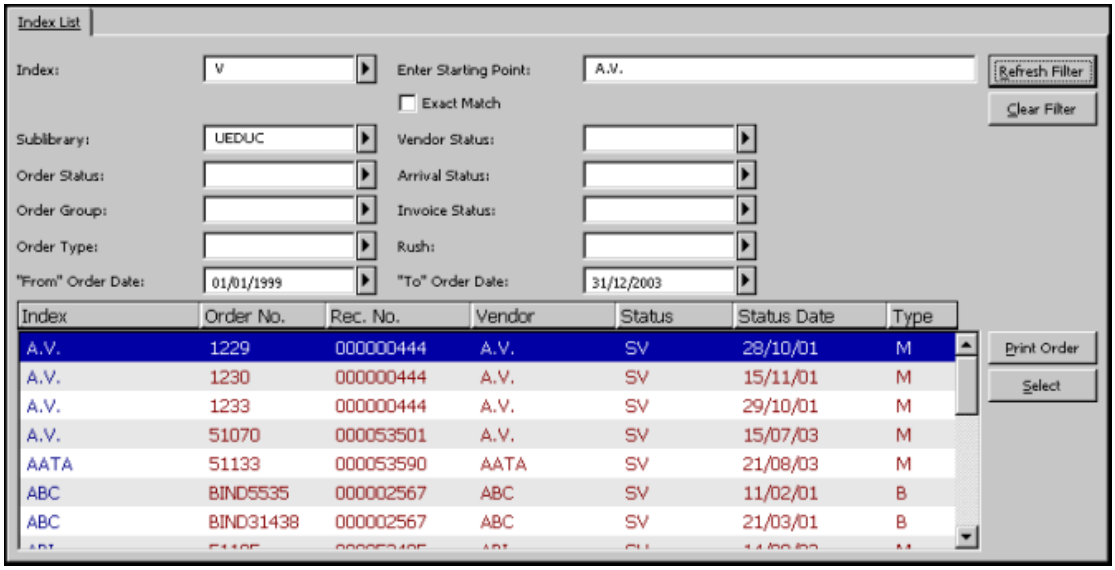
36.1 Configuring tab_z105 61

36.2 Configuring order_sta_update 62

37 ORDER CHANGES ALERT 63

1 Order Index

The Order Index is used in the New Order and Update Order functions, when the user chooses a record to work on (see screen below).



The Order Index enables the user to sort records by various indexes, such as order number, vendor code, etc.

You can add another index by editing the `tab_acq_index` file in the `data_tab` directory of the administrative library of interest.

After adding an index, you must run the "Rebuild Order Index" service in the Services module.

In addition, add the new index(es) to the drop-down menu that appears in the Order Index List screen. This is done by editing the pc_tab_exp_field file. In that file, the menu is identified by the ID "ACQ-INDEX-TYPE". The columns that are displayed in the Order Index List are defined in pc_tab_col.lng: PC_ACQ_INDEX.

In order to set the rules for alphabetizing the Order Index (for example, to define that ä should be filed along with the other "a"s), edit the `tab_character_conversion_line` table.

In the Acquisitions module, the OPAC Request List index also uses the bibliographical indexes defined in ./xxx50/tab/tab_acq_index.
If you want to add a new index code to the drop-down menu that appears in the OPAC Request List Order Index List screen, you should edit ./xxx50/tab/tab/pc_tab_exp_field.lng. In that file, the menu is identified by the ID "ACQ_OPAC_LIST".

2 Print Templates

2.1 Available Templates

The names of the Acquisitions template files are as follows (where *nn* is a number identifying the version):

GUI Templates

Template	**Purpose**	**print.ini function name**
Order Slip acq-m-order-slip-nn (for Monographs) or acq-s-order-slip-nn (for Serials)	Order slip to vendor for material that will be sent to the Acquisitions department or directly to the person for whom it was ordered	OrderLetter
Arrival Slip acq-arrival-slip-nn	Arrival slip to be attached to arriving order and sent to the sublibrary or directly to the person for whom the material was ordered	ArrivalSlip
Order Information acq-m-order-info acq-s-order-info	For printing information in the Acquisitions/Serials GUI module	OrderInfo
Claim Letter (GUI) acq-claim-nn	For printing a claim in the Acquisitions/Serials GUI module	ClaimLetter
Budget Information budget-information-nn	For printing budget information in the Acquisitions/Serials GUI module	BudgetInfo
Budget Transactions Details budget-details-nn	For printing budget transactions details in the Acquisitions GUI module	Transactions

Services Templates

Template	**Purpose**
List of Orders acq-order-list-nn	For the "Send List of Orders to Vendor" service. Send the list to the vendor for material that will be sent to the Acquisitions department or directly to the person for whom it was ordered
Claim Letter (Services) acq-first-claim-nn acq-additional-claim-nn	For the "Claim Report & Letters" service
Budget Summary budget-statuses-nn	For the "Budget Summary" service

Invoice Report invoice-report-nn	For the "Invoice Report" service
Serials Renew Order acq-subs-renew-list-nn	For the "Subscription Renewal Letters" service
Renew Order Encumbrance Report renew-order-encumbrance-nn	For the "Renew Order Encumbrance for Monograph" and "Renew Order Encumbrances for Serials and Standing Orders" services
Transfer Order Encumbrance from Budget to Budget Report transfer-order-encumbrance-nn	For the “Transfer Orders from Budget to Budget” service

Note

Libraries can share the same forms by adding a definition to the `path_convert` table under the library's tab directory. Your ADM library is probably linked to your BIB library:

```
$usm50_dev/usm50/form_eng       $usm01_dev/usm01/form_eng
```

3 Open Annual Budgets

The Open Annual Budgets (acq-05) service is available from the Acquisitions Services menu. It enables you to create new annual budgets for those budgets defined as "annual" using the name format "code-YEAR" (for example, BoOKs-2002). The system will create new budgets having the same code and will add a new suffix for the new year.

A report will be created listing the new budgets that have been opened. The file can be found in the node's ALEPHE/SCRATCH directory.

4 Renew Order Encumbrances

The Renew Order Encumbrances (acq-06) service is available in the Acquisitions Services menu. There are separate services for Monographs: Renew Order Encumbrances for Monographs (acq-06-a), and for Serials and standing orders: Renew Order Encumbrances for Serials (acq-06-b). Before running any of these Renew Order Encumbrances services, you must first run Open Annual Budgets.

The Renew Order Encumbrances service checks for annual budgets that have been assigned to each order. If the encumbrance for the order is still active (i.e., there are incomplete invoices), the encumbrance will be assigned to the next year's budget. If some or all invoices have been paid, the old annual budget remains assigned. The acq-06-b service (Renew Order Encumbrances for Serials and Standing orders) offers the user to select on which base the new year order's encumbrance amount should be renewed: last year’s encumbrance or last year’s payment amount.

Two output files are created:

- **Output File 1:** For cases where no invoices have been paid for a particular order, a report will be created listing the orders for which new budgets have been assigned. The file can be found later in the node's ALEPHE/SCRATCH directory.
- **Output File 2:** For cases where some invoices have been paid for a particular order, a report will be created listing the orders for which the budgets have not changed. The file can be found later in the node's ALEPHE/SCRATCH directory.

5 Transfer Remaining Balance

The Transfer Remaining Balance (acq-07) service is available in the Acquisitions Services menu. Before running this service, you must first run Open Annual Budgets. This service (Transfer Remaining Balance) enables you to transfer the funds remaining in the annual budgets at the end of the year to the new year's annual budgets. This service only works for budgets defined as "annual" and which use the name format "code-YEAR" (for example, Books-2002).

A report will be created listing each annual budget and indicating the amount (if any) that was transferred. The file can be found later in the node's ALEPHE/SCRATCH directory.

6 Drop-down Menus

Most of the choices available on the drop-down menus of the GUI Acquisitions/Serials module have been set by the developers and you can not add or delete choices. Still, there are a few drop-down menus that you can edit. To do this, connect to the administrative library and edit `pc_tab_exp_field.lng`.

You can edit choices for the following menus:

Index (in Index List)
If you have added an order index by editing the `tab_acq_index` file in the `data_tab` directory, be sure to add the new index to the drop-down menu used in the Index field in the Index List (this list is accessed by selecting the Index List node from the Order Search tab). In `pc_tab_exp_field.lng`, the ID identifies the menu: ACQ_INDEX_TYPE.

Order Index Type
The list of search options available on the Order Index Type field's drop-down menu (the left field on the Order bar) is defined in the ORDER-SEARCH menu of pc_tab_exp_field.lng.

Claim Format
If you have added new forms for claim printout, be sure to add them to the list of menu choices. In `pc_tab_exp_field.lng`, the menu is identified by the ID ACQ_CLAIM_FORMAT.

Budget Group 1-5

You can edit choices of the Budget Groups menu. In `pc_tab_exp_field.lng`, the five menus are identified by the IDs: BUDGET-GROUP-1 till BUDGET-GROUP-5. The budget group code must be not more than five characters in order to fit z76-sub-key-1.

Method of Acquisition
You can add new choices to the Method of Acquisitions menu. In `pc_tab_exp_field.lng` the menu is identified by the ID ACQ_ORDER_METHOD.

Material Type
You can add new choices to the Material Type menu. In `pc_tab_exp_field.lng`, the menu is identified by the ID ACQ_ORDER_MATERIAL.

VAT Recipient
You can edit choices of the VAT Recipient menu. In `pc_tab_exp_field.lng`, the menu is identified by the ID VAT_RECEIVER.

Order Group
You can edit choices of the Order Group menu. In `pc_tab_exp_field.lng`, the ID identifies the menu ACQ_ORDER_GROUP.

Budget Type
You can edit choices of the Budget Type menu. In `pc_tab_exp_field.lng`, the ID identifies the menu BUDGET_TYPE.

Order Letter Format
If you have added new forms for Order Letters, be sure to add them to the list of menu choices. In `pc_tab_exp_field.lng`, the ID identifies the menu: ACQ_LETTER_TYPE.

Order Status
You can add new choices to the Order Status menu. In `pc_tab_exp_field.lng`, the ID identifies the menu: ACQ_ORDER_STATUS.

Note

If you add new order statuses, they will not be system-generated and will have to be manually operated by the user.

Material Delivery
You can add new choices to the Material Delivery Type menu. In `pc_tab_exp_field.lng`, the ID identifies the menu: ACQ_ORDER_MAT_DEL.

Invoice Status
You can add new choices to the Invoice Status menu. In `pc_tab_exp_field.lng`, the ID identifies the menu: ACQ_INVOICE_STATUS.

Invoice Type
You can add new choices to the Invoice Type menu. In `pc_tab_exp_field.lng`, the ID identifies the menu: ACQ_INVOICE_TYPE.

Invoice Line Sorting
You can edit the list of sort option of the invoice . In pc_tab_exp_field.lng, the menu is identified by the ID INV-SORT-TYPE.

Vendor Delivery Type
You can add new choices to the Vendor Delivery Type menu. In `pc_tab_exp_field.lng`, the ID identifies the menu: VENDOR_DELIVERY_TYPE.

Budget Department
You can edit the choices of the Budget Department menu. In `pc_tab_exp_field.lng`, the ID identifies the menu: DEPARTMENT_CODE.

Invoice Payment Status (not via pc_tab_exp_field.lng)
You can add new choices to the General Invoice Payment Status.

Note

This menu is not handled in pc_tab_exp_field.lng. In order to add/edit invoice payment statuses use tab48.lng of the Administrative library.

Object Code
You can edit the list of object codes that appears in Budget Information Tab 2 and Invoice Line Item. In `pc_tab_exp_field.lng`, the menu is identified by the ID OBJECT-CODE. Up to 20 object codes can be defined.

Approval Dept.
You can add new choices to the General Invoice Approval Department menu of the General Invoice. In `pc_tab_exp_field.lng` the menu is identified by the ID: ACQ_APPROVAL_DEPARTM. The default value of the approval department can be set in the [Invoice] section of `acq.ini`.

Index (in OPAC Request List)
If you have added an order index by editing the `tab_acq_index` file in the `data_tab` directory, be sure to add the new index to the drop-down menu used in the Index field in the OPAC Request List (this list is accessed by selecting the OPAC Request List node from the Order Search tab). In `pc_tab_exp_field.lng`, the ID identifies the ACQ_OPAC_LIST menu.

Reject Note - OPAC Request
In the OPAC Request List, when the **Reject** button is clicked, the user may select a reason from a pre-define list of reasons. In `pc_tab_exp_field.lng`, the ID identifies the ACQ_REJECT_OPAC_NOTE menu.

7 Budget Check

You can decide that before sending an order to the vendor, the system will check that there are budgets that will cover the encumbrance (estimated sum). The CHECK-ORDER-BUDGET variable in tab100 determines whether or not this check will be performed by the system.

8 Arrival Form

You can have the system automatically update the Arrival Form with the number of units arrived and date arrived, after the user adds a new Line Item to the General Invoice. To do so, go the client's ACQ/TAB directory and open the ACQ.INI file. Go to the section labeled [Invoice]. Following is an example of the relevant section:

```
[Invoice]
CreateArrival=Y
```

If you want the Arrival Form to be automatically updated, type "Y." If not, type "N."

Note that even if the flag is set to "Y," the system will still ask the user to confirm that he wants the material recorded as having arrived. If the user chooses "Yes," the system will then update the Arrival Form. If the flag is set to "N," the system will not ask the user for confirmation, and will not update the Arrival Form.

Vendor Reply

You can add new choices to the list of vendor replies to a claim. In pc_tab_exp_field.lng, the menu is identified by the ID: VENDOR-CLAIM-REPLY.

You can also set up the system to product a pop-up message if any of the checked-in items have been requested. To do this, define the following section in acq.ini:

```
[Arrival]

ArrivalMessages=Y
```

To disable this option, set `ArrivalMessages` to `N`.

9 Item Records

The librarian can choose to have the system automatically create Item Records for a monograph order by checking **Create Items Records** on the Quantity and Price tab of the Order Form. The default value for **Create Items Records** is determined by the tab100 variable CREATE-ITM-FORM-ORDER-M

In tab36, the System Librarian defines per order sublibrary (col.1) and order material type (col.2) the default values that are assigned to the following automatically created item records: item statues, item material type, item collection, item call number, and item call number type.

The following is an example from the table:

```
  1    2  3     4    5                6                                7
!!!!!-!!- !!-!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!
LAW   BK  01 BOOK  GEN                                                0
LAW   DV  02 DVD   GEN                                                0
LAW   MP  02 MAP   GEN                                                0
LAW   ##  01 BOOK  GEN                                                0
##### ##  01 BOOK  GEN                                                0
```

Col .1 – Order Sublibrary filter

Col .2 – Order material type filter
Col. 3 – Item status
Col. 4 – Item material type
Col 5 – Item collection
Col 6 – Item call number
Col 7 – Item call number type

Notes:

The item sublibrary is the same as the order sublibrary.

If the item's collection is not defined in the Quantity and Price tab of the Order form, tab36 is checked.

The actual number of item records that are created is determined by the "number of units" entered in the Quantity and Price tab of the Order Form.

The item values entered in columns 3 thru 6 (inclusive) should match corresponding values as defined in the following Aleph configuration tables: tab25.lng (item material type), tab15.lng (item status), tab40.lng (item collection), and ITEM-LOCATION-TYPE menu of pc_tab_exp_field.lng (item call number type).

10 Order Number Counters

You can set up counters that will be used to automatically assign an order number to a new order. You can set up different counters for use by different libraries, or for different types of orders (for example, monographs vs. serials).

To set up counters, follow these steps:

Step 1:
Use UTIL G/2 to create a new "last-order-no" sequence, with a code attached that identifies the counter (for example, "last-order-no-ueduc").

Step 2:
Go the client's ACQ/TAB directory and open the ACQ.INI file. Go to the section labeled [OrderNumber]. Following is an example of the relevant section:

```
[OrderNumber]
EnablePrefix=Y
PrefixString=ueduc
```

If you want the user to be able to choose a particular counter, go to EnablePrefix and type Y to the right side of the equal (=) sign. The "Order Number Counter Code" field will then appear on the Order Type form. If you do not want the user to be able to choose a particular counter, type N. The "Order Number Counter Code" field will not be displayed.

To determine which counter will be entered as the default in the "Order Number Counter Code" field that appears on the Order Type form, go to PrefixString and type the code that identifies the particular counter (for example, ueduc for "last-order-no-

UEDUC"). Regardless of the counter that appears as the default, the user will be able to change the counter.

Another type of counter that you can use does not appear in UTIL G/2, but you can type its code next to the PrefixString, as follows:

```
PrefixString=DocNo
```

If you type DocNo, the order number that will be assigned will be a combination of the ADM record number and the sequence number of the order (each order that is opened for an ADM record is assigned a running number). Examples of such a number are: 0000500-1, 0000500-2.

If you do not type a code to the right side of the equal (=) sign for the PrefixString parameter, the system will leave blank the "Order Number Counter Code" field that appears on the Order Type form. If the user does not enter a code for one of the defined counters, the system will automatically assign a number based on the "last-order-no" parameter in UTIL G/2.

It is also possible to set up an order counter (from Util/g/2) to be used when a new order record is created from Verde. The [adm_lib]/tab/verde_open_order table, Col 10 contains up to 5 characters and matches the definition of the order counters in util/g/2. Only the first line is filled in. For example, if the column is set to "s" in verde_open_order of XXX50, the system uses the counter "last-order-no-s" of the XXX50 library. If the column is empty or it has a value but the counter does not exist, the default counter "last-order-number" is used.

11 ISBN/ISSN

It is up to you to decide whether or not you want the system to check the validity of the ISBN/ISSN number entered in the Order form. To set up your preference, edit the tab100 table using UTIL G/4/100 in an Administrative library. The relevant tab100 variable is:

```
CHECK-ORDER-ISBN-ISSN
```

If you set it as Y, the system checks the validity of the ISBN/ISSN number. If N, the system does not check the validity and any value entered is accepted.

The ISBN validity check routine is capable of validating both types.

Both 10-digit ISBNs and 13-digit ISBNs are supported by the system and considered valid.

You can populate an order's ISBN/ISSN field with additional bibliographic fields as well as the standard MARC 21 ISSN/ISBN field. You do this by define the additional bibliographic fields in the tab100 ORDER-ISBN-ISSN-ADD-TAG parameter, as in the following example:

```
ORDER-ISBN-ISSN-ADD-TAG=021##a
```

12 Order Status and Item Process Status

You can set the system to automatically change the item process status when the order process status changes. To do so, edit tab42 in an Administrative library.

13 Order Log

The Order Log keeps track of various order-related activities that have occurred, such as a change in the Order Status or Next Claim Date. It is up to you, the System Librarian, to determine which transactions will be included in the Order Log. To do so, edit tab45.

Note:

It is recommended to set Col.3 (Order Log can/cannot be entered manually) to Y only for the following transactions:

00 - General

13 - General Invoice Note

95 - Subscription History Note

98 - Reply from Vendor (no claim)

99 - Note to Vendor

To define the text that will be displayed in the "Details" column of the Order Log, edit `$aleph_root/error_lng/acq_logger`. The transactions that are displayed in the expand list of the transaction type field and can be entered manually are determined by tab45.lng, Column 3.

13.1 Order/Subscription/Invoice Log - Sorting Routine

The sort routine of the Acq Order/Subscription/Invoice Log can be set in `pc_server_defaults`. The relevant lines are:

```
setenv acq_user_z71_sort_routine
setenv acq_user_z71_sort_order
```

The parameters are as follows:

Sorting routines (setenv acq_user_z71_sort_routine):

- 00-by the open date & hour of the transaction
- 01-by sequence no. of the log
- 02-by sequence no. of the log, the logs with the Z71-ACTION-DATE will always be sorted on the top by Z71-ACTION-DATE. If Z71-ACTION-DATE is zero, the system will use instead in Z71-OPEN-DATE,Z71-OPEN-HOUR, and Z71-OPEN-SECOND combined together as string.

Sorting order (setenv acq_user_z71_sort_order):

- A - ascending

- D- descending

Example of setup:

```
setenv acq_user_z71_sort_routine  00
setenv acq_user_z71_sort_order     D
```

14 Item Price for Monograph Order

In order to compute the default price in the Item Record, the system divides the local price by the number of units ordered.

When the order is closed (CLS), the item price is calculated again: All invoices assigned to this order are divided by the number of items.

If the tab100 variable Z30-PRICE-FROM-ORDER is set to N, Z30-PRICE will never be created automatically, either from the estimated price or from invoices. When it is set to Y, Z30-PRICE will be created as explained above.

15 Vendor's Currency

If the variable CHECK-INVOICE-CURRENCY is YES, at least one currency must be entered in the Account tab on the Vendor form; when you assign a general invoice to an order, the system will only allow you to pick one of the four defined currencies. Currency 1 is the default.

If the variable CHECK-INVOICE-CURRENCY is NO, the currency field in the Account tab on the Vendor form is not mandatory; when you assign a general invoice to an order, the system will allow you to choose a currency that is not defined in the Vendor Form.

16 Acquisitions-related Entries in tab100

This section details all the variables that are relevant to Acquisitions and are set in `tab100` of the Administrative library. Most of this information also appears elsewhere in this System Librarian chapter.

BUDGET-PER-ORD-UNIT
This variable determines the filtering mechanism of the Budget List

Possible values are: Y or N.

Y = Budgets will be listed only if their assigned Order Units/Sublibrary (Z602) matches the user's assigned Order Unit/Sublibrary (Z602).

The regular GUI sublibrary filter will not be used.

N = Budgets will be listed according to the regular GUI filters.

CALC-INV-LINE
Determines if the difference between the Total Amount Line Items (Z75) and the Total Amount of General Invoice (Z77) is distributed among the preceding invoice line –items (if less then 0.05).

Possible values are: Y or N.

Y = If the difference between the Total Amount Line Items (Z75) and the Total Amount of General Invoice (Z77) is less than 0.05 then the balance is distributed among the preceding invoice line-items.

N = The difference between the Total Amount Line Items (Z75) and the Total Amount of General Invoice (Z77) is not recalculated.

Default value is Y.

CHECK-INVOICE-CURRENCY
Determines which currency must be assigned to the vendor's record when registering a new vendor (Z70).

Possible values are: Y or N.

Y = currency on invoice must match one of the vendor currencies. When registering a new vendor (Z70), at least one currency must be assigned to the vendor's record.

Default value is N.

CHECK-ORDER-BUDGET
Determines the value for "P" (purchase) orders.

Possible values are: Y or N.

Y = 0.00 is not allowed as the estimated price of an order. When the order is sent, a budget encumbrance is required. Budget checks (validity and balance) are performed. If errors are detected, the order status is changed to DNB (Delay, No Budget) and the order is not sent.

N = 0.00 is allowed as the estimated price of an order. If a budget is assigned, then when the order is sent, budget checks are performed and errors are reported. Even if errors are detected, the order is sent and the order status is not changed to DNB.

If no budget was assigned to the order, the invoice line item can be registered without a budget.

Default value is Y.

CHECK-ORDER-ISBN-ISSN
Checks the validity of the ISBN/ISSN number in the Acquisitions order.

Possible values are: Y or N.

Y = checks the ISBN/ISSN number entered in the Acquisitions order.

Default value is N.

CLS-ORDER-UPON-ARRIVE

This variable controls the automatic closure of monograph and standing order upon order’s arrival completion.

Possible values are: 0 or 1.

0 – Default. Upon completion of monograph/standing order arrival, the order status is not automatically changed to “Closed”. The automatic closure of the order occurs when the order is marked as “completely arrived”, “completed invoiced” or “paid”.

1 – Upon reporting a complete arrival of monograph/standing order, the order status is automatically changed to “Closed:. This occurs regardless to the order’s invoice information.

CREATE-ITM-FORM-ORDER-M
Checks whether the item is automatically created from the order form.

Possible values are: Y or N.

Y = check box for automatic item creation (on tab 3 of order form) is set to "checked".

Default value is Y.

DEFAULT-GEN-INV-STATUS
The possible value is "XXX", where XXX represents a three-character value.

When creating a general invoice, the default value for the invoice status comes from this variable instead of being automatically set to "REG". If this variable is empty in tab100, the default value for the invoice status will be"REG".

This flag is also valid for General Invoices that are created by the EDI load.

DEFAULT-GEN-INV-TYPE
Possible value is "XXX", where XXX represents a three-character value.

When creating a general invoice, the default value for the invoice type comes from this variable instead of being automatically set to "REG". If this variable is empty in tab100, the default value for the invoice status is "REG".

This flag is also valid for General Invoices that are created by the EDI load.

EDI-OUT-VENDOR-NOTE
Possible values are: Y or N

Y = The note will be added to an outgoing EDI message when sent to a vendor

Default value is Y

EDI-OUT-LIBRARY-NOTE
Possible values are: Y or N

Y = The note will be added to an outgoing EDI message when sent to a vendor

Default value is Y

EDI-OUT-PRICE-NOTE
Possible values are: Y or N

Y = The note will be added to an outgoing EDI message when sent to a vendor

Default value is Y

EDI-OUT-QUANTITY-NOTE
Possible values are: Y or N

Y = The note will be added to an outgoing EDI message when sent to a vendor

Default value is Y

ITEM-BOR-INFO-CIRC-ONLY
Determines if patron information is hidden when item information is displayed in the Catalog and Acquisitions modules.

Possible values are Y and N.

Y = Patron Information (such as Patron Name, ID) is only shown in the Circulation module, and not in the Acquisitions and Cataloging modules.

N = Patron Information (such as Patron Name, ID) is shown in all modules.

The default value is N.

ITEM-DATE-FROM-ORDER
Controls the automatic update of the arrival date of monograph orders.

The new flag has the following possible values: 0/1:

0 - Default. The arrival date is not recorded (old functionality).

1 - The arrival date is recorded in the item record (new functionality).This occurs only if the item's arrival date is not populated (Z30-ARRIVAL-DATE=00000000). If the item's arrival date is already populated, the item arrival date is not updated.

Notes:

- This functionality updates all order's related items regardless of the number of units ordered or the number of units arrived and is not related to the **Order completely arrived** checkbox in the Arrival Form or Order Form.

- If an Order arrival record is deleted, the item arrival date is not removed from the item record.

ORDER-ISBN-ISSN-ADD-TAG
Determines which bibliographic field can be used in order to populate an order's ISBN/ISSN field in addition to standard MARC 21 ISBN/ISSN fields.

Possible values: A tag, two indicators, and subfield should be defined for this parameter.

ORDER-ISBN-ISSN-ADD-TAG=021##a
where tag=021, ##=two indicators, a=subfield

OVER-EXP-INCLUDE-ENC
Determines if the maximum over expenditure of a budget is checked in relation to the encumbrance or to the expenditure.

Possible values are: Y or N.

Y = Maximum over expenditure of a budget is checked in relation to the encumbrance. The system checks the maximum over expenditure against the free balance. The free balance is the total allocation - (paid + unpaid invoices + encumbrances).

N = Maximum over-expenditure of a budget is not checked in relation to the encumbrance but in relation to the expenditure. The system checks the maximum over expenditure against the actual balance of the budget. The actual balance is the total allocations - (paid + unpaid invoices).

Default value is Y.

TWO-LEVEL-VENDOR
Determines whether the system uses one or two levels of vendor.

Possible values are: Y or N.

Y = The system uses two levels of vendor.

N = Only one level of vendor is managed.

Default value is N.

UPD-SUBS-FROM-ORDER

The variable UPD-SUBS-FROM-ORDER determines whether or not updating of serial information within the order record will affect the equivalent fields in the order's related subscription record. The relevant fields are: Vendor Code, Vendor Reference, Delivery Type, Subscription Start, and Subscription End Dates.

Values can be:

- Y- Update of a serials order affects its subscription record (new enhancement).
- N- Default value - Update of a serials order does not affect the subscription.

USE-OBJECT-CODE
Determines whether the budget's object code is assigned to the invoice and matches the budget's object codes.

Possible values are: Y or N.

Y = The object code must be assigned to the invoice and must match the assigned budget's object codes.

N = When registering the invoice, there is no check against the budget's object code and filling the invoice's object code is not mandatory. The object code can be used for reporting only.

Default value is N.

USE-ORDER-UNIT
Determines whether the Acquisitions system works with ordering units or with sublibraries. This variable deals with the Budget's (Z76) order unit, the Vendor's (Z70) order unit, the Order's (Z68) order unit and user-passwords (Z66).

Possible values are: Y or N.

N = each sublibrary is an order unit.

Y = ordering units are defined in tab_sub_library.lng using sublibrary type 5.

Default value is N.

VALIDATE-ORDER-FIELDS

Defines whether or not to validate the values entered in GUI-Acquisitions-Order Form and Multi Order Form.

This is relevant for the order's fields: Order Group and Material Format

0 = (default value). There are no special checks for the Order Group and Material Format fields.

1 = The values entered by staff user in Order Group and Material Format fields are always checked to ensure that they are included in the list of possible values. The possible values are defined ACQ_ORDER_GROUP and ACQ_ERM_TYPE menus of the ADM library table: pc_tab_exp_field.<lng>.

VENDOR-SHARING

Determines whether the Vendor filter in Search mode for Vendors is active and is set to filter by a particular sublibrary or order unit.

Possible values are: 0 or 1.

0 = This is the default value if this parameter is not defined in the table. When a Vendor Filter in Search mode for Vendors is active and is set to filter by a specific sublibrary/order unit, vendors that have not been assigned a sublibrary/order unit are considered to be "general use" vendors and are therefore displayed in the list. Such vendors can be assigned to any order. If the vendor has even one sublibrary/order unit, it must match the sublibrary/order unit of the assigned order.

1 = This can be useful for sites that have many vendor records that are irrelevant to a particular library. This can happen when the site is sharing a single vendor table for multiple ADM libraries, or when the site has done a batch load of general vendor records.

If the variable is set to 1, then:

- When a filter is active on the vendor list, vendors that have not been assigned a relevant sublibrary/order unit are not displayed in the list.
- When a vendor is assigned to an order, the vendor must have the same sublibrary/order unit as the order. Vendors that do not have any sublibrary/order unit cannot be used on orders.

Default value is 0.

VIEW-NON-AUTHORIZED

Determines whether the budget information in the lower pane is displayed when “view Authorized” is not checked.

Possible values are: Y or N.

Y = The budget information in the lower pane of the budget list (Balance, Transactions, etc.) is displayed for all budgets listed in the upper pane.

N = The budget information in the lower pane of the budget list (Balance, Transactions, etc.) is not be displayed for non-authorized budgets even when the budget list is not filtered for authorized budgets only.

Default value is N.

Z30-PRICE-FROM-ORDER
Determines if the item price field is controlled and automatically updated from the acquisition price.

Possible values are: Y or N.

Y = the item price field is controlled by, and automatically updated from, the acquisition price (estimated and invoiced).

Default value is N.

17 Acquisitions Tables

1. **edi_in_attr**

   `edi_in_attr` is a table for defining special attributes of incoming EDI messages as needed for specific vendors.

2. **edi_out_attr**

   `edi_out_attr` defines special attributes of outgoing EDI messages as needed for specific vendors.

3. **form_sub_library_address**

   `form_sub_library_address` table defines address types for Acquisitions, ILL and Circulation forms. `form_sub_library_address` works in conjunction with column 2 (address type) of `tab_sub_library_address.lng` (alephe table).

4. **pc_tab_acq_fast_cat.lng**

   `pc_tab_acq_fast_cat.lng` defines fields for quick cataloging in the ACQ GUI. This table has to be edited in the BIB library (for the creation of the BIB record) and in the Administrative library (for the creation of the ADM record).

   There is an option to determine the bibliographic fields in the `pc_tab_acq_fast_cat.lng` table located in the administrative library by using column 5 of the table that will specify that the tag is a BIB tag.

   If the `pc_tab_acq_fast_cat.lng` table located in the administrative library includes at least one line with a tag that is specified as a BIB tag, the `pc_tab_acq_fast_cat.lng` table in the Bibliographic library will be ignored.

   This option can be used by a Multi-ADM environment as well by a single-ADM environment.

5. tab100

`tab100` is the central configuration table for system-level, server-level and library-level variables. The variables that are relevant to ACQ are:

- BUDGET-PER-ORD-UNIT
- CHECK-INVOICE-CURRENCY
- CHECK-ORDER-BUDGET
- CHECK-ORDER-ISBN-ISSN
- CREATE-ITM-FORM-ORDER-M
- EDI-OUT-VENDOR-NOTE
- EDI-OUT-LIBRARY-NOTE
- EDI-OUT-PRICE-NOTE
- EDI-OUT-QUANTITY-NOTE
- ORDER-ISBN-ISSN-ADD-TAG
- OVER-EXP-INCLUDE-ENC
- TWO-LEVEL-VENDOR
- USE-OBJECT-CODE
- USE-ORDER-UNIT
- VENDOR-SHARING
- Z30-PRICE-FROM-ORDER

See also Acquisitions Related Entries in tab100 and tab100's header.

6. **tab35**

   Every sublibrary can have different EDI ID numbers at different vendors. In addition, different sublibraries can have one account with the same vendor. tab35 defines these parameters for each branch library. In addition, the table registers each sublibrary's VAT number.

7. **tab36**

   The system automatically opens items for monograph orders according to the number registered in the **Number of Units** field in the order record if the **Create item records** checkbox on the monograph order form is selected. The default value for the **Create Items Records** is determined by the tab100 variable CREATE-ITEM-FROM-ORDER-M.

   tab36 defines per order sublibrary (col.1) and order material type (col.2) the default values that are assigned to the following automatically created item records: item statues (col.3), item material type (col.4), item collection (col.5), item call number (col.6), and item call number type (col.7) .For more details, see section 9 Item Records.

8. **tab42**

   `tab42` defines parameters for automatic update of the item processing status, according to the acquisition order status for monograph orders.

9. **tab45.lng**

   tab45 defines whether a transaction creates a log for related actions of Order, Subscription, and Invoice Payment.

   tab45 also defines whether the transaction is system generated or can be entered manually.

   The order/subscription/invoice log is part of the GUI-ACQ module. It depends on definitions set in tab45. For example, an entry can be made when an order or item is created, when the order status or item process status is changed, when subscriptions are created, when invoice payment status is updated, or when the user manually adds his own log notes.

   For more information, see tab45.lng.

10. **tab_acq_index**

    `tab_acq_index` indexes orders by bibliographic fields. Note that the index entry is updated only if the order record is updated – that is, a change in the bibliographic record is not automatically updated in the index in the Acquisitions/Serials Client.

11. **tab_vendor_sub_lib**

This table 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. For more information and for the group definition of Z70, see the table's header.

12. tab_checksum

`tab_checksum` is used for defining the checksum programs. The programs can be used for creation, validation and correction of item barcode, patron barcode, vendor code prefix, budget code prefix, ILL supplier code prefix, vendor additional code, ILL supplier additional code. The relevant programs for Acquisitions are:

- vendor_abn
  This program performs ABN (Australian Business Number) checksum on Vendor Additional Code (Z70-ADDITIONAL-VENDOR-CODE). If data is entered in the Vendor Additional Code field, then it must match the ABN check. Alternately, the field can be left empty (with no data).

- code_prefix
  This program performs a check of the vendor code prefix and the budget code prefix. It works in conjunction with the prefixes that are defined in the Administrative table: tab_code_prefix.

13. tab_code_prefix

This table is used in the Administrative library for defining prefixes for budget code, vendor code and ILL supplier code. This table works in conjunction with the Administrative table: `tab_checksum`. If the checksum programs for vendor code, supplier code and budget code are used in `tab_checksum`, then the check is compared to the prefixes that are defined in `tab_code_prefix`.

18 Acquisitions Values of pc_server_defaults

pc_server_defaults is an ALEPH configuration located under $alephe_root. This configuration defines the default value for the pc server. Following are the relevant entries for the Acquisitions GUI:

- **setenv acq_user_z71_sort_routine**

  Sorting routines for Acq Order Log and Action Log Messages Possible values:

  - 00-by the open date & hour of the transaction
  - 01-by sequence no. of the log

- 02-by sequence no. of the log. The log with the Z71-ACTION-DATE will always be sorted on top by Z71-ACTION-DATE.

- **setenv acq_user_z71_sort_order**

  Sorting order for Acq Order Log and Action Log Messages Possible values:

  - A - Ascending
  - D - Descending

- **setenv default_lock_period**

  Locked Acquisitions records are automatically unlocked after a period defined in this section.

  The period is defined in seconds. The default has been set to 300 seconds.

  Note

  The value entered in setenv default_lock_period is also shared by Items, Circulation and ILL records.

- **max_unit_price_diff_percent**
  This parameter defines the maximum percent difference between the Invoice Line Item's actual unit price and the order's estimated unit price. If the difference exceeds more than the defined percent, an alert message is displayed. For more details see section Line Item Invoice - Alert Message on page 39 of the System Librarian chapter.

- **acq_user_z75_sort_routine**

  Sorting order for the invoice lines that are displayed in order information screens. The following options may be used:

  - INV-1-A : Vendor/Invoice Number (Ascending)
  - INV-1-D : Vendor/Invoice Number (Descending)
  - INV-2-A : Date Range (Ascending)
  - INV-2-D : Date Range (Descending)
  - INV-3-A : Invoice Type (Ascending)
  - INV-3-D : Invoice Type(Descending)
  - INV-4-A : Related Budget (Ascending)
  - INV-4-D : Related Budget (Descending)
  - INV-5-A : Invoice Line Note (Ascending)
  - INV-5-D : Invoice Line Note (Descending)
  - INV-6-A : Invoice Number (Ascending)
  - INV-6-D : Invoice Number (Descending)

19 Setting the Local Currency

This scenario shows how to set USD (US dollars) as your local currency. In order to set the local currency, perform the following steps:

1. In the GUI's Acquisitions/Serials module, select the Administration tab. Select the Currency node. The Currency tab is displayed in the upper pane. Define USD as one of your currencies, and set its ratio to 1.000.

2. Go to the `./alephe` directory. Define USD as your local currency in `./alephe/aleph_start`.

   ```
   setenv local_currency   USD
   ```

3. After exiting the `aleph_start` file, run the command:

   ```
   source aleph_start
   ```

4. Restart pc server (UTIL W).

The system's local currency is now USD. The GUI user is unable to add, replace or delete the ratio (1.000) for USD.

20 Column Headings (pc_tab_col.lng)

`pc_tab_col.lng` defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (USM01) table pc_tab_col.lng using UTIL I/9 or the ALEPHADM module. For more information about pc_tab_col.lng, see the section in the ALEPH User Guide - General chapter - GUI Lists - Column Headings.

The following is a list of the Acquisitions windows that use `pc_tab_col.lng` for formatting data, and their identifiers (Column 1 in `pc_tab_col.lng`).

Identifier	**Acquisitions/Serials GUI Windows**
PC_ACQ_BUDGET	Order Budget *
PC_ACQ_LOGGER	Action Log Messages Order Log
PC_ACQ_GLOBAL_SEARCH	New and Cancelled Order Index
PC_ACQ_LIST	Order List *
PC_ACQ_INDEX	Index List
PC_ACQ_O_I_L	Order Invoice Lines
PC_ACQ_O_H_I_L	Invoice Line Items
PC_ACQ_O_A_L	Arrival List

PC_ACQ_CLAIM_L	List of Claims
PC_COM_CURRENCY_LIST	Currency List
PC_COM_BUDGET_LIST	Budget List
PC_COM_TRANSACTION	Transactions List of Budget
PC_COM_VENDOR_LIST	Vendor List
PC_ITEM_ITEM	Items List
PC_COM_D_TRIG_LIST	List of Triggers for Order
PC_ACQ_MULTI_ORDER	Multi-Order Index
PC_ACQ_OPAC_LIST	OPAC Request List
PC_ACQ_BULK_ORDER	Bulk Ordering List
PC_ACQ_BULK_ARRIVAL	Bulk Arrival List
PC_ACQ_BULK_RESULT	Bulk Arrival Result

* In this GUI table, an optional color/font can be used by the system for color/font differentiation between values in the same column. The alternative font and color can be defined in Column 8 and 9 of pc_tab_col.lng.

21 Client Setup (acq.ini)

The acq.ini file defines settings for the Acquisitions/Serials client. This chapter presents and explains the following sections of the acq.ini file.

21.1 ACQ.INI SETTINGS

- [OrderNumber]
- [Invoice]
- [Arrival]
- [NewCancelIndexList]
- [RfidMedia]
- [LibraryRelation]
- [Messages]
- [AdminTree]
- [UserList]

Explanations of other setting definitions, relating to more than one module, can be found in the *.Ini Files chapter of the General module of the ALEPH System Librarian's Guide.

[OrderNumber]

```
[OrderNumber]
EnablePrefix=Y
PrefixString=uarcv
```

The [OrderNumber] section defines the default order number prefix. It works together with the order number counter in UTIL G/2 (for example: last-order-no-uarcv).

[Invoice]

```
[Invoice]
CreateArrival=Y
SubscriptionDateRangeCheck=Y
ApprovalDep=APPROVAL 1
```

CreateArrival=Y
When a line item invoice is registered, this section determine if the user will be asked whether he would like to record the material as having arrived. If it is set to N, the question regarding the arrival of material will not pop up.

When the staff user chooses to register an arrival and the arrival record is created, the claim date of standing order only is updated by the system as follows:

Current date + Z68-MAX-NO-ARRIVAL-DAYS

SubscriptionDateRangeCheck=Y
This section determines the default value of the field: Check Subs. date overlap. This field appears in line item invoice of serials orders and standing orders. Possible values are:

Y - The check box is initially selected

N - The check box is initially clear

ApprovalDep=APPROVAL 1
This section determines the Approval Department field of the General Invoice.

[Arrival]

```
[Arrival]
RushNote=Y
LibraryNote=Y
ArrivalMessages=Y
```

RushNote=Y
When registering the arrival of a rush order, this section will determine whether to notify the user that this is a rush order.

LibraryNote=Y
When registering the arrival of an order which has a library note, this section will determine if the library note will pop up.

ArrivalMessages=Y
When the arrival of an order whose items have been requested is registered, this setting determines whether or not the user is notified that it is requested.

[NewCancelIndexList]

```
[NewCancelIndexList]
OrderType=M
MaterialType=
OrderGroup=
VendorCode=010
OpenDateFrom=0
OpenDateTo=0
RefreshFilter=Y
RefreshOnStart=Y
```

The [NewCancelIndexList] section defines the default parameters for the New and Cancelled Order List.

[RfidMedia]

```
[RfidMedia]
ActivateReader=Y
SuccessMessage=Y
```

The [RfidMedia] section is required when library items are RFID-tagged. For more information on using RFID equipment, please refer to the How-to document in the Ex Libris Documentation Center that is relevant to your RFID vendor (for example, for Bibliotheca, you would refer to *How to set up a BiblioChip® interface in ALEPH® 500 - 18.01*).

ActivateReader=Y
This variable determines whether or not the RFID Reader is updated when the relevant GUI actions are triggered.

SuccessMessage=Y
This variable determines whether or not a message indicating success is issued when the RFID Reader update action succeeds.

[LibraryRelation]

```
SecondBib=N
```

If there are two BIB libraries connected to the same ADM library, set the value Y (SecondBib=Y).
If there is only one BIB library, set N (default value).

[AdminTree]

DefaultTree=B

This variable sets the first branch that is opened when the Admin tab is clicked in GUI-ACQ/SERIAL.

The possible values are:

B (Budgets), V (Vendors), C (Currency), E (EDI Load Log), L (Action Log Messages), T (Triggers), R (Budget Transfer), J (File List), A (Batch Log), Q (Batch Queue), D (Print Daemon).

The default value is B (Budgets).

[UserList]

```
[UserList]
BorrowersOnly=N
FilterType=0
```

BorrowersOnly=N
Defines if the "Display Local Patron Only" check box in the Patron List is selected by default:

- If set to **Y**, only patrons with at least one local record are displayed. Whether or not patrons whose only local record is ALEPH are displayed depends on the setup of the user_list_aleph_local parameter in pc_server_defaults.

- If set to **N**, all patrons are displayed.

FilterType=0
The Patron List can be sorted according to one of the following parameters: Patron ID, Patron Name or Patron Barcode. This parameter sets the default value for the sorting of the Patron List.

- FilterType=1 sorts list by Patron ID
- FilterType=0 sorts list by Patron Name
- FilterType=2 sorts list by Patron Barcode

You can change the sort of the Patron List manually using the "Sort by" option in the Patron List in the Acquisition client.

22 Invoice Payment Status (tab48)

The administrative table tab48.lng (General Invoice Payment Status) controls the following functions:

- This table defines the General Invoice Payment Status (Z77-P-STATUS). The library can define its own payment status by setting this new table.

- Payment Status (Col.1 and Col.5: code and caption).

- Default Payment Status (Col.2) - This column defines the default status when a general invoice is created.

- Freeze Invoice (Col.3)- This column defines which statuses will freeze the invoice. This means that all operations in the General Invoice and related line items (except for changes in payment status) are not allowed.

- Permissions (Col.4) - This column defines which user_function routines should be used in addition to the standard checks for different statuses. Multiple Invoice Payment Statuses can use the same authorization. Following are "payment status" permissions of the alephe table: user_function.lng (these permissions are also used in tab48):

    - 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.

  Each library can add new permissions to `user_function` according to the new payment status. For example: the library defines in tab48, a new Payment Status: C - Check Before Paying (Col.1 and Col.5), a separate user authorization can be defined in `user_function.lng`:

  ACQ L Acquisitions INVOICE-HEAD-P-S-C L Update "check before paying" general invoice and/or its line invoices

  This user function can then be used in Col.4 of tab48.lng. Alternately, the library can use existing permissions (for example, ACQ/INVOICE-HEAD-P-S-Y) to define permission for payment status C. In this case, payment statuses Y and C will share the same user permission.

  The permissions that are defined in `tab48.lng` affect not only when the user manually tries to change the Invoice Payment Status, but also when the user

tries to change a payment status by running the Invoice Report service (acq-10).

23 Checksum of Vendor Additional Code Format (ABN)

Checksum of vendor additional code (Z70-ADDITIONAL-VENDOR-CODE) can be performed by setting the ADM table `tab_checksum`. This table is used for defining the checksum programs.

`vendor_abn` is the relevant program for performing ABN (Australian Business Number) checksum on Vendor Additional Code. If data is entered in the Vendor Additional Code field, then it must match the ABN check. Alternately, the field can be left empty (with no data).

Following is a sample setup of `tab_checksum`. This example shows how to perform checksum (valid and correct) of ABN:

```
      1                                2
!!!!!!!!!!!!!!!!!!!!!---!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CSV-VENDOR-ADD-CODE     chksm_valid_vendor_abn
CSCO-VENDOR-ADD-CODE    chksm_correct_vendor_abn
```

Note

The table's programs can be used for creation, validation and correction of item barcode, patron barcode, budget code prefix, vendor code prefix, ILL supplier prefix and vendor additional code.

24 Budget and Vendor Code Prefix

`tab_code_prefix` is an Administrative table that is used for defining prefixes for the budget code, vendor code and ILL supplier code. This table works in conjunction with the Administrative table: `tab_checksum`. If the checksum programs for vendor code, ILL supplier code and budget code are used in `tab_checksum`, then the check is compared to the prefixes that are defined in `tab_code_prefix`. Only a vendor, ILL supplier or budget that has one of the defined prefixes is allowed by the system.

For example: tab_checksum is set to perform a check of the vendor, supplier and budget code prefix:

```
! 1                                                              2
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-----------
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CSV-VENDOR-CODE                  checksum_valid_code_prefix
CSV-BUDGET-NUMBER               checksum_valid_code_prefix
```

The valid prefixes are defined in tab_code_prefix:

```
! 1
!!!!!
BUD-
VEN-
```

SUPP-

Only Vendor, ILL supplier and budget code, which has one of these prefixes, will be registered in the system. Any other prefix will be blocked and the record will not be registered.

25 Order Units

The Acquisitions system can be based on either order units or sublibraries, depending on the tab100 variable, USE-ORDER-UNIT. This variable determines whether the Acquisitions system works with order units or with sublibraries. It deals with the Budget's (Z76) order unit, the Vendor's (Z70) order unit, the order's (Z68) order unit and user-passwords (Z66).

If this variable is set to N, then each sublibrary is an order unit. If the variable is set to Y, then ordering units are defined in `tab_sub_library.lng` using sublibrary type 5. Addresses of ordering units are set in `$alephe_tab/tab_sub_library_address.lng`.

Following are special instructions and emphasis for a system that is set to work with order units (the tab100 variable USE-ORDER-UNIT is set to Y) in contrast to a system that is set not to work with order units (the tab100 variable USE-ORDER-UNIT is set to N).

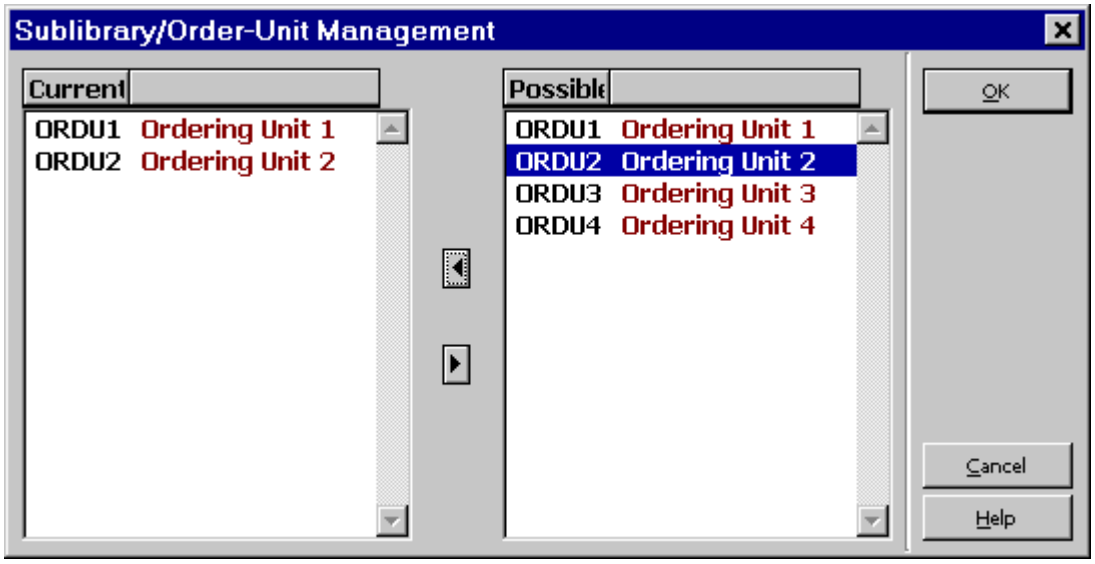
Order Form

If the system is set to work with order units (the tab100 variable USE-ORDER-UNIT is set to Y), a mandatory field appears in the order form: Order Information tab: Order Unit (Z68-ORDERING-UNIT). The Order Unit that is selected for the orders is checked against the order's vendor and the order's budget order unit. The user must assign to the order a vendor/budget that matches the order unit.

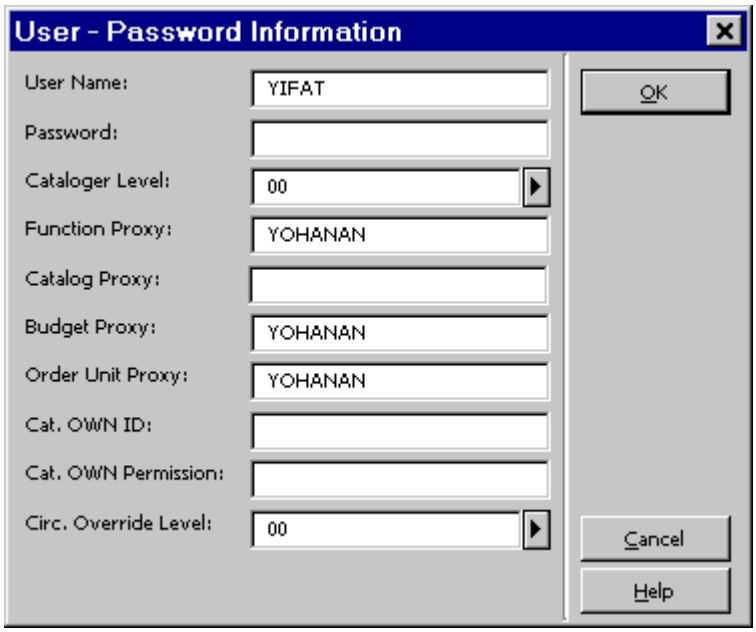
User Permissions

When you click the key icon and select Staff Privileges, the Staff Privileges window will appear. From this window, a button can be selected for Order Units/Sublibraries. This button can be labeled either "Order Units" or "Sublibraries". If your system is set to work with order units, the button should be named Order Units, and if it is set to work with sublibraries, it should be named Sublibraries.

The User's Order Units/Sublibraries must be defined in the Current list window; otherwise, the user will have no privileges for order units/sublibraries. It depends on the tab100 variable USE-ORDER-UNIT whether the list contains sublibraries or order units. For global permission, select USM50. Following is an example of user-sublibrary management of a system that uses order units:



In the User-Password Information window, there is an Order Unit/Sublibrary Proxy field (Z66-ORDER-UNIT-PROXY) that is used to redirect your acquisitions ordering unit/sublibraries permission to another user.

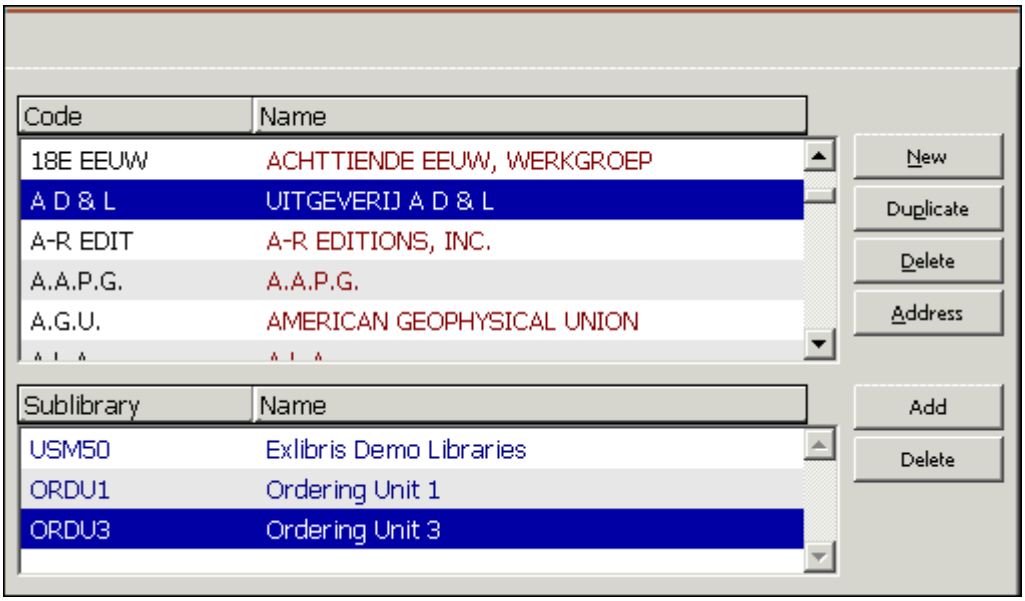


Vendors and Budgets

The ordering unit concept is also available for the vendor and budget record.

1. In an order unit system, the vendor/budget allowed sublibraries (Z602) will be order units.

2. If your system is set to work with two-level vendors (the variable TWO-LEVEL-VENDOR is set to Y), then the sub-level vendor record will be registered on the ordering unit/sublibrary depending on the variable USE-ORDER-UNIT. Following is an example of a vendor sub-level list of an order unit system:



3. Checks for vendor and budget usage permission are made against the Z68-SUB-LIBRARY or Z68-ORDERING-UNIT depending on the USE-ORDER-UNIT setup.

Order Unit Address

If the USE-ORDER-UNIT variable is set to Y in GUI forms and service forms, the sublibrary address will be the Z68-ORDERING-UNIT address and not the Z68-SUB-LIBRARY address. If USE-ORDER-UNIT is set to N (so that the Z68-ORDERING-UNIT is empty), the address will be the Z68-SUB-LIBRARY address. Addresses of ordering units and sublibrary addresses are set in `$alephe_tab/tab_sub_library_address.lng`.

GUI List

The Order Unit column can be displayed in the following GUI lists.

- Order list for ADM record - menu PC_ACQ_LIST in `pc_tab_col.lng`.
- New and cancelled order index - menu PC_ACQ_GLOBAL_SEARCH in `pc_tab_col.lng`.
- Multi order index menu PC_ACQ_MULTI_ORDER in `pc_tab_col.lng`.

The Order Unit field can also be displayed in the Order Details window using the z68-ordering-unit tag in the order-info HTML template.

window.dat Setup

If your system is using order units, several buttons, fields and screen titles will need to be changed. These changes of button labels, field names and screen titles are made in the following pc files:

```
acq\tab\lng\window.dat,
alephcom\tab\lng\window.dat.
```

Screens, fields and buttons that appear in more than one module are dealt with in

ALEPHCOM. Those that appear in a single module are dealt with in the module's specific window.dat file.

Example of renaming a button label:
The Sublibrary button in Budget List must be renamed Order Units. To do this, change the following line in `alephcom\tab\lng\window.dat`:

From:

```
VendorListAdmin.Btn.SubLibrary      &Sublibrary
```

To:

```
VendorListAdmin.Btn.SubLibrary      &Order Units
```

Example of renaming a window title:
The Sublibrary Management window of the Vendor/Budget List should be renamed Order Units Management. To do this, change the following line in `alephcom\tab\lng\window.dat`:

From:

```
RecordSubLibrary.Title           Sublibrary Management
```

To:

```
RecordSubLibrary.Title           Order Units Management
```

Example of renaming a field:
The Sublibrary Proxy field in the User - Password Information window should be renamed Order Unit Proxy. To do this, change the following line in `alephcom\tab\lng\window.dat`:

From:

```
PrivilegesUpdate.Static.OrderUnitProxy  #Order Unit/Sublibraries \n
Proxy:
```

To:

```
PrivilegesUpdate.Static.OrderUnitProxy  #Order Unit Proxy:
```

Services
Order Unit is used as a filter in the following services:

- Claim Report and Letters for Monograph Orders (acq-12)
- Claim Report for Serials Orders (acq-19)
- Claim Report for Standing Orders (acq-11)
- Send List of Orders to a Vendor (acq-14)
- Order-Log Report (acq-22)
- Subscription Renewal Letters (acq-23)

- General Retrieval Form (acq-02-01)
- Partially-Filled Monograph Orders (acq-02-02)
- Monograph Orders - Status "NEW" (acq-02-03)
- General ADM and Orders Creation (acq-24)

If your system is set not to use order units, the Order Unit field should become a hidden field.

Example from the service: Send List of Orders to a Vendor (acq-14)
If your system is set to work with order units, set the Order Unit in the file `./alephe/pc_b_eng/p-acq-14.xml` as follows:

```
<control>
        <expand_check box>
                <label>Order Unit</label>
                <source>ORDER-UNIT</source>
                <argname>F04</argname>
        </expand_check box>
</control>
```

If your system is set not to work with order units, set the Order Unit in the file `./alephe/pc_b_eng/p-acq-14.xml` as follows:

```
<control>
    <hidden>
        <expand_check box>
                <label>Order Unit</label>
                <source>ORDER-UNIT</source>
                <argname>F04</argname>
        </expand_check box>
    <hidden>
</control>
```

26 Line Item Invoice - Alert Message

When a new invoice's line item is created or an existing one is modified, and the actual unit price exceeds the estimated unit price by more than a defined percent, a non-recurrent alert message is displayed (upon clicking the **OK** button).

The actual unit price is calculated by dividing the "Total Amount" (Z75-I-TOTAL-AMOUNT) by the "Number of Units" (Z75-I-NO-UNITS). Both fields are part of the GUI-ACQ-Invoice Line Item window.

The estimated unit price is calculated by dividing the "Final Price" of the order (Z68-E-PRICE) by the "Number of Units Ordered" (Z68-NO-UNITS). Both fields are part of the GUI-ACQ-Order Form window-Quantity and Price tab.

The difference in percents between the unit prices takes into account the currency of the order (Z68-E-CURRENCY) and of the invoice (Z77-I-CURRENCY). If they differ, the comparison is made in local currency units.

The maximum percent difference is determined by setting the value of the environment variable "max_unit_price_diff_percent" in the file `./alephe/pc_server_defaults`, for example:

```
setenv max_unit_price_diff_percent      10
```

The alert message is located in the file `$aleph_root/error_lng/pc_acq_c0513`:

```
1011 0000 L Actual unit price ($1 $2) exceeds the estimated unit
price ($3 $4) by more than $5 percent. Continue?
```

Note

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.

27 Order Bar Search Options

The list of search options is defined in the ORDER-SEARCH menu of `pc_tab_exp_field.lng`.

Example:

```
ORDER-SEARCH    L Order number                  ON
ORDER-SEARCH    L Group order number            GON
ORDER-SEARCH    L Vendor reference              RF
ORDER-SEARCH    L ISBN                          ISBN
```

This allows you to configure fewer entries for the Order bar as opposed to the Order Index that uses the ACQ_INDEX_TYPE menu.

In addition, it is possible to find an entry through the Search function.

```
ORDER-SEARCH       L BIB Sys No.         BIB-SYS
ORDER-SEARCH       L ADM Sys No.         ADM-SYS
ORDER-SEARCH       L BIB Title           BIB-TIT
```

BIB-SYS - look for system number in the first BIB relation of connected library.
ADM-SYS - look for system number in connected library.
BIB-XXX - Does a CCL search (XXX =)

In the case of BIB-XXX there can be multiple hits. If so, there is an option to view the set in the Search function via confirmation messages.

Example of adding a BIB search value:

```
ORDER-SEARCH          L BIB Author            BIB-AUT
```

Note

In this example, AUT must be an index code that is used by the system.

The option to define the BIB search option by values defined in `tab_acq_index` is also valid for the order Search bar and those values can be added to the ORDER-SEARCH menu.

28 Sending a List of Orders Sent via EDI to a Vendor including More than One Sublibrary

The Send List of Orders to a Vendor (acq-14) service enables a staff user to send a vendor a letter with a list of orders. All orders whose 'Order Delivery Type' is LI (List) and whose status is RSV (Ready Send to Vendor) will be included in this service.

To enable this service to include orders of all sublibraries, or orders of more than one sublibrary sent by EDI check, you must set up a line in the `./xxx50/tab/tab35` configuration table for the active ADM library code (for example, USM50), from which the service is being submitted. For example:

```
! 1             2                    3           4           5
6
!            7                                 8
!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!-
!!!!!!!!!!!!!!!!!!!!-!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
USM50 AACU                 EXL-SER               091
IL 111 222 331                 melvyl.dewey@exlibris.edu
```

29 Setting Up the Budget List

You can decide that the Budget List will contain only those budgets with assigned Order Units/Sublibraries(Z602) which match the user's assigned Order Unit/Sublibrary (Z602).

The setup of the BUDGET-PER-ORD-UNIT variable in tab100 determines the filtering mechanism of the Budget List.

Note

If the BUDGET-PER-ORD-UNIT variable is set to Y and the Search Modes are Group or Wildcard, when one budget in the group is denied, all budgets included in the group are not displayed.

To enable the display of all budget information in the lower pane when “View Authorized” is not checked, use VIEW-NON-AUTHORIZED variable in tab100.

30 Setting Up Acquisitions for EDI

Edit tab100 for EDI - set the following variables:

- EDI-OUT-VENDOR-NOTE
- EDI-OUT-LIBRARY-NOTE
- EDI-OUT-PRICE-NOTE
- EDI-OUT-QUANTITY-NOTE

to determine whether or not the following notes will be added to an outgoing EDI message:

- Vendor Note
- Library Note
- Price Note
- Quantity Note

To set EDI claims codes for a serials claim, define the following lines in ./xxx50/tab/pc_tab_exp_field.lng SERIAL-CLAIM-FORMAT:

```
SERIAL-CLAIM-FORMAT L EDI claim - missing issue                90
SERIAL-CLAIM-FORMAT L EDI claim - wrong issue supplied         91
SERIAL-CLAIM-FORMAT L EDI claim - damaged copy                 92
SERIAL-CLAIM-FORMAT L EDI claim - too many copies received     93
SERIAL-CLAIM-FORMAT L EDI claim - insufficient copies received 94
SERIAL-CLAIM-FORMAT L EDI claim - other reason                 95
```

31 Setting up Vendor Sharing

The following setup is required for sites that share a single vendor table for multiple ADM libraries.

Steps

1. From each ADM library's `file_list`, remove LS (logical synonyms) for Z70 and Z72. Z70 and Z72 should only be defined in the `file_list` of the shared vendor library, for example, XXX50.
2. In the `tab100` of each ADM library, define VENDOR-SHARING=**1**.
3. In `./alephe/aleph_start`, define the following variable:

   ```
   setenv vendor_library XXX50
   ```

4. In the Acquisitions GUI, run the Build Word Indexes for a Record (manage-111) service to create keywords for the shared vendor library.

32 Acquisitions Values of aleph_start

setenv unique_order_number_2 Y

This environment variable is used for 'Generic Vendor Records Loader (file-96)'. It defines whether to ignore Z68-ORDER-NUMBER-2 variable which is set in the

service's configuration table. This is done in order to avoid duplicates. When it is set to 'Y', any value for Z68-ORDER-NUMBER-2 defined in the configuration table will be ignored and it will not be added to the new order record.

setenv vendor_library XXX50

Used to define the shared vendor library only in a multi-ADM environment which is using the shared vendor model.

33 Load Invoice and Line Items (acq-32)

This chapter describes how to load new general invoice and line items using the Acquisitions batch service, Load Invoice and Line item (acq-32).

The following is a general description of the batch activities:

1. The batch service applies to an input file (in an xml format) which is placed in the ADM library/scratch directory and is populated with a pre-defined XML tags including invoices and line items information.

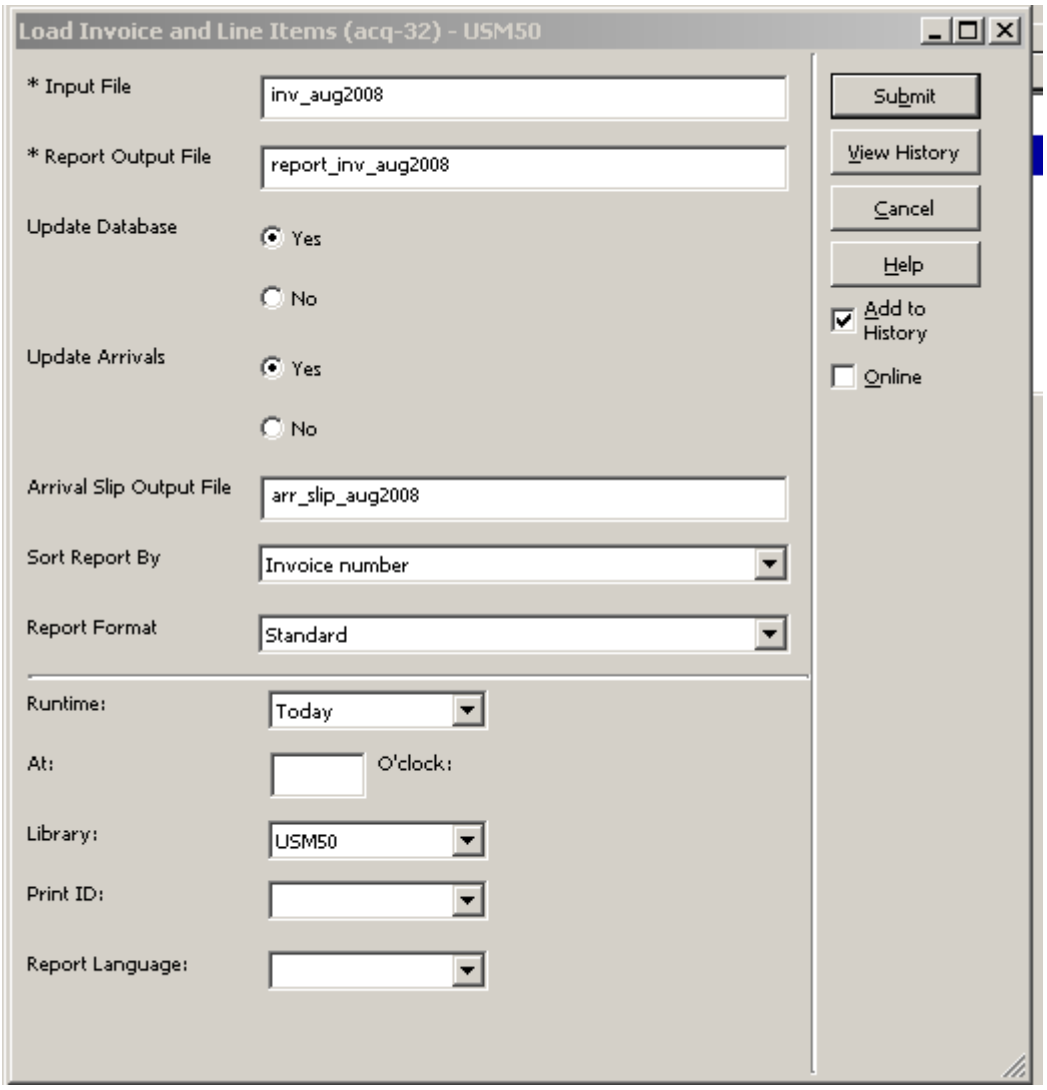
2. Based on the input data, new general invoice and line item records are created in Aleph and all necessary actions are performed.

3. The batch service produces an output report detailing the successful and failed actions of the batch execution.

4. For Monograph and standing Orders, the batch service enables the registering of material arrival and the production of arrival slips.

33.1 The Batch Service Interface

The batch service "Load Invoice and Line item (acq-32)" is accessed via the GUI-Acquisition/Serials module from **Services → Other**.



- **Input file** – Mandatory. The name of the xml file that contains the invoice data. The file should be located in the ADM library's scratch directory (./xxx50/scratch).

- **Report Output File** – Mandatory. The name of the file in which you want to save the output report.

- **Update Database** – Selecting "Yes" generates a report output file, creates new general invoices and line items data in Aleph, and updates all other corresponding Aleph data (such as orders and budget transactions). Selecting "No" generates a report output file but without updating the database.

- **Update Arrivals** – Selecting "Yes" registers a new arrival record (Z78) for monograph and standing orders. In this case, the order's arrival status for monograph orders is changed to "Complete". Selecting "No" does not update arrival information.

- **Arrival Slips Output File** – This field is relevant only when Update Arrival is "Yes". If you want to print arrival slips (acq-arrival-slip-00.xsl), enter the name of the file in which you want to save the output arrival slips.

- **Sort By** – The parameter by which the output report is sorted. The options are:

    - **Invoice Number** – Ascending. Within a single general invoice, the sort is by line item sequence number.
    - **Additional order number 1** (also known as: common order number) – ascending
    - **Order number** – Ascending

    Failed/Update Status – The report is sorted so that the failed transactions are listed first and the successful transactions are listed afterward.

- **Report Format** – Select from a list of predetermined report formats (template file name: acq-invoice-load.xsl).

33.2 The Input File

The input file should be built with xml scheme according to the following structure:

```
<?xml version="1.0"?>
<general-invoice-list>
 <general-invoice>
      <invoice-number>12381</invoice-number>
      <vendor-code>21NM</vendor-code>
      <orig-invoice-number></orig-invoice-number>
      <type>REG</type>
      <status>REG</status>
      <inv-credit-debit>D</inv-credit-debit>
      <currency>USD</currency>
      <currency-ratio></currency-ratio>
      <inv-net-amount>00000000000000</inv-net-amount>
      <inv-ship-amount>00000000000000</inv-ship-amount>
      <inv-over-amount>00000000000000</inv-over-amount>
      <inv-insu-amount>00000000000000</inv-insu-amount>
      <inv-disc-amount>00000000000000</inv-disc-amount>
      <inv-total-amount>00000000050000</inv-total-amount>
      <inv-date></inv-date>
      <ship-date></ship-date>
      <inv-note>Invoice for your orders from Aug 2008</inv-note>
      <pay-date>00000000</pay-date>
      <pay-check-no></pay-check-no>
      <pay-amount>00000000000000</pay-amount>
      <pay-status>N</pay-status>
      <pay-approval-dept></pay-approval-dept>
      <pay-approval-number></pay-approval-number>
      <inv-vat-code></inv-vat-code>
      <inv-vat-method>I</inv-vat-method>
```

```
<vat-per-line></vat-per-line>
<inv-vat-amount></inv-vat-amount>
<vat-receiver></vat-receiver>
<line-item>
    <order-number>51696</order-number>
    <add-order-number-1></add-order-number-1>
    <object-code></object-code>
    <line-credit-debit>D</line-credit-debit>
    <line-net-amount>00000000000000</line-net-amount>
    <line-total-amount>00000000030000</line-total-amount>
    <line-vat-code></line-vat-code>
    <line-vat-amount>00000000000000</line-vat-amount>
    <number-units></number-units>
    <budget-code></budget-code>
    <line-note>Charge for paper. Map is free</line-note>
    <date-from>00000000</date-from>
    <date-to>00000000</date-to>
    <date-range>N</date-range>
</line-item>
<line-item>
    <order-number>51697</order-number>
    <add-order-number-1></add-order-number-1>
    <object-code></object-code>
    <line-credit-debit>D</line-credit-debit>
    <line-net-amount>00000000000000</line-net-amount>
    <line-total-amount>00000000010000</line-total-amount>
    <line-vat-code></line-vat-code>
    <line-vat-amount>00000000000000</line-vat-amount>
    <number-units></number-units>
    <budget-code></budget-code>
    <line-note></line-note>
    <date-from>00000000</date-from>
    <date-to>00000000</date-to>
    <date-range>N</date-range>
</line-item>
<line-item>
    <order-number></order-number>
    <add-order-number-1></add-order-number-1>
    <object-code></object-code>
    <line-credit-debit>D</line-credit-debit>
    <line-net-amount>00000000000000</line-net-amount>
    <line-total-amount>00000000010000</line-total-amount>
    <line-vat-code></line-vat-code>
    <line-vat-amount>00000000000000</line-vat-amount>
    <number-units></number-units>
    <budget-code></budget-code>
    <line-note></line-note>
    <date-from>00000000</date-from>
    <date-to>00000000</date-to>
    <date-range>N</date-range>
```

```
    </line-item>
 </general-invoice>
</general-invoice-list>
```

The XML input file should contain the compound elements:

- <general-invoice-list>
- <general-invoice>
- <line-item>

The XML file is started and ended by the <general-invoice-list> tag.
The XML file can contain one or more general invoices <general-invoice>.
Each general invoice can contain one or more line items <line-item> – up to 500 line items for an invoice.

The <general-invoice> tags:

vendor-code
Input value is mandatory. The code of the vendor that supplies the invoice (Z77-VENDOR-CODE). The input must be registered in Aleph Vendor database (Z70). If input is missing/non-valid or vendor is Not-Active; the general invoice and related line items are not loaded and the output file reports an error.

invoice-number
Input value is mandatory. Up to 15 characters. The general invoice number (Z77-INVOICE-NUMBER). The system checks in the database and in the input file that the invoice number does not exist for the same vendor. In case the invoice number is already registered for the same vendor the general invoice and related line items are not loaded and the output file reports an error.

orig-invoice-number
Up to 15 characters. May contain the invoice number to which this invoice refers to (Z77-ORIG-INVOICE-NUMBER).

type
3 characters. Type of invoice (Z77-I-TYPE). Value should be one of the codes defined in ACQ_INVOICE_TYPE menu of ./xxx50/tab/pc_tab_exp_field.lng. E.g.: REG. If input code is not recognized, the general invoice is loaded and the inconsistency is reported in the output report.
If <type> input is missing or blank, the system populates in Z77-I-TYPE the default value as defined in ./xxx50/tab/tab100: DEFAULT-GEN-INV-TYPE.

status
3 characters. Status of the invoice (Z77-I-STATUS). Value should be one of the codes defined in ACQ_INVOICE_STATUS menu of ./xxx50/tab/pc_tab_exp_field.lng. E.g.: REG. If input code is not recognized, the general invoice is loaded and the inconsistency is reported in the output report.
If <status> input is missing or blank, the system populates in Z77-I-STATUS the default value as defined in ./xxx50/tab/tab100: DEFAULT-GEN-INV-STATUS.

inv-credit-debit

1 character: D (debit) or C (credit). Indicates whether this is a debit invoice or a credit invoice (Z77-I-CREDIT-DEBIT). If input is missing or non-valid, the system populates D.

currency
Input value is mandatory. Currency code used for the general invoice (Z77-I-CURRENCY). In case the input currency code is not set in Aleph data; the general invoice and related line items is not be loaded and the output file reports an error.
The system also checks for the CHECK-INVOICE-CURRENCY flag in ./xxx50/tab/tab100.
If CHECK-INVOICE-CURRENCY=Y and the currency code is not one of the vendor's currency, then the general invoice is loaded and the inconsistency is reported in the output report.
If CHECK-INVOICE-CURRENCY=N, the system does not apply this check.

currency-ratio
12 digits, last six digits are decimal (e.g.: 000004500000 represent 4.5). Explicit currency ratio for the general invoice (Z77-I-CURRENCY-RATIO). If the input is missing or invalid, the system applies the currency ratio as set in Aleph data (Z82).

inv-net-amount
14 digits, last two digits are decimal (e.g.: 00000000010050 represent 100.5).The general invoice net amount (Z77-I-NET-AMOUNT). If input is missing or non-valid, the system populates zeros.

inv-ship-amount
14 digits, last two digits are decimal. The general invoice ship amount (Z77-I-SHIP-AMOUNT). If input is missing or non-valid, the system populates zeros.

inv-over-amount
14 digits, last two digits are decimal. Invoice overhead amount (Z77-I-OVER-AMOUNT). If input is missing or non-valid, the system populates zeros.

inv-insu-amount
14 digits, last two digits are decimal. Invoice insurance amount (Z77-I-INSU-AMOUNT). If input is missing or non-valid, the system populates zeros.

inv-disc-amount
14 digits, last two digits are decimal. Invoice discount amount (Z77-I-DISC-AMOUNT). If input is missing or non-valid, the system populates zeros.

inv-total-amount
14 digits, last two digits are decimal. Invoice total amount (Z77-I-TOTAL-AMOUNT). If input is missing, or non-valid, Z77-I-TOTAL-AMOUNT is computed from sum of Z77-I-NET-AMOUNT, Z77-I-SHIP-AMOUNT, Z77-I-OVER-AMOUNT, and Z77-I-INS-AMOUNT minus Z77-I-DISC-AMOUNT.

If Z77-I-TOTAL-AMOUNT is computed as zeros, the general invoice is loaded and the inconsistency is reported in the output report.

inv-date

YYYYMMDD. E,g. 20080825 represent 25-Aug-2008. The invoice date (Z77-I-DATE). If input is missing or non-valid, the system populates current-date.

ship-date
YYYYMMDD (Z77-I-SHIP-DATE). Date invoice was sent by the vendor to the library. If input is missing or non-valid, the system populates current-date.

inv-note
Up to 60 characters. Free-text note (Z77-I-NOTE).

pay-date
YYYYMMDD (Z77-P-DATE). Date invoice was paid.
If input is missing or non-valid, the system populates zeros.
If Z77-P-STATUS is "P" (paid) but Z77-P-DATE is zeros, the general invoice is loaded and the inconsistency is reported in the output report.

pay-check-no
Up to 15 characters. The check number that was used for paying the invoice (Z77-P-CHECK-NO).
If general invoice in not paid (Z77-P-STATUS is not "P") but the check number is populated (Z77-P-CHECK-NO is not blank), the general invoice is loaded and the inconsistency is reported in the output report.

pay-amount
14 digits, last two digits are decimal. The amount paid for this invoice (Z77-P-AMOUNT). If input is missing or non-valid, the system populates zeros.

pay-status
1 character. The general invoice's payment status (Z77-P-STATUS).The value should be one of the codes defined in ./xxx50/tab/tab 48. E.g.: N (Not paid) or P (Paid).
If payment status is Paid (Z77-P-STATUS=P), then the payment date (Z77-P-DATE) and payment amount (Z77-P-AMOUNT) cannot be zeros.
If the input is missing or non-valid, the system populates Z77-P-STATUS as follows: If Z77-P-AMOUNT and Z77-P-DATE are populated (not zeros), the value P (Paid) is set in Z77-P-STATUS. If Z77-P-AMOUNT and Z77-P-DATE are zeros, the value N (Not Paid) is set in Z77-P-STATUS.

pay-approval-dept
Up to 20 characters. The department responsible for processing the approval of the invoice payment (Z77-APPROVAL-DEPARTMENT).
If <pat-approval-dept > is set in the input file, it must match one of the codes defined in ACQ_APPROVAL_DEPARTM menu of ./xxx50/tab/pc_tab_exp_field.lng.
If the input is not-valid, the general invoice is loaded and the inconsistency is reported in the output report.

pay-approval-number
Up to 20 characters. Payment approval number assigned to the invoice (Z77-APPROVAL-NUMBER).

inv-vat-code
Up to 5 characters. Invoice's VAT code (Z77-VAT-CODE).

If <inv-vat-code> is set in the input file, it must match one of the codes defined in./xxx50/tab/tab_vat_percent.lng. If the input is not-valid, the general invoice is loaded and the inconsistency is reported in the output report.

inv-vat-method
1 character. Indicates the VAT calculation method (Z77-VAT-METHOD).
Values can be I (Inclusive- VAT amount is already included in total amount) or E (exclusive- amount is added to total amount). If the input is missing or non-valid, the system populates "I" in Z77-VAT-METHOD.

vat-per-line
1 character: Y (Yes) or N (No). Indicates whether the VAT amount is set in the general invoice or calculated from VAT amount of related line items (Z77-VAT-PER-LINE). Value N (No) indicates that the VAT amount of the general invoice is fixed. VAT values of all attached line items are calculated from Z75-I-TOTAL-AMOUNT according to the VAT percent of the general invoice.
Value Y (Yes) - The VAT amount of the general invoice is always zeros in the database. It is calculated on the fly from the VAT amounts of all attached line items. Line item VAT amount can be set individually for each line item.
If input is missing or non-valid, the system populates N in Z77-VAT-PER-LINE.

inv-vat-amount
14 digits, last two digits are decimal (e.g.: 00000000001000 represent 10). Invoice's VAT amount (Z77-VAT-AMOUNT). If input is missing or non-valid, the system populates zeros.

vat-receiver
Up to 40 characters. The body that receive the VAT amount (Z77-VAT-RECEIVER).

The <line-item> tags:

order-number
Up to 30 characters. The number of the order (Z68-ORDER-NUMBER) for which the invoice line item is refer (Z68-ORDER-NUMBER).
If <order-number> is missing, the system loads a line item that is not related to an order.
If <order-number> tag is populated, the system checks that there is such order in Aleph data. If a matching order record cannot be found, the specific line item is not loaded and the output file reports an error.

add-order-number-1
Up to 30 characters. The additional order number 1of the related order's (also known as "Common order Number", Z68-ORDER-NUMBER-1). This input tag can be used for matching the order with the invoice line item. It is not populated into Aleph's order database. If the input code is not found in Aleph data, then the line item is loaded and the inconsistency is reported in the output report. This tag is used only after the order-number has been matched to a database as an additional verification – not as a matching identification.

object-code
Up to 5 characters. The line item's object code (Z75-I-OBJECT-CODE).

If <object-code> is set in the input file, its value is checked against codes defined in OBJECT-CODE menu of ./xxx50/tab/pc_tab_exp_field.lng.
The USE-OBJECT-CODE flag in ./xxx50/tab/tab100 is checked. If USE-OBJECT-CODE=Y, then the line item object code (Z75-I-OBJECT-CODE) should match the budget object code (Z76-OBJECT-CODE).
If the input code is not valid, then the line item is loaded and the inconsistency is reported in the output report.

line-credit-debit
1 character: D (debit) or C (credit). Indicates whether this is a debit line item or a credit line item (Z75-I-CREDIT-DEBIT). If the input is missing or non-valid, the system populates the value D.

line-net-amount
14 digits, last two digits are decimal (e.g.: 00000000010050 represent 100.5). Line item net amount in the vendor's currency (Z75-I-NET-AMOUNT). If the input is missing or non-valid, the system populates zeros in Z75-I-NET-AMOUNT.

line-total-amount
14 digits, last two digits are decimal. Line item's total amount in the vendor's currency (Z75-I-TOTAL-AMOUNT). If the input is missing or non-valid, Z75-I-TOTAL-AMOUNT is computed from sum of Z75-I-NET-AMOUNT plus added charges. (Added charges are not stored in the database. The added charges value is calculated based on the invoice added charges). If Z75-I-TOTAL-AMOUNT is computed as zeros, then the general invoice is loaded and the inconsistency is reported in the output report.

line-vat-code
Up to 5 characters. Line VAT code (Z75-I-VAT-CODE). Relevant only if <vat-per-line> of the general invoice (Z77-VAT-PER-LINE) is set to 'Y". Value should be one of the codes defined in./xxx50/tab/tab_vat_percent.lng. If the input code is not valid, the line item is loaded and the inconsistency is reported in the output report.

line-vat-amount
14 digits, last two digits are decimal. Line item VAT amount (Z75-I-VAT-AMOUNT). If <line-vat-amount> input is zeros and the VAT code mechanism is used (Z75-I-VAT-CODE), then this field is automatically calculated by the system according to the definitions of ./xxx50/tab/tab_vat_percent.lng table. If the Z77-VAT-PER-LINE field of the general invoice is set to 'N', then the Z75-I-VAT-AMOUNT of all invoice attached line items are calculated from Z75-I-TOTAL-AMOUNT according to the VAT percent of the general invoice.

number-units
5 digits. (e.g 00005 represents 5). Number of units being charged for by this invoice line item (Z75-I-NO-UNITS). If <number-units> input is zeros or non-valid, the system populate Z75-I-NO-UNITS by taking the Z68-NO-UNITS field of the order and subtracting the Z75-I-NO-UNITS of other Z75 linked to the same order. If line item is not related to an order, the system populates 00000 in Z75-I-NO-UNITS.

budget-code

Up to 50 characters. This tag can contain the budget code for which the line item amount is debited (Z601-BUDGET-NUMBER). The system performs the same checks the same as if budget is manually assigned to cover an invoice line item (budget code, monetary coverage, dates, etc). If ./xxx50/tab/tab10 flag CHECK-ORDER-BUDGET=Y and the budget code is not valid or the check is failed; the line item is loaded without creating a budget transaction; the inconsistency is reported in the output report.
If <budget-code> is empty, the system attempts to create budget transactions based on the orders' encumbered budget (if it exists).

line-note
Up to 200 characters. Free-text note (Z75-I-NOTE).

date-from
YYYYMMDD. E,g. 20080101 represent 1-Jan-2008**.** Starting date of the period for which the Serial order or Standing order is charged for (Z75-I-DATE-FROM).
If the input is not relevant or not valid, the system populates zeros.

date-to
YYYYMMDD. E,g. 20081231 represent 31-Dec-2008**.** Ending date of the period for which the Serial order or Standing order is charged (Z75-I-DATE-TO).
If the input is not relevant or not valid, the system populates zeros.

date-range
1 character: Y (Yes) or N (No). Defines whether the Z75-I-DATE-FROM and Z75-I-DATE-TO are checked against all the line items records (Z75) attached to the same order record to determine whether there is a date overlap (Z75-I-DATE-RANGE). Relevant only for Serials and Standing orders. If set to 'Y' then Z75-I-DATE-FROM and Z75-I-DATE-TO cannot be zeros. If input is missing or non-valid, the system populates N.

33.3 Service Execution – A Successful Update

The load of a general invoice and line items affects the following acquisitions components as if a manual creation has been done by an operator.

- General invoice
- Invoice Line items
- Budget Transactions
- Arrivals
- Orders

General invoice (Z77) and Line Items (Z75)
New General invoice/s and line items records are created based on information supplied in the input XML file.

Budget Transactions (Z601)
In case the XML input file contains <budget-code> tags in the <line-item> section, the data is checked and new budget transaction of type INV (invoice) are registered.

If <budget-code> is empty and the system is set to work with budget control (tab100 flag: CHECK-ORDER-BUDGET=Y), the budget which is already registered for the order is used.

Arrivals Z78
If batch is run with "Update Arrivals – Yes" then 'Arrival" records are created for the related monograph and standing orders.
The number of units arrived (Z78-NO-UNITS-ARRIVED) is set based on the number of unit invoiced (as Z75-I-NO-UNITS). If Z75-I-NO-UNITS is 00000, the system does not create Z78 arrival record for the related line item.
Arrival action might also affect the related item's process status of the related item records (Z30-ITEM-PROCESS-STATUS, depending on ./xxx50/tab/tab42 setup).

Orders Z68
When a line item which is related to an order is created, then monograph orders (Z68-ORDER-TYPE=M) is updated as follows:

- Order Invoice Status (Z68-INVOICE-STATUS) is set to C (Complete).
- If the batch is run with "Update Arrivals - Yes"; the Monograph Order Arrival status (Z68-ARRIVAL-STATUS) is set to C (Complete).
- If Order Invoice Status and Order arrival Status (Z68-INVOICE-STATUS and Z68-ARRIVAL-STATUS) are both C (Complete) and Invoice Payment Status (Z77-P-STATUS) is P (Paid), then the Order Status (Z68-ORDER-STATUS) is set to CLS (Closed).

33.4 The Service Output Files

The "Load Invoice and Line item (acq-32)" produces two output printed files:

- The Load Invoice Output report that summarize all batch activities (template name: acq-invoice-load.xsl)
- File containing "arrival slips" (template file name: acq-arrival-slip-00.xsl) per arrival. The arrival slips are produced only if the batch service field "Update Arrival" field is Yes and the "Arrival Slips Output File" is populated.

The Load Invoice Output Report

- The standard report (file name ./form_lng/acq-invoice-load.xsl) is in a grid format and includes a line for each general invoice that is handled and a line for each line item that is handled. The translation file (./form_lng/ acq-invoice-load.trn) converts the Aleph data into readable information.

- The report format includes information for general invoice, line items, and budget transactions. The standard report offers the following data in a grid structure: Invoice No., Vendor, Success/Fail, Doc No., Order Number, Additional Order Number 1, Budget Number, and Info and Errors.
- The Info and Errors column reports problems that occurred during the load action. This column contains Error information for lines which failed to be loaded and Info details for lines which were loaded but a problem was detected.

- The system librarian may customize the standard report (acq-invoice-load.xsl) by adding any other required info from the General Invoice (Z77), Line Items (Z75), Order (Z68), and Arrival (Z78).

33.5 User Privileges

In order to activate the Load Invoice and Line item (acq-32) service, the user must be assigned with the following ADM library staff privilege:

Acquisitions/Invoicing Services/ACQ-32 Load Invoice and line

Relevant entry in ./alephe/tab/user_function.lng

```
INVOICING-S             L Invoicing Services                ACQ-32
L ACQ-32 Load Invoice and line items
```

Beside the above user access right, there are no further user permission checks.

33.6 Statistics Report

The acq-32 batch service generates statistics information which can be retrieved using the TCO Batch Summary Report (sys-90). The summary includes the following statistics information:

- Total number of general invoices included in the input file
- Total number of line items included in the input file
- Number of general invoice lines that were successfully loaded
- Number of line items that were successfully loaded

34 Vendor Name in Japanese: Kanji and Katakana

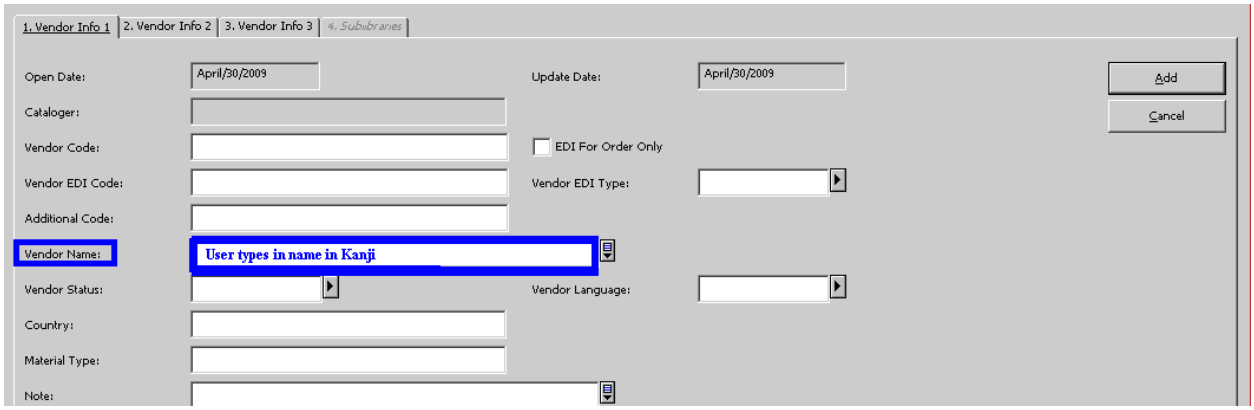
This chapter describes how to perform the following functions when the vendor name is in Kanji and Katakana:

- Store vendor name in Kanji and Katakana
- Search for vendor name in Kanji and Katakana
- Sort vendor name list in Katakana

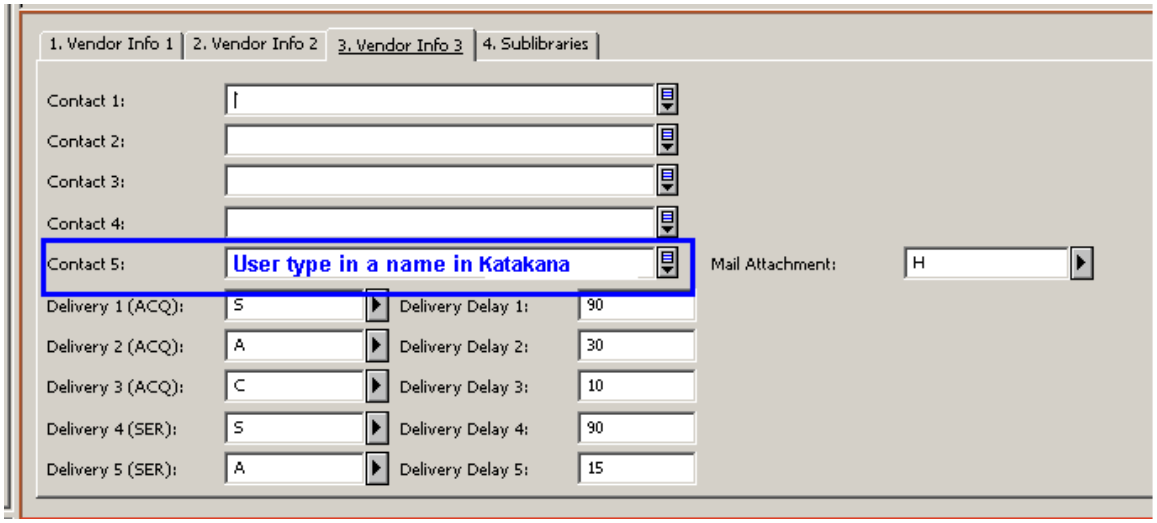
34.1 Storing Vendor Name in Kanji and Katakana

The vendor record can maintain vendor names in both Kanji and Katakana.

Kanji – The vendor name is stored in the Vendor Record in the Name field, Z70-VENDOR-NAME.



Katakana – The vendor name is stored in the Vendor Record in the Contact 5 field, Z70-CONTACT-5.



The Vendor Name (Kanji) is a mandatory field. User cannot register a vendor record without populating Vendor Name field.

The Contact-5 field (Katakana) can become a mandatory field by setting up a system variable located in ./alephe/alephe_start:

```
setenv   JAPANESE_ENABLED   TRUE
```

This variable can be set with the values FALSE (default) or TRUE.
Setting the variable with TRUE enables special Japanese features such as defining the vendor's contact-5 field (Katakana vendor name) as a mandatory field.
After setting this variable, users cannot register or update a vendor record without populating the Vendor's Contact 5 field. If this field is missing, the following error message is displayed: "Missing Katakana Name".

Notes:

- **Segmentation** – Both versions of the name are not necessarily segmented. Special treatment is performed for segmenting vendor names for creating indexes.

- **Katakana Name field Caption** – Libraries may rename the caption of the Contact-5 field to "Katakana Name".
  This is done by updating the field name in the file: .\acq\tab\lng\window.dat

  Replace the following line:
  VendorsExpand.Static.VendorContact5 Contact 5:

  With:
  VendorsExpand.Static.VendorContact5 Katakana Name

- **Letters to Vendor Templates** – Some of the "Letters to Vendor" template files (XSL files) include the vendor Contact-5 field (Z70-CONTACT-5). The vendor contact-5 field has to be removed from the XSL templates so that the Katakana name is not mistakenly used in the letters.
- Usually, the Vendor Contact-5 field is listed in the Order Form -Vendor 3rd Tab. When the JAPANESE_ENABLED flag has the value TRUE, the Contact-5 field is not displayed in the order form.

34.2 Sorting Vendor Name in Katakana

The sorting of the vendor name list is set according to the Katakana name.

Usually, the patron names list is sorted by Z70-NAME-KEY which is created from Z70-VENDOR-NAME.
When the JAPANESE_ENABLED flag has the value TRUE, the Z70-NAME-KEY is created from Z70-CONTACT-5.

The usual normalization of the vendor name for sorting is performed, as in the below sample setup:
The sorting form is created according to the VENDOR-NAME-KEY procedure, as set in the configuration table tab_character_conversion_line.

The instance VENDOR-NAME-KEY should be set in
./alephe/unicode/tab_character_conversion_line.

```
          1              2   3         4                                    5            6
!!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!-!
VENDOR_NAME_KEY      ##### # line_utf2line_utf              adm_name_key
```

In the above sample setup, activate filing normalization of vendor name as defined in ./alephe/unicode/adm_name_key.
The library may activate another filing routine by setting up another routine in addition to the VENDOR-NAME-KEY instance.

Note:

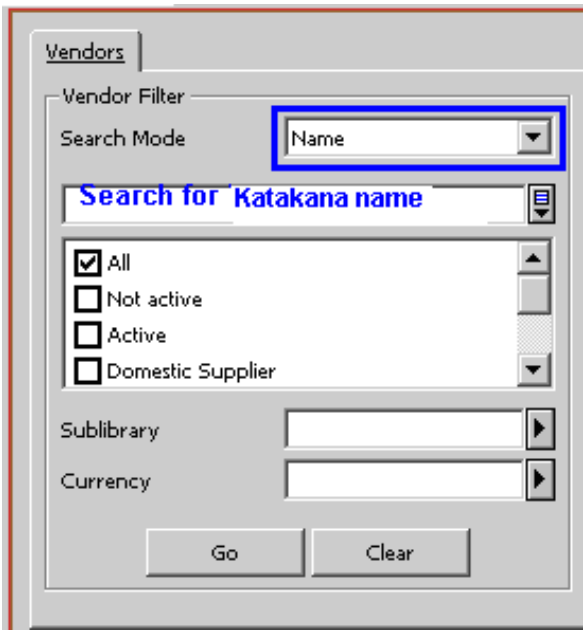
Z70-NAME-KEY length is 40 bytes; therefore the sort does not distinguish between names that start with the same thirteen normalized Katakana characters.

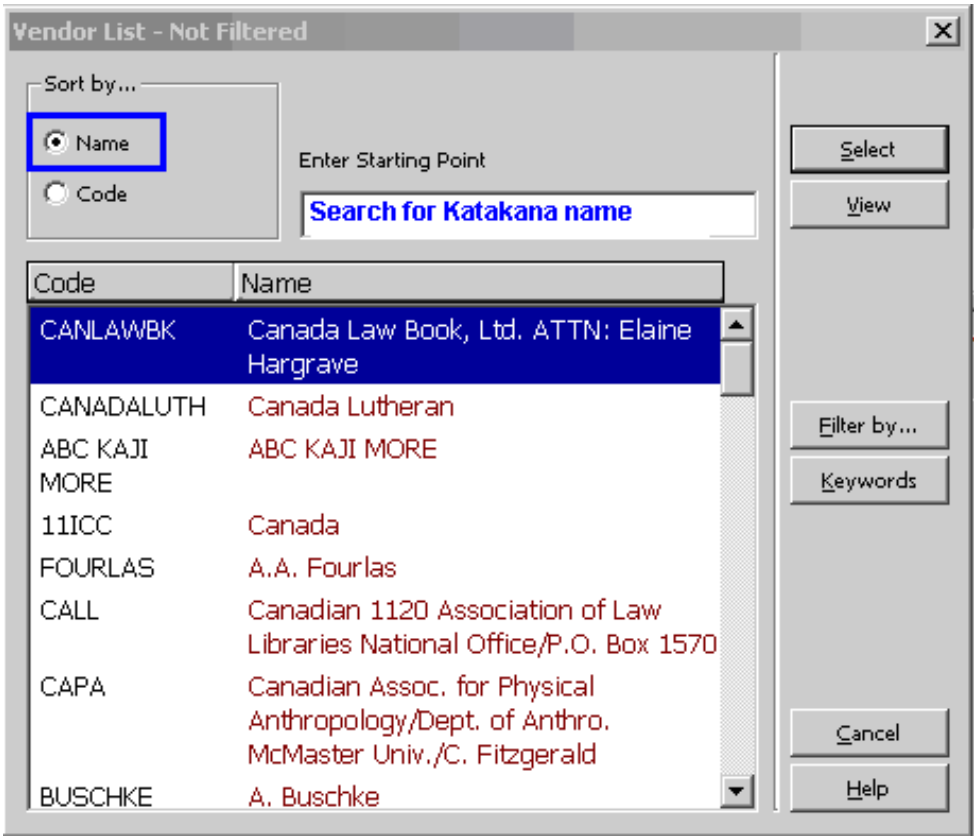
34.3 Searching for Vendor Name in Kanji and Katakana

Searching for the vendor name is enabled for both: Kanji and Katakana.

34.3.1 Searching for Vendor Katakana Name

Search by Name (Name starts with …) is a search option that uses the name key to set the starting point; therefore, the search query must be entered in Katakana. The search query is also normalized based on `VENDOR_NAME_KEY` entry.





The result list is sorted according to the VENDOR-NAME-KEY which relies on the Katakana name.

34.3.2 Vendor Keywords Search – Kanji and Katakana

Searching for both Kanji and Katakana versions of the name can be performed by using the "Keywords" mechanism.

The Z111 keywords mechanism enables the indexing of keywords from the Z70-VENDOR-NAME (Kanji name) and Z70-CONTACT-5 (Katakana name) fields.

Z111 indexes phrases included in certain vendor fields which enables searching for any words within the indexed string (not just the "Start with" string).

Searching for the Katakana and Kanji version of the name can be done by activating the "Keywords" button for the Vendor and typing Katakana/Kanji name at the "Enter Keyword/s" field.

The configuration table ./alephe/tab/z111_index enables the library to decide which fields, out of a pre-defined list of fields, are indexed and searched using Keywords.

Z111_Index Table

The ./alephe_tab/z111_index allows the library to set which field Z111 keywords are created so that only required fields are indexed when the record is updated or the p-manage-111 batch service is run (build keywords).

The table contains a single column:
Column 1: field to index (30 characters).
This column can contain the following fields (fields which are listed in column 1 are indexed and available for "Keywords Search")

Z303-NAME
Z303-PROXY-FOR-ID
Z303-PRIMARY-ID
Z303-SALUTATION

Z308-KEY-DATA

Z304-ADDRESS
Z304-ZIP
Z304-EMAIL-ADDRESS

Z310-BARCODE
Z310-ID

Z13-TITLE
Z13-AUTHOR

Z70-VENDOR-NAME
Z70-VENDOR-CONTACT
Z70-COUNTRY
Z70-MATERIAL-TYPE
Z70-ADDITIONAL-VENDOR-CODE
Z70-CONTACT-5

Z700-REC-KEY
Z700-NAME
Z700-GEN-INFO
Z700-GEN-CONTACT

Z76-EXTERNAL-BUDGET
Z76-CURRENCY
Z76-NAME
Z76-DEPARTMENT

Z72-VENDOR-ADDRESS

It is highly recommended to index only the Kanji and Katakana versions of the name (vendor-name and contact-5 fields) and disable the indexing of other fields (country and alike), as indexing of other fields introduces inconsistencies.

Vendor Names Segmentation for Keyword Search

Both versions of the name are not segmented; therefore, there is a special treatment for indexing and search of keywords.

Z111 Index Creation

The usual normalization of the vendor name for keyword purposesis applied as in the following setup:

The keywords form is created according to ADM-KEYWORD-KEY procedure as set in the configuration tables: tab_character_conversion_line.

The instance ADM-KEYWORD -KEY should be set in
./alephe/unicode/tab_character_conversion_line

```
          1              2   3        4                                5
!!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
ADM_KEYWORD_KEY      ##### # line_utf2line_utf              adm_name_key
```

The above sample setup activates the filing of normalization of vendor name as defined in ./alephe/unicode/adm_name_key.
The library may set any other table to be used for normalization.

When the value in the JAPANESE_ENABLED flag is TRUE, after normalizing the vendor name, all spaces are removed, and an index is created of all suffixes of the compressed name.

For example: If the compressed name is ABCDE….XYZ, the index is:
ABCDE….XYZ
BCDE….XYZ
CDE….XYZ
….
YZ
Z

Note:

Z111-TEXT length is 50 bytes; therefore, there is a limit to the length of the suffixes that are created.

Keyword Searching

The search text is also normalized according to ADM_KEYWORD_KEY, and spaces are removed.

Notes:

- Right truncation is applied automatically. As suffixes are indexed, entering substrings of compressed vendor names are retrieve for this vendor.
- Spaces are removed only if the JAPANESE_ENABLED flag is set to TRUE.

35 Updating the BIB Record with the Date of the ISSN Formal Notice

When a publisher (vendor) produces a new book, it must request an ISSN from the national library. The ISSN is sent to the publisher in an ISSN formal notice. There is an option to update the 906$$b field of the BIB record with the date that the ISSN formal notice is sent to the publisher.

To enable this option, configure the tab_z105 and aleph _start tables. Then perform the usual acquisition ordering workflow.

35.1 Configuring tab_z105

The tab_z105 table needs to be configured to update the BIB record every time a monograph order that matches the criteria set in aleph_start is sent to a vendor.

To configure the tab_z105 table:

1. Set the action, SEND-ORDER with message type, "j" in ./xxx50/tab/tab_z105 (column 1 and 2 respectively).
2. In column 3, define the BIB library to be updated when an order is sent to a vendor.

The following is an example of ./usm50/tab/tab_Z105:

```
Col. 1 – Action that trigger the message

Col. 2 – Message type

Col.3 and onwards - Code of library to which message is sent

!      1        2   3     4     5     6     7     8     9    10    11
12

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

SEND-ORDER      j USM01
```

3. Restart ue_11 (util E/11) in the Z105_library table (for example, USR00).

The BIB record is updated every time a monograph order that matches the criteria set in aleph_start is sent to a vendor.

35.2 Configuring aleph_start

Set the following two parameters in aleph_start to define the conditions in which 906$$bof the BIB record is updated with the date a monograph order is sent to a vendor:

- order_unit – the acquisition order unit designated for assigning ISSN notices for new publications (Z68-ORDERING-UNIT).
- letter_type – the order letter type that is set within the order used for sending the ISSN format notice (Z68-LETTER-TYPE).

The following is an example of /alephe/aleph_start:

```
! Parameters for creating Z105 type j (update BIB 906$$b
! when monograph order is sent)

       setenv    order_unit         2ISSN
       setenv    letter_type        12
```

Notes:

- The BIB 906$$b update applies only if the sent monograph order matches both parameters defined in aleph_start. If one or both parameters are missing or do not match the order, 906$$bis not updated.
- If the 906$$b field of the BIB record already exists, it is not modified.
- This feature is relevant only for libraries that are set to work with the acquisition ordering unit (./xxx50/tab/tab100 is set with: USE-ORDER-UNIT=Y).

36 Updating the BIB STA Field After Order Creation and Arrival

There is an option to update the STA$$a field of the BIB record with values (configurable by the library) when the first order record is created for a BIB record and the first order's arrival is registered.

To achieve this functionality, configure the tab_z105 and order_sta_update tables.

36.1 Configuring tab_z105

1. Set the action, UPDATE-ORDER and message type, "k" in ./xxx50/tab/tab_z105 (column 1 and 2 respectively).
2. Define in column 3 the BIB library to be updated when sending an order to a vendor.

The following is an example of ./usm50/tab/tab_Z105:

```
Col. 1 – Action that trigger the message

Col. 2 – Message type
```

```
Col.3 and onwards - Code of library to which message is sent
!      1        2   3     4     5     6     7     8     9    10    11
12
!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!!!!!-!!!!!
UPDATE-ORDER      k USM01
```

This configuration updates the BIB record each time an order that matches the criteria set in the order_sta_update table is created or arrives for the first time.

36.2 Configuring order_sta_update

The ADM library configuration table, order_sta_update, defines the values populated in STA$$a as a result of a first order creation and first order's arrival.

The following is an example of ./usm50/tab/order_sta_update

```
COL  1. - The action that trigger the STA update:
          ORDER-CREATE or ORDER-ARRIVAL.
          ORDER-CREATE action can be set for all order types (column
3:
          M/S/O).
          ORDER-ARRIVAL action can be set for Monograph
           and Standing order (column 3: M/O).
 COL  2.  -  Order's  sublibrary/Order unit filter.
             Library  that  uses  'ordering  unit'(TAB100-USE-ORDER-
UNIT=Y)
             should set the Order Unit code. Otherwise, set Order's
             Sublibrary code. Set ##### for all units/sublibraries.
 COL  3. -  Order Type filter: M/S/O/#
               M- Monograph, O-Standing order, S-Serials
               # - match all;
 COL  4. -  The value to be set in BIB STA$$a field.

!     1             2     3    4
!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!!
ORDER-CREATE         ORDU1 M A2
ORDER-CREATE         ORDU1 O A3
ORDER-ARRIVAL        ORDU1 # C1
```

After configuring the tab_z105 table with action UPDATE-ORDER and message type k and configuring the order_sta_update table, each time an order matches the filters set in columns 1-3 of order_sta_update, the STA$$a field of the bibliographic record is filled with the value set in column 4 of order_sta_update.

To activate this functionality, restart ue_11 (util E/12 and util E/11) in the Z105_library table (for example, USR00).

Notes:

- If the STA$$a field of the BIB record does not exist or contains a value beginning with an "A", it is modified.
- If the STA$$a field of the BIB record exists and contains a value beginning with any value other than "A", it is not modified.
- ORDER-ARRIVAL action refers to monograph and standing orders only. Serial orders arrivals (which are managed via issue items workflow) do not affect the BIB STA field.

37 Order Changes Alert

You can configure Aleph to display the following message if a user updates an order and then moves to another order without saving the changes:

```
Order has been changed locally, but the changes have not been saved
on the server. Continue anyway?  Yes/No
```

This functionality is flag dependent.

1) In order to have this order change alert, set the following in alephcom.ini:

```
[ItemList]

NotifyItemChanged=Y
```

2) If localization is required, add the following line to `Acq/Tab/lng/Message.dat`:

```
OrderChange                    ALEPH GUI            YN  2 Q Order %s
has been changed locally, but the changes have not been saved on the
server. Continue anyway?
```

Note: This message appears only when moving between orders within the orders list pane. When moving to the left pane or any other area that is not the orders list, the message does not appear.

System Librarian's Guide - ADAM

Version 23

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 GENERAL SETUP 5

1.1 Oracle Text 5

1.2 Storage Directories for Objects (tab_z403_directory.<lng>) 5

1.3 File Types of Objects (tab_z403_extension.<lng>) 7

1.4 Create Objects (pc_tab_exp_field.<lng>) 9

1.5 ADAM-SUPPORT Setup in tab100 10

2 DISPLAYING OBJECTS IN THE ALEPH CLIENT 10

2.1 Display in the Record Manager 10

2.2 Display of the Objects List 11

2.3 Display of Object Details 12

3 DISPLAYING OBJECTS IN THE OPAC 13

3.1 apache_media 13

3.2 tab_buf_z403 and tab_expand 13

3.3 Brief list OPAC (www_tab_short.<lng>) 14

3.4 Full View in the OPAC (edit_doc_999.<lng>) 14

3.5 Extended Object Information (tab_z403_info.<lng>) 14

3.6 Options for Displaying Thumbnails (tab100) 15

4 COPYRIGHT NOTES IN OPAC 17

5 FULLTEXT SEARCH 18

5.1 Set Up the Fulltext Index (tab00.<lng>) 18

5.2 Integrate the Fulltext Index for Web and OPAC Searching 18

5.2.1 Fulltext Search in the GUI (pc_tab_sear.<lng>) 18

5.2.2 Search in the OPAC (find-code-include et al.) 18

5.3 Character Set 18

6 SERVICES 19

6.1 Services for Import and Export 19

6.1.1 Create Digital Objects Upload File (adam-01) 20

6.1.2 Import Records and Digital Objects (adam-02) 21
6.1.3 Export Records (adam-04) 22

6.2 Other Services 22
6.2.1 Change Object Location (adam-03) 22
6.2.2 Create Thumbnail or Full Text (adam-05) 23
6.2.3 Technical Report (adam-06) 23
6.2.4 Usage Report (adam-07) 24
6.2.5 Create Digital Objects Upload File from DigiTool Export (adam-08) 24
6.2.6 Retrieval of ADAM Records (Z403) (adam-10) 25
6.2.7 Deletion of ADAM records (Z403) (adam-11) 26
6.2.8 Update Full Text Index (manage-91) 27

7 SETUP OF ADDITIONAL ORACLE TABLES 28

7.1 Z403 28

7.2 Z91 29

7.3 Z56 29

8 ADAM SCAN JOBS 29

9 CREATE THUMBNAILS 31

1 General Setup

1.1 Oracle Text

For fulltext indexing of objects (z91), the component 'Oracle text', also referred to as *context,* and the Oracle user "ctxsys" are needed.

Normally, 'Oracle text' will be installed automatically. If the instance was aborted, 'Oracle text' has to be installed later.

The following command will check if 'Oracle text' is available on your system:

```
$ORACLE_HOME/ctx/bin/ctxhx
```

Output:

```
/exlibris/app/oracle/product/920/ctx/bin/ctxhx - Converts the input file to plain
text or html file.
Usage:  /exlibris/app/oracle/product/920/ctx/bin/ctxhx InputFile OutputFile
        [FallbackCharacterSet
        [OutputCharacterSet
        [H | T
        [Meta | Nometa
        [TimeOut
        [Heuristic | Fixed]]]]]]
Where:  "InputFile" is the file to be converted to HTML or TEXT.
        "OutputFile" is where to place the converted file.
        "FallbackCharacterSet" is the character set of InputFile
        which is assumed to be plain-text when the format is not supported.
        Default fallback character set is ASCII8.
        "OutputCharacterSet" is the character set of OutputFile.
        Default character set is ISO8859-1.
        "H | T" is to define the output format as HTML or plain text.
        Default output format is HTML.
        "Meta | Nometa" is to enable/disable meta data extraction.
        Default is Nometa.
        "TimeOut" is the polling interval in integral seconds to determine
        whether to terminate by force.
        Supported range is 0..42949672.
        Default is 0 to disable time-out.
        "Heuristic | Fixed" is the type of time-out.
        Default is Heuristic.
```

1.2 Storage Directories for Objects (tab_z403_directory.<lng>)

Every digital object has to be stored in a directory in the file system of the ALEPH server unless you link to an object on a remote server by entering a URL.
This directory must include the following subdirectories:

- View
- Index
- Thumbnail
- Backup

The table tab_z403_directory.<lng> defines the storage directories for objects on the ALEPH server. The table may be placed in the BIB library's tab directory. If so, it is possible to store the objects in different subdirectories, if more than one BIB library is active on the same ALEPH installation. The table tab_z403_directory.<lng> may also be placed in the directory $alephe_tab if storing related to a specific BIB library is not needed.

All of the storage directories which are defined in tab_z403_directory.<lng> have to be created manually on the server.

The maximum number of files per directory is defined by the parameter *max_no_storage_files* in $alephe_root/aleph_start file. The following is an eample of how to set this parameter:

```
setenv max_no_storage_files 1000
```

The default value (if the variable is not defined) is 1000.

ADAM automatically creates new sub-directories under the defined storage directory, if the maximum number of files is reached. The directory names are numeric (starting with 1). The first 1000 objects are automatically stored in directory /1. Subsequent objects are stored in directory /2 and so on.

See the following example:

```
! 1       2                     3                                4
!!!!!-!!!!!!!!!!!![…]-!!!!!!!!!!!!!!!!!!!!!![…]-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!![…]
ALEPH VIEW           Zentraler Speicher       $aleph_dev/demo_objects/usm01/view
ALEPH INDEX          Index (Allgemein)        $aleph_dev/demo_objects/usm01/index
ALEPH THUMBNAIL      Thumbnail (Allgemein)    $aleph_dev/demo_objects/usm01/thumbnail
ALEPH BACKUP         Backup (Allgemein)       $aleph_dev/demo_objects/usm01/backup

MAB50 VIEW           Primaerer Speicher       $aleph_dev/demo_objects/mab50/view
MAB50 THUMBNAIL      Thumbnails               $aleph_dev/demo_objects/mab50/thumbnail
MAB50 INDEX          Index                    $aleph_dev/demo_objects/mab50/index

MEDUC VIEW           MEDUC Speicher           $aleph_dev/demo_objects/mab50/meduc/view
MEDUC THUMBNAIL      MEDUC Thumbnails         $aleph_dev/demo_objects/mab50/meduc/thumbnail
MEDUC INDEX          MEDUC Index              $aleph_dev/demo_objects/mab50/meduc/index
```

Key to the table:

Column 1:	1) Sublibrary code, e.g. MEDUC. Storage directories for objects, which are assigned to a specific sublibrary 2) Code of the ADM library, e.g. USM50. Storage directories for objects, which are assigned to the ADM library ALEPH storage directories for objects where the field “Cat. sublibrary” is empty
Column 2:	Usage type of the object. Valid options: INDEX, VIEW, THUMBNAIL, BACKUP *)
Column 3:	Description of the storage directory, appears in the pull down list in the GUI
Column 4:	Pathname to the storage directory

Usage type of objects:

BACKUP: When loading objects with the service adam_02, copies may remain in the specified directory. These objects have the usage type BACKUP.

INDEX: An object was automatically built by ADAM when performing the Indexing function.

VIEW: Usage type of all objects that are displayed in the OPAC.

THUMBNAIL: An object was automatically built by ADAM when performing the Thumbnail function.

1.3 File Types of Objects (tab_z403_extension.<lng>)

The table $alephe_tab/tab_z403_extension.<lng> includes definitions for file types of objects. Depending on the file extension, some parameters must be determined, e.g. the icon that represents the object in the OPAC or the permission to create thumbnails or to perform a full text indexing of an object.

Example:

```
! 1       2            3               4 5  6       7          8      9
!!!!!-!!!!!!!![…]-!!!!!!!!!!!!!!!!!!!!-!-!-!!!!-!!!!!!!!!!-!!!!!!!!!!-!!![…]!->
asf   asf        f-tn-movie.jpg
avi   avi        f-tn-movie.jpg
bmp   bmp        f-tn-image.jpg         Y 100k convert
djv   djv        f-tn-image.jpg
djvu  djvu       f-tn-image.jpg
doc   doc        f-tn-doc.jpg           Y 100k            gen
ead   ead        f-tn-text.jpg
gif   gif        f-tn-image.jpg         Y 100k convert
html  html       f-tn-link.jpg
ip2   ip2        f-tn-image.jpg
ipg   ipg        f-tn-image.jpg
jpg   jpg        f-tn-image.jpg         Y 100k convert
mid   mid        f-tn-sound.jpg
mp3   mp3        f-tn-sound.jpg
mpg   mpg        f-tn-movie.jpg       N
mpeg  mpeg       f-tn-movie.jpg       N
pdf   pdf        f-tn-pdf.jpg                           gen
ppt   ppt        f-tn-ppt.gif           Y 100k
ram   ram        f-tn-sound.jpg
```

```
rtf   rtf        f-tn-doc.jpg                            gen
sid   sid        f-tn-image.jpg
tif   tif        f-tn-image.jpg
txt   txt        f-tn-text.jpg          N 100k           txt
url   url        f-tn-link.jpg
wav   wav        f-tn-sound.jpg
wmv   wmv        f-tn-sound.jpg       N
xls   xsl        f-tn-xcel.jpg
xml   xml        f-tn-XML.jpg           Y 100k
#####            f-tn-unknown.jpg
```

Structure of the table:

Column 1:	Object's file extension (max. 5 characters)
Column 2:	File type's description (max. 20 characters)
Column 3:	File name of the icon used in the OPAC. The icons must be located in *$alephe_root/www_f_<lng>/icon* They will be displayed in the brief list and the full view in the OPAC. The maximum number of icons being displayed in brief list and full view depends on the value of the variable "www_short_max_no_ext_links" in *$alephe_root/www_server.conf*. If no specific icon is stated, a standard icon will be used, which is defined in the last line of the table (##### in Column 1)
Column 4:	Defines if a new browser window is opened when calling an object. Default value is Y.
Column 5:	Immediate display of the object. Value is Y or N. Defines if an object is immediately displayed in the GUI when the ALEPH browser is in use. This definition depends on the object's size. (see column 6)
Column 6:	Object size (only applies in conjunction with column 5) Defines from which object size the immediate display in the GUI will be suppressed. The size may be stated in kilobytes (K), megabytes (MB) or gigabytes (GB).
Column 7:	Program that is used for creating thumbnails. Valid value: convert If you want to create thumbnails from objects, the value 'convert' must be set in Column 7 for those objects. If this column is left empty, the **Thumbnail** button in the Object List is deactivated.
Column 8:	Program that is used for Fulltext indexing. Valid value: gen The entry 'gen' in Column 8 must be set for all objects for which a fulltext indexing should be performed. If this column is left empty, the **Indexing** button in the Object List is deactivated.
Column 9:	Viewer If you want to use an external viewer for displaying objects, the pathname to the viewer's executable file must be entered here. If this column is left empty, the default plugin of the browser will be used.

1.4 Create Objects (pc_tab_exp_field.<lng>)

The table pc_tab_exp_field.<lng> contains selective lists for fields of the Object Form. This table is located in tab-directory of your ADM library (lib50/tab). The following values must be filled in:

```
!        1               2      3                                                  4
!!!!!!!!!!!!!!!!!!!!-----!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
OBJECT-USAGE-TYPE        L View                                                VIEW
OBJECT-USAGE-TYPE        L Thumbnail                                           THUMBNAIL
OBJECT-USAGE-TYPE        L Indexing Text                                       INDEX
!*
OBJECT-CHAR-SET          L ISO Latin                                           iso8859-1
OBJECT-CHAR-SET          L Cyrillic 1251                                       windows1251
OBJECT-CHAR-SET          L Cyrillic 8859                                       iso8859-5
OBJECT-CHAR-SET          L Hebrew                                              iso8859-8
!*
OBJECT-RESOLUTION        L High Resolution                                     high
OBJECT-RESOLUTION        L Low Resolution                                      low
OBJECT-RESOLUTION        L Medium Resolution                                   medium
!*
OBJECT-COPYRIGHT-OWN     L Public                                              PUBLIC
OBJECT-COPYRIGHT-OWN     L Local                                               LOCAL
!*
OBJECT-COPYRIGHT-TYP     L Registered                                          REGISTERED
OBJECT-COPYRIGHT-TYP     L Copyrighted                                         COPYRIGHTED
OBJECT-COPYRIGHT-TYP     L Licensed                                            LICENSED
OBJECT-COPYRIGHT-TYP     L Public Domain                                       PUBLIC
OBJECT-COPYRIGHT-TYP     L Contractual                                         CONTRACT
OBJECT-COPYRIGHT-TYP     L Donated                                             DONATED
!*
OBJECT-COPY-NOTICE-T     L Free for all                                        FREE
!*
```

Key to the table:

Column 1:	Internal code for GUI field
Column 2:	ALPHA Always “L“.
Column 3:	Description which is displayed in the dropdown list
Column 4:	Code for the entry in the dropdown list. Only internal codes can be used.

The internal codes represent the following fields:

- OBJECT-USAGE-TYP
  The entries define the dropdown list for the object’s Usage Type. Note that the usage type is connected to system functionality. Valid options are VIEW; THUMBNAIL, and INDEX.

- OBJECT-CHAR-SET
  The entries define the dropdown list for the field ‘Character set’ (Technical Data Tab), which is used for full text indexing. Definitions in table ./alephe/unicode/tab_character_conversion_line have to be defined for each character set which is in use. These lines define the required character conversion.

  See the following example:
  ```
  iso8859-1      #### # line_sb2line_utf 8859_1_to_unicode
  ```

  The following character sets are supported by the indexing routines:

- ISO8859-1
- ISO8859-2
- ISO8859-3
- ISO8859-4
- ISO8859-5
- ISO8859-6
- ISO8859-7
- ISO8859-8
- ISO8859-9
- Gb2312
- Big5
- Shiftjis
- Windows1250
- Windows1251
- Windows1252
- Windows1253
- Windows1253
- Windows1254
- Windows1255
- Windows1256
- Windows1257
- Koreanhangul
- Windows874#
- Unicode

- OBJECT-RESOLUTION
  The entries are used to define the dropdown list which contains the resolution for image objects in tab "Technical Data".

- OBJECT-COPYRIGHT-OWNER
  This option is used for the dropdown list Copyright Owner in the Copyright tab

- OBJECT-COPYRIGHT-TYP
  This option is used for the dropdown list Copyright Type in the Copyright tab.

- OBJECT-COPY-NOTICE-T
  This option defines the dropdown list for Copyright Notice type in the Copyright tab.
  Note that the code in column 4 of the table corresponds with the suffix to the file `media-permission` in `$alephe_root/www_f_<lng>`.

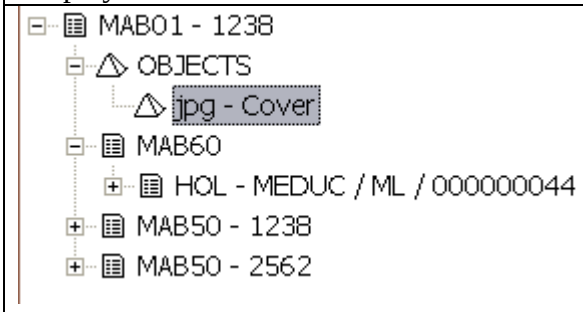
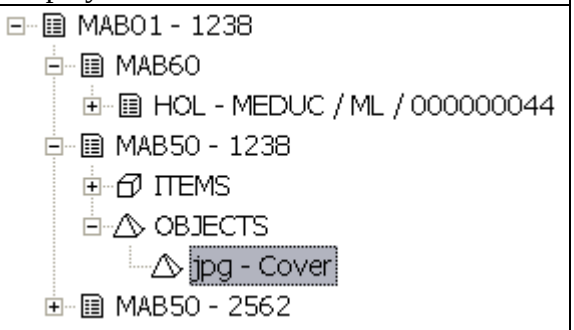
1.5 ADAM-SUPPORT Setup in tab100

The ADAM-SUPPORT switch of tab100 is used to define whether ADAM objects should be replicated as physical digital objects or not.
Relevant values: Y (default) or N.
N – If a local system does not have a full ADAM license, it is recommended to set ADAM-SUPPORT=N in order to receive the links to the objects as part of the bibliographic record instead of physical digital objects.
Y – Local libraries that have a full ADAM license should set: ADAM-SUPPORT=Y (default value).
The ADAM-SUPPORT switch can be defined locally per library (for example, at ./xxx50/tab/tab100) or on a global level at ./alephe/tab/tab100.

2 Displaying Objects in the ALEPH Client

2.1 Display in the Record Manager

Objects are linked to a record in the bibliographic library (lib01). In the Record Manager (lower part of the left frame) of the Cataloging module, the nodes for each object are displayed. The display of these nodes is similar to the display of items. The display of objects may be under the BIB record or under ADM record:

Display related to the BIB record	Display related to the ADM record
	

This is defined by the variable `item_object_tree_style` in the file *$alephe_root/pc_server_defaults*. The following values are valid:

```
setenv item_object_tree_style       1
```

The objects displayed are related to the ADM record.

```
setenv item_object_tree_style       2
```

The objects displayed are related to the ADM record, if the stated sublibrary in the field ‘Cat. Sublibrary’ is assigned to the ADM library or if the field is left empty.

```
setenv item_object_tree_style       3
```

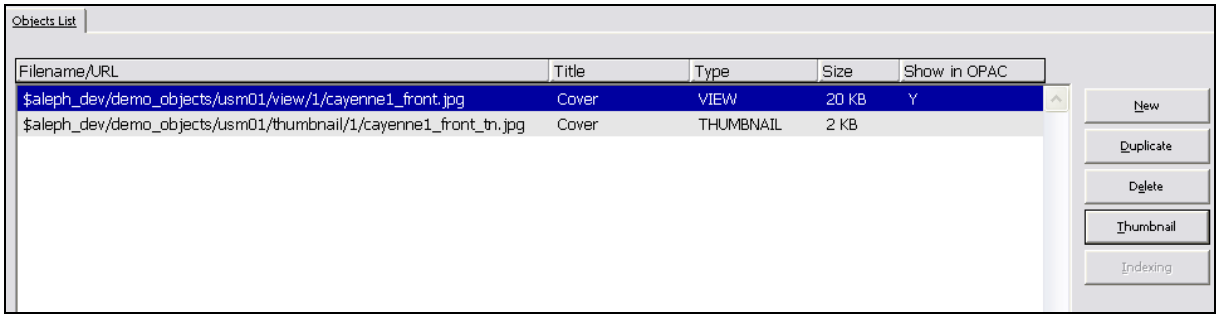
The objects displayed are related to bibliographic record (option 4), and to the ADM record (option 2).

```
setenv item_object_tree_style       4
```

The objects are displayed related to the BIB record.

2.2 Display of the Objects List

The display of the Objects List is defined in pc_tab_col.<lng>, which is located in the tab-directory of your BIB library (lib01).

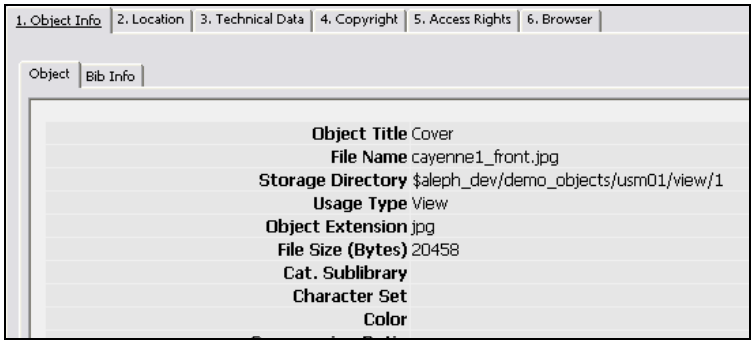



```
!          1              2              3              4    5    6    7    8    9    10
```

```
!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!-!!!-!!-!!!-!!-!!!-!!![…]
PC_OBJECT_LIST       L Dateiname/URL        01 035 01 C01
PC_OBJECT_LIST       L Titel                02 035 01 C01
PC_OBJECT_LIST       L Typ                  03 010 01 C01
PC_OBJECT_LIST       L Größe                04 010 01 C01
PC_OBJECT_LIST       L Anzeigen im OPAC     05 010 01 C01
```

2.3 Display of Object Details

Some windows in the GUI are generated from HTML files. Those HTML files are located in the directory pc_display_<*lng*> in the ADM library. The display of the object details is defined by the file "object-expand".



3 Displaying Objects in the OPAC

3.1 apache_media

For displaying objects in the OPAC, the directory apache_media must be created. In addition, you have to define the variable “setenv APACHE_MEDIA” in file $alephe_root/aleph_start. This directory contains temporary files and should be cleared regularly, for example, by using [Util X/6].

aleph_start

```
setenv APACHE_MEDIA     ${ALEPH_MOUNT}/a${ALEPH_VERSION}_$ALEPH_COPY/apache_media
```

Create a softlink (apache)

Create a softlink in the directory $httpd_root/htdocs/exlibris/aleph/a18_1 that links to the directory ‘apache_media’. The command is:

```
ln –s <target directory as defined in aleph_start> <source directory>
```

See the following example:

```
surprise-a18(1) >>pwd
/exlibris/aleph/a18_1/alephe/apache/htdocs/exlibris/aleph/a18_1
surprise-a18(1) >>ls -l
total 6
drwxrwxr-x   6 aleph    exlibris     512 Nov 20 02:07 aleph/
lrwxrwxrwx   1 aleph    exlibris      34 Nov 20 02:07 apache_media ->
/exlibris/aleph/a18_1/apache_media/
lrwxrwxrwx   1 aleph    exlibris      25 Nov 20 02:07 tmp ->
/exlibris/aleph/a18_1/tmp/
```

3.2 tab_buf_z403 and tab_expand

For displaying objects in the OPAC, define the following line in table ./[BIB_library]/tab/tab_buf_z403.

```
!   1                   2                         3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!>
########## get_buf_z403_media              VIEW
```

In addition, further entries in this table are necessary for displaying the fields 655 (electronic address, MAB) and 856 etc. (MARC) which contain URLs.

For MAB environment:

```
########## get_buf_z403_bib                655#1,,,,D,
```

For USM environment:

```
########## get_buf_z403_bib                856##,,y,*,Y,
########## get_buf_z403_bib                856##,,y,-,U,
########## get_buf_z403_bib                505##,,u,*,U,
########## get_buf_z403_bib                520##,,u,*,D,
```

```
########## get_buf_z403_bib              530##,,u,*,D,
########## get_buf_z403_bib              545##,,u,*,D,
########## get_buf_z403_bib              552##,,u,*,D,
########## get_buf_z403_bib              555##,,u,*,D,
########## get_buf_z403_bib              583##,,u,*,D,
```

In a MAB environment, the table ./[BIB_library]/tab/tab_expand must contain the following definitions for displaying objects:

```
BUF-Z403   expand_doc_mab
```

3.3 Brief list OPAC (www_tab_short.<lng>)

Two new columns must be defined through ./[BIB_library]/tab/www_tab_short.<lng> for displaying thumbnails and icons for objects in the brief list of the OPAC.

```
! 2   3          4    5  6   7   8 9    10      11   12   13
!-!-!!!!!![…]-!!![…]-!!-!!-!!!!-!-!-!!!!!!!!!!-!!-!!!!!-!!!!!-[…]
[…]
7 L Photo            00 00 0040 S             ## TN
8 L Media            00 00 0040 S             ## MED
```

Note: It might be necessary to activate the additional columns through the files short-a-head and short-a-body in $alephe_root/www_f_<lng>.

3.4 Full View in the OPAC (edit_doc_999.<lng>)

The table ./[BIB_library]/tab/edit_doc_999.<lng> must contain the following line for displaying objects in the Full view:

```
!                                                 1     1       1
!1   2   3      4     5 6 7         8          9 0 11  2  13   4 15 16
!!-!!!!!-!-!!!!!!!!!!-!-!-!!!!!!!!!!!!!!!!!!!!-!-!-!!!-!-!!!!!-!-!!-!!
## MED##                D LDigital Object      Y M     E
```

3.5 Extended Object Information (tab_z403_info.<lng>)

In the Full view of a bibliographic record, each object which is linked to the BIB is displayed with an icon and the title of the object. Basic information will be displayed if the icon is highlighted with the mouse.

The table $alephe_tab/tab_z403_info.<lng> determines which information from the object's metadata is displayed.

See the following example:

```
!    1                                        2           3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!![…]
z403-title                                                Title:^
z403-f-filename                                           Filename:^
z403-object-extension                                     File Extension:^
z403-original-filename                                    Original Filename:^
z403-object-size                          SIZE            Object Size:^
z403-note-1                                               File type:^
z403-update-date                          DATE            Last Update:^
```

Key to the table:

Column 1:	Field of z403 record that shall be displayed
Column 2:	Object type. This column defines whether the type of the object is to be converted into a formatted display. Values: DATE – for date fields SIZE – for the object size
Column 3:	Prefix for the display.

3.6 Options for Displaying Thumbnails (tab100)

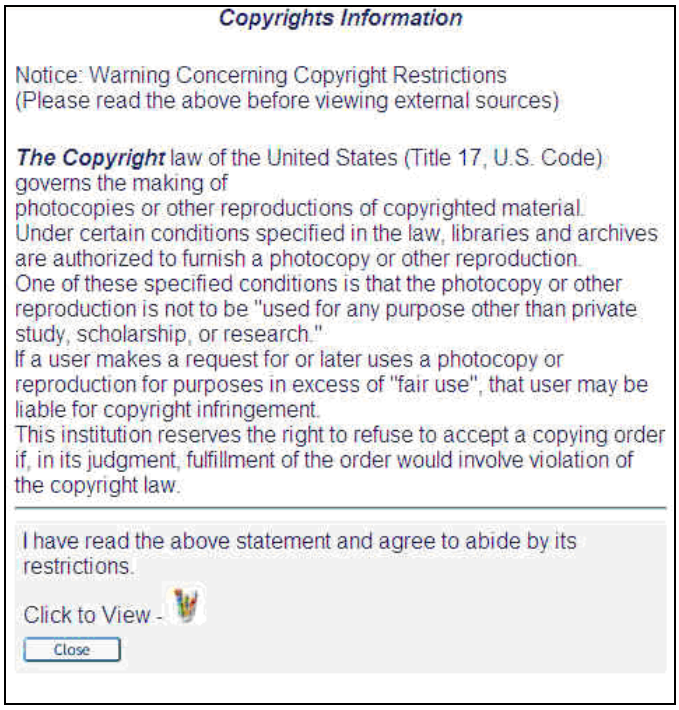
The parameter THUMBNAIL-LINK defines whether the thumbnail can be used as a link for displaying the object in the Full view. It is set in ./[BIB_library]/tab/tab100 located in the tab-directory of your BIB library (lib01). Valid values are 1 or 2 (default: 1).

THUMBNAIL-LINK=1
If a thumbnail is linked to an object, the object will be displayed by clicking on the thumbnail. If no object is linked, the object will be displayed in the Full view.

THUMBNAIL-LINK=2
Thumbnails cannot be used for displaying objects.

4 Copyright Notes in OPAC

You may determine if a copyright window will be displayed before users can access objects.



The related HTML file is $alephe_root/www_f_<lng>/media-permission.

You may design different pages for Copyright Notices by copying the file “media-permission” and adding specific suffixes, e. g. media-permission-**free**.
The field **Copyright Notice Type** in the GUI (Copyright Tab) defines which notice will be displayed. The field is connected to the table ./[ADM_library]/tab/pc_tab_exp_field.<lng>. The internal code is “OBJECT-COPY-NOTICE-T“:

```
!         1                2       3                                  4
!!!!!!!!!!!!!!!!!!!!-----!-!!!!!!!!!!!!!!!!!!!
```

5 Fulltext Search

5.1 Set Up the Fulltext Index (tab00.<lng>)

The fulltext index in ADAM is managed by the index codes TXT an BASE. Add the following lines to the keyword indices in table ./[BIB_library]/tab/tab00.<lng>:

```
!   2     3   4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
[…]
H TXT   W-nnn       00        01        W-Full text
H BASE  W-nnn       00        00        Base
```

The size of full text indexing for files given by the URL is limited to 500KB. You may increase the size by adding the following parameter to `$alephe_root/pc_server_defaults`:

```
setenv max_url_file_size 999999999
```

Note that the BASE line is necessary for the ADAM full text index. The base is added in addition to the search term.

5.2 Integrate the Fulltext Index for Web and OPAC Searching

5.2.1 Fulltext Search in the GUI (pc_tab_sear.<lng>)

To use the fulltext search in the GUI, the index code TXT must be added to the table ./[BIB_library]/tab/pc_tab_sear.<lng> in section FI.

```
!           2             3               4                      5
!!--!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!
FI  BIB01                 L Full text                          TXT
```

5.2.2 Search in the OPAC (find-code-include et al.)

The index code, TXT, must also be added to the search in the OPAC. You must add the TXT code to the other search types that are defined in the find-code-include file (in the $alephe_root/www_f_<lng> directory):

```
<option value=IDN $$FXX-S"IDN">ID</option>
<option value=TXT $$FXX-S"TXT">Full text</option>
```

5.3 Character Set

ADAM-INDEX-CHAR-SET

This value defines which character set is chosen as a default when creating indexes on text objects if no other value is entered in tab “Technical Data” of the object forms. The value for ADAM-INDEX-CHAR-SET in tab100 may be up to 20 characters. The default character set that is ISO 8859-1 (ISO-Latin).

6 Services

Detailed instructions how to run these services can be found in the GUI online help.

6.1 Services for Import and Export

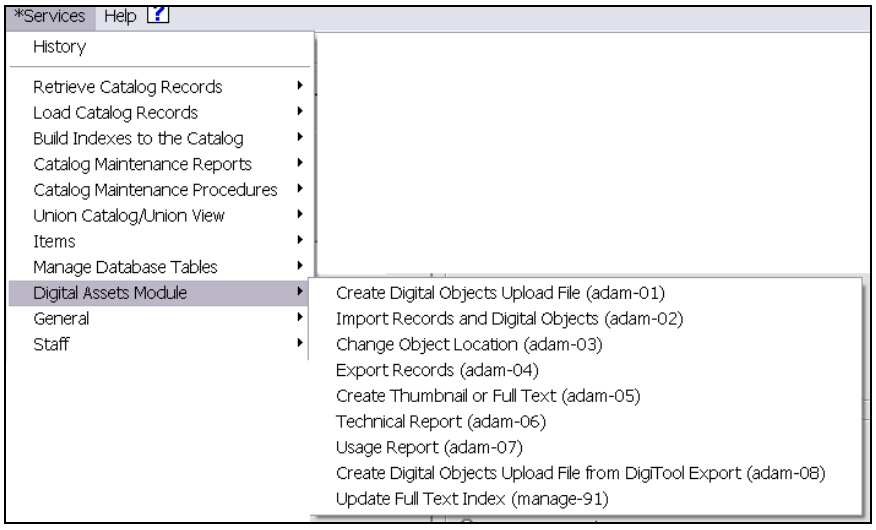
ALEPH offers batch-services to import and export digital objects and their corresponding metadata. While descriptive metadata (bibliographic) is stored as part of the bibliographic record (z00), the technical metadata is stored in its own Oracle table, z403.

Both import and export are based on XML-Files that contain specific sections for descriptive metadata and technical metadata.

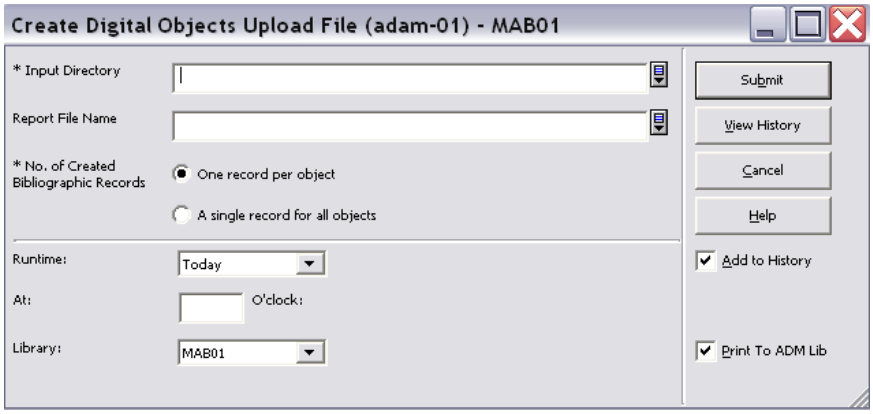
To upload objects into the system, it is usually necessary to run two services (adam-01 and adam-02). Both are part of the services menu in the Cataloging module.

- adam-01: Create Digital Objects Upload File
  Accumulates descriptive and technical metadata into XML-File based on template.

- adam-02: Import records and Digital Objects
  Loads the XML-File, which has been created by using adam-01, into the database.



6.1.1 Create Digital Objects Upload File (adam-01)



This service is used to create an XML-file which is required to upload files into the system that do not have descriptive metadata (bibliographic records) or technical metadata (z403). The XML-File created by this service contains descriptive and technical metadata generated by using a template (z403_template.xml) that may be edited by the user.

The XML-file created by this service may be taken to load objects and metadata into the system using service adam-02. An example of an XML template that can be edited by the user can be found in ./[BIB_library]/tab.

The XML-File contains different sections for descriptive `(<record>)` and technical metadata `(<z403>)`.

The following is an example for the XML-file

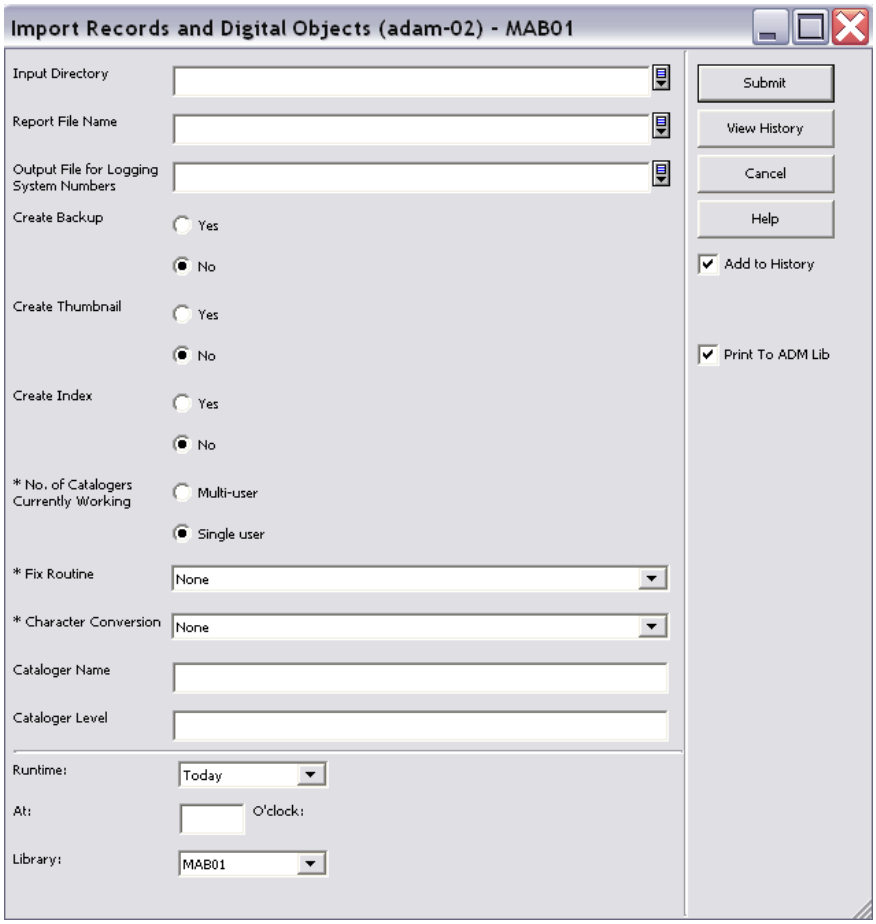
```
<?xml version="1.0" encoding="UTF-8"?>
<file>
<record xmlns="http://www.ddb.de/professionell/mabxml/mabxml-1.xsd">
<leader>00798nM2.01200024||||||h</leader>
  <datafield tag="100" ind1=" ">
    <subfield code="a">Verfasser</subfield>
  </datafield>
  <datafield tag="331" ind1=" ">
    <subfield code="a">Titel</subfield>
  </datafield>
</record>

<z403>
  <z403-doc-number></z403-doc-number>
  <z403-sequence></z403-sequence>
  <z403-derived-from-sequence></z403-derived-from-sequence>
  <z403-title></z403-title>
  <z403-f-directory></z403-f-directory>
  <z403-f-filename></z403-f-filename>
  <z403-original-file-name></z403-original-file-name>
  <z403-object-extension></z403-object-extension>
  <z403-object-size></z403-object-size>
  <z403-u-path></z403-u-path>
  <z403-usage-type>VIEW</z403-usage-type>
[…]
</z403>
</file>
```

As a result of service adam-01, another XML file is generated that contains the descriptive and technical metadata and a relative path to the digital objects. This XML file has the same name as the directory it is in plus the file extension “xml”. For example, if the directory that contains the objects and the template is named createxml, the XML result of adam-01 will be createxml.xml. It can be used for the actual upload with the adam-02 service.

Note that the XML file will contain the digital object’s file name as the title if no explicit title is specified in the `<record>`section of the template.

The standard is that per each object one descriptive record will be created. Uploading multiple objects with just one descriptive record can be done by editing the XML file. Limit the `<record>` section to only one appearance within the XML file by deleting further `<record>`sectors.

6.1.2 Import Records and Digital Objects (adam-02)



This service takes the XML file generated with adam-02 or any other XML file with appropriate format and content and runs the upload into the system. This service creates records in z00 (bibliographic) and z403 (technical metadata) according to the definitions in the XML file.
Note that the XML file to be loaded is required to have the same name of the directory it is in plus the file extension “XML”, otherwise it will not be loaded.

It is possible to upload digital objects to existing bibliographic records without changing the descriptive metadata (bibliographic). For this purpose, edit the XML file generated with

p_adam_01 or delivered in an appropriate format and leave all field tags in <record> sectorempty and add the following line to the <record>sector in the template.

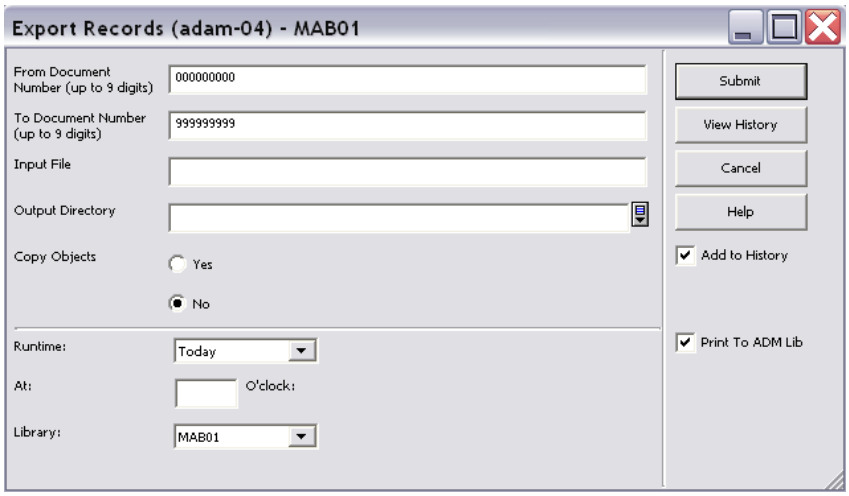
```
<?xml version="1.0" encoding="UTF-8"?>
<file>
<record xmlns="http://www.ddb.de/professionell/mabxml/mabxml-1.xsd">
    <controlfield tag="SYS">NNNNNNNNN</controlfield>
</record>
```

NNNNNNNNN is a variable for the record's system number. For example
<controlfield tag="SYS">000050018</controlfield>
will upload objects to bibliographic record with system no. 000050018.

Note that if the system-number does not exist in the catalog, no object will be uploaded.

6.1.3 Export Records (adam-04)

This service exports Z00 (descriptive) and Z403 (technical metadata) Oracle records in the defined XML format which is similar to the XML-format that is used for import. In addition, the service allows exporting not only the metadata but also the objects itself.

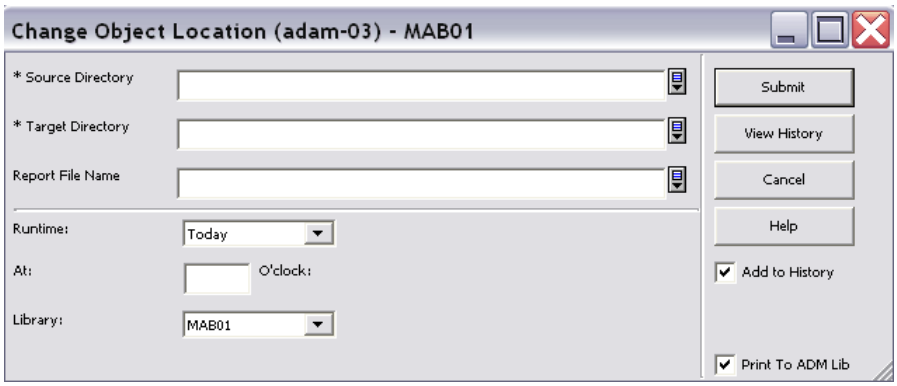


You can limit this service on a range of records contained in an input file created by service p-ret-01. Note that the output directory will automatically be created as subdirectory to the library's export directory.

6.2 Other Services

6.2.1 Change Object Location (adam-03)

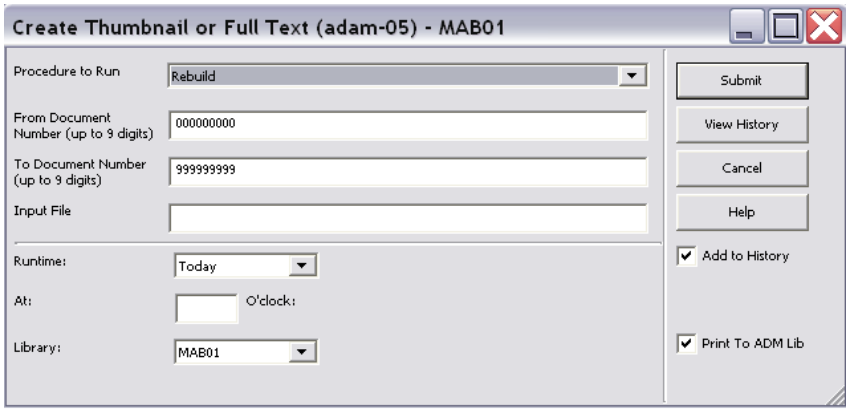
Digital objects are stored in a defined storage directory on the server. If you want to move objects from one directory to another directory when, for example, adding another disk to the server, you can use this batch server to transfer the objects to a new destination directory.



Enter the source directory that contains the objects that should be transferred, the destination directory that is designed to be the new storage directory for the objects, and the library on which you wish to run the service.

6.2.2 Create Thumbnail or Full Text (adam-05)

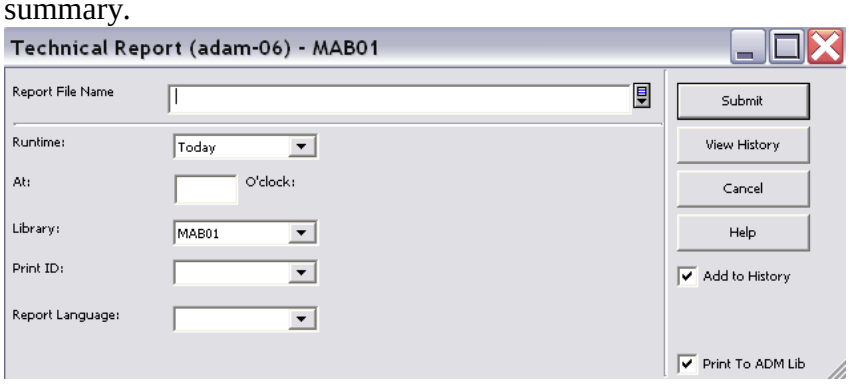
Thumbnails for image objects and full text for text objects can either be created by using the functions in the objects lists or by using this service. Especially when working with a huge amount of new records, it may be reasonable to create thumbnail and full text with the batch service instead of updating the objects one by one.



To run this service you can define a range of records or use an input file that may be created by ret-01.

6.2.3 Technical Report (adam-06)

This batch service is used to create a statistical report. It counts all Z403 records and prints a summary.

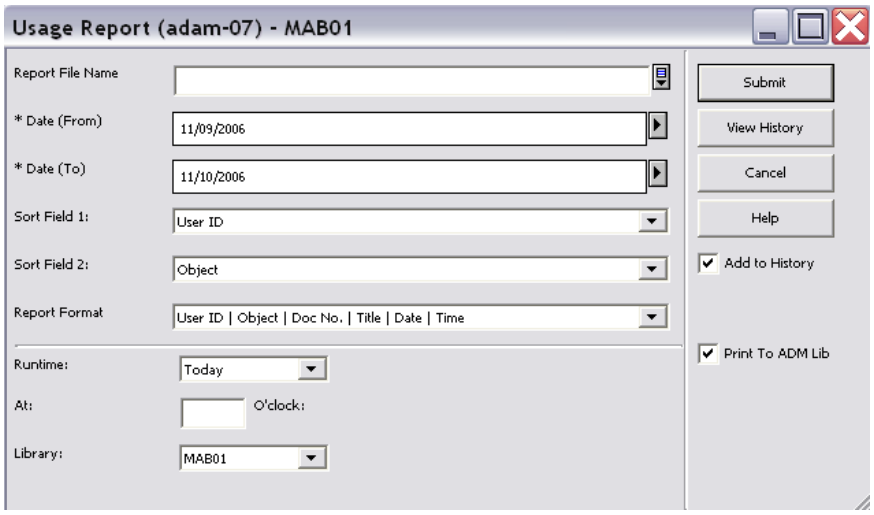


The technical report includes the following information:

- Total number of objects
- Number of objects per usage type
- Number of objects per ADM
- Total Storage
- Storage per directory
- Storage per extension

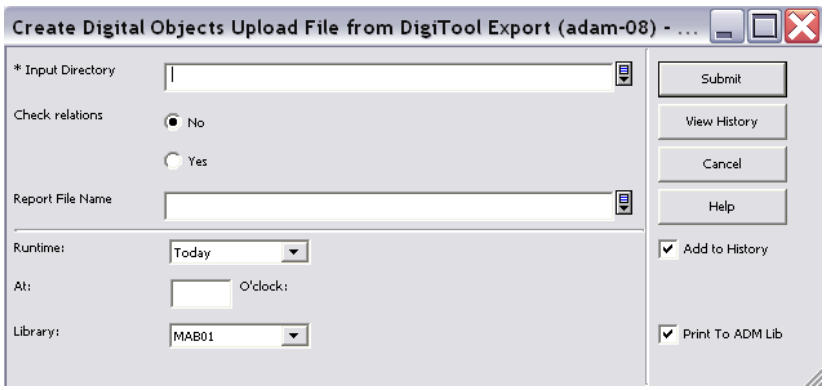
6.2.4 Usage Report (adam-07)

This service is designed for monitoring of user accesses. It generates a detailed breakdown of accesses sorted by various criteria. You can limit this report on a range of date.



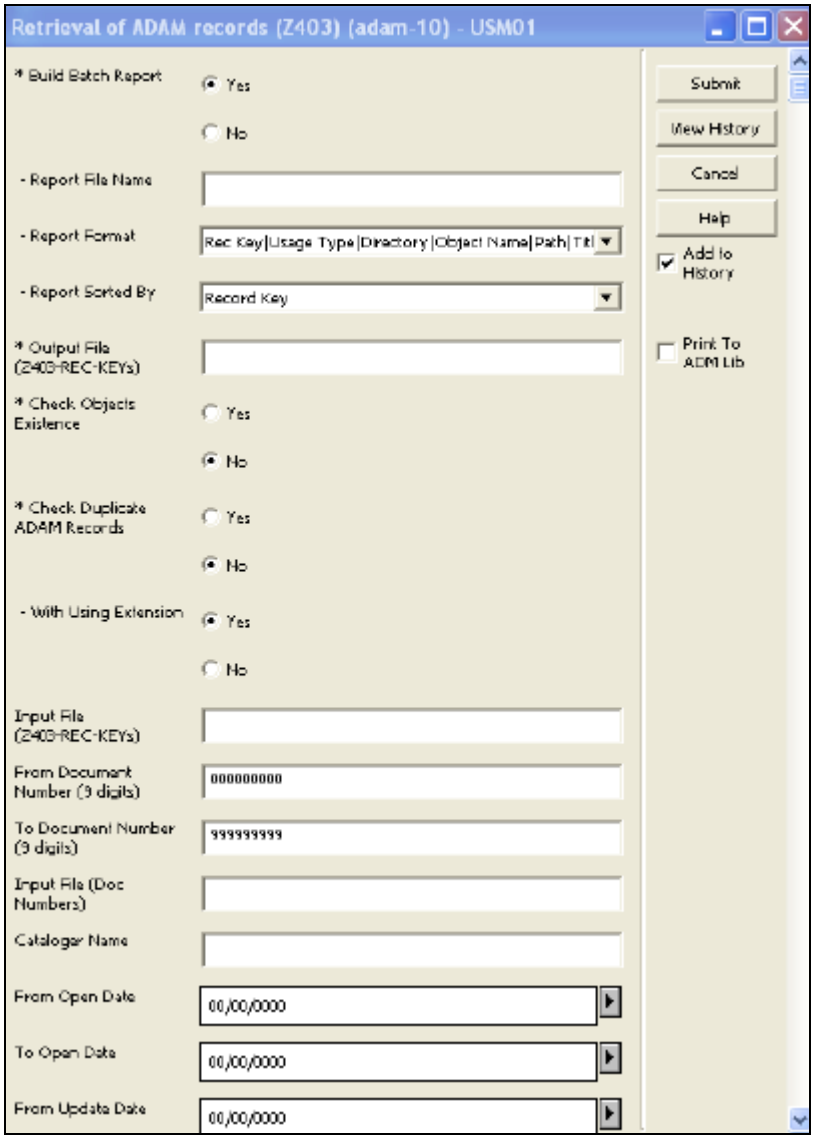
6.2.5 Create Digital Objects Upload File from DigiTool Export (adam-08)

This service can be used to convert exported DigiTool data (digital entities) into a format which can be handled by adam-02 service. The objects and XML files have to be transferred to the ADAM import directory in order to be available for adam-08. It is only possible to convert data of usage type "VIEW".



6.2.6 Retrieval of ADAM Records (Z403) (adam-10)

This service enables you to retrieve z403 records based on contents of specific z403 fields. You can also retrieve records which have no digital objects linked or duplicate z403 records. The service produces an output file which contains Z403-REC-KEYs. This file may be used as input file for service p_adam_11 and also as an additional filter for a new run of p_adam_10 himself.



The table pc_tab_exp_field.<lng> contains selective lists for fields of the service. The following values can be user-defined:

```
!!
!! For P-ADAM-10-SORT you can use (= Z403-......)
!!    REC-KEY, F-FILIE-NAME, OPEN-DATE, UPDATE-DATE, EXPIRY-DATE
!! and all values of COL.4 of  P-ADAM-10-FIELD
!! in any combinations to define a set of sort-keys
!!
P-ADAM-10-SORT           L Record Key                                    REC-KEY
P-ADAM-10-SORT             L Storage directory | File Name                 F-DIRECTORY|F-
FILENAME
P-ADAM-10-SORT           L Cataloger | Usage type | Object title          CATALOGER|USAGE-
TYPE|TITLE

P-ADAM-10-USAGE          L VIEW                                          VIEW
P-ADAM-10-USAGE          L INDEX                                         INDEX
```

```
P-ADAM-10-USAGE          L THUMBNAIL                                    THUMBNAIL

P-ADAM-10-FIELD          L Object title                                 TITLE
P-ADAM-10-FIELD          L Storage directory                            F-DIRECTORY
P-ADAM-10-FIELD          L Object extension                             OBJECT-EXTENSION
P-ADAM-10-FIELD          L URL                                          U-PATH
P-ADAM-10-FIELD          L Usage type                                   USAGE-TYPE
P-ADAM-10-FIELD          L Cat. Sublibrary                              SUB-LIBRARY
P-ADAM-10-FIELD          L Note 1                                       NOTE-1
P-ADAM-10-FIELD          L Note 2                                       NOTE-2
P-ADAM-10-FIELD          L Note 3                                       NOTE-3
P-ADAM-10-FIELD          L Note 4                                       NOTE-4
P-ADAM-10-FIELD          L Note 5                                       NOTE-5
P-ADAM-10-FIELD          L Cataloger                                    CATALOGER
P-ADAM-10-FIELD          L Character Set                                CHARACTER-SET
P-ADAM-10-FIELD          L Guest                                        GUEST
P-ADAM-10-FIELD          L Copyright type                               COPYRIGHT-TYPE
P-ADAM-10-FIELD          L DigiTool PID                                 PID
```

Key to the table:

Column 1:	Internal field code
Column 2:	ALPHA Always “L“.
Column 3:	Description which is displayed as value in the dropdown list
Column 4:	Code for the entry in the dropdown list. Only internal codes can be used!

The several internal codes represent the following fields:

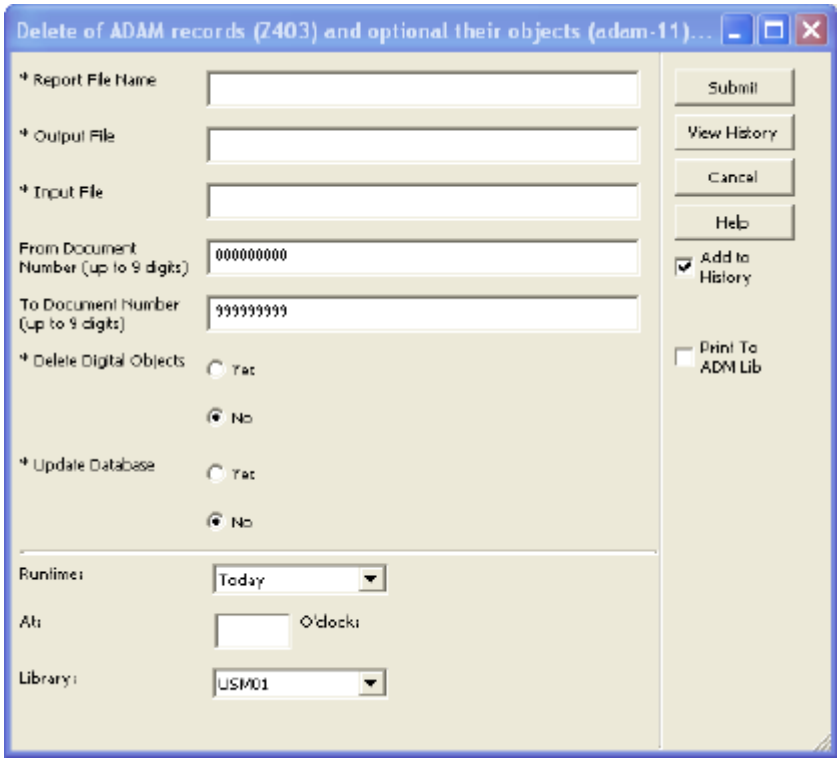
P-ADAM-10-SORT
The entries define the dropdown list for the sort type

P-ADAM-10-USAGE
The entries define the dropdown list for the different object types.

P-ADAM-10-FIELD
The entries define the dropdown list for the different fields.

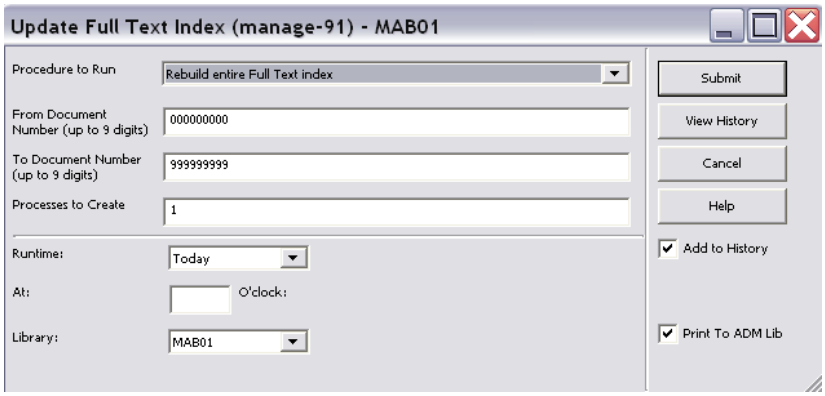
6.2.7 Deletion of ADAM records (Z403) (adam-11)

This service enables you to delete z403 records and their corresponding digital objects that are stored in the local file system. If the input file contains z403_rec_keys from records that have z403_usage_type=VIEW, the corresponding thumbnails and index files will be also deleted. The input file can be generated with p-adam-10.



6.2.8 Update Full Text Index (manage-91)

This service updates the Full Text Index of the database. This procedure locks the ALEPH system and should only be run when the library is closed.



There are two options on how to run this service:

- **Rebuild entire Full Text index** - rebuilds the entire Full Text index, deleting and rebuilding the tables.
- **Update Full Text index** - updates the Full Text index without deleting the tables.

7 Setup of Additional Oracle Tables

For managing digital objects in ALEPH, three additional Oracle tables have to be set up. The tables are called **z403**, **z91** and **z56**.

7.1 Z403

The technical metadata of the digital objects are handled by the z403 table. The z403 is created in the BIB library.

The following is the structure of the z403 table.

```
Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
 Z403_REC_KEY                              NOT NULL CHAR(15)
 Z403_DERIVED_FROM_SEQUENCE                         NUMBER(6)
 Z403_TITLE                                         VARCHAR2(500)
 Z403_F_DIRECTORY                                   VARCHAR2(100)
 Z403_F_FILENAME                                    VARCHAR2(100)
 Z403_ORIGINAL_FILE_NAME                            VARCHAR2(200)
 Z403_OBJECT_EXTENSION                              CHAR(5)
 Z403_OBJECT_SIZE                                   NUMBER(9)
 Z403_U_PATH                                        VARCHAR2(2000)
 Z403_USAGE_TYPE                                    VARCHAR2(20)
 Z403_SUB_LIBRARY                                   CHAR(5)
 Z403_NOTE_1                                        VARCHAR2(500)
 Z403_NOTE_2                                        VARCHAR2(500)
 Z403_NOTE_3                                        VARCHAR2(500)
 Z403_NOTE_4                                        VARCHAR2(2000)
 Z403_NOTE_5                                        VARCHAR2(2000)
 Z403_OPEN_DATE                                     NUMBER(8)
 Z403_UPDATE_DATE                                   NUMBER(8)
 Z403_CATALOGER                                     CHAR(10)
 Z403_CHARACTER_SET                                 VARCHAR2(100)
 Z403_COLOR_SETTING                                 VARCHAR2(100)
 Z403_RESOLUTION                                    VARCHAR2(100)
 Z403_DIMENSIONS                                    VARCHAR2(100)
 Z403_COMPRESSION_RATIO                             VARCHAR2(100)
 Z403_CREATION_DATE                                 NUMBER(8)
 Z403_CREATION_HARDWARE                             VARCHAR2(100)
 Z403_CREATION_SOFTWARE                             VARCHAR2(100)
 Z403_COPYRIGHT_CONTACT                             VARCHAR2(20)
 Z403_COPYRIGHT_OWNER                               VARCHAR2(100)
 Z403_COPYRIGHT_TYPE                                VARCHAR2(100)
 Z403_COPYRIGHT_NOTE                                VARCHAR2(2000)
 Z403_COPYRIGHT_NOTICE                              CHAR(1)
 Z403_COPYRIGHT_NOTICE_TYPE                         CHAR(10)
 Z403_DISPLAY_LINK                                  CHAR(1)
 Z403_DISPLAY_CODE                                  CHAR(5)
 Z403_EXPIRY_DATE                                   NUMBER(8)
 Z403_GUEST                                         CHAR(1)
 Z403_IP_ADDRESS                                    VARCHAR2(200)
 Z403_COURSE                                        VARCHAR2(200)
 Z403_USER_STATUS                                   VARCHAR2(50)
 Z403_RESTRICT_SUB_LIBRARY                          CHAR(5)
 Z403_NO_OF_COPIES                                  NUMBER(3)
 Z403_VIEW_TIME                                     NUMBER(3)
 Z403_ITEM_LIBRARY                                  CHAR(5)
 Z403_ITEM_DOC_NUMBER                               NUMBER(9)
 Z403_ITEM_SEQUENCE                                 NUMBER(6)
 Z403_ENUMERATION_A                                 VARCHAR2(20)
 Z403_ENUMERATION_B                                 VARCHAR2(20)
 Z403_ENUMERATION_C                                 VARCHAR2(20)
 Z403_CHRONOLOGICAL_I                               VARCHAR2(20)
```

```
Z403_CHRONOLOGICAL_J                              VARCHAR2(20)
```

7.2 Z91

The z91 table is also created in the BIB library, as well as in all other bases where the indexing process UE-01 is running. This table contains the indices of the fulltext indexing of text-based objects like MS Word or PDF files.

```
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 Z91_REC_KEY                               NOT NULL CHAR(9)
 Z91_FLAG                                           CHAR(1)
 Z91_FILE_NAME                                      VARCHAR2(256)
```

7.3 Z56

The z56 table is created in the vir01 library and limits access for the object by allowing only a certain number of simultaneous accesses of digital objects in the OPAC.
If an object is opened in the OPAC, a session will be registered and any further session will be compared to the entry in the field, z403_NO_OF_COPIES. If the defined limit is exceeded, the patron cannot open the object. A session will be automatically terminated after 10 min., if it is not terminated by the patron before that.

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------------
 Z56_REC_KEY                               NOT NULL CHAR(30)
 Z56_SESSION_ID                                     CHAR(50)
 Z56_TIME                                           CHAR(10)
 Z56_CLIENT_ADDRESS                                 VARCHAR2(20)
 Z56_CLIENT_TYPE                                    CHAR(1)
```

8 ADAM scan jobs

Note:
Using the Scan Interface is subject to purchasing a special license.

The following configurations are required in order to implement the ADAM Scan Jobs functionality:

- tab100

To activate the ALEPH Scan Interface, add the following parameter to the tab100 table of the BIB library:

CREATE-SCAN-JOB=Y

- pc_tab_exp_field.lng

The selection menus for scan parameters have to be added to the pc_tab_exp_field.lng table of the BIB library. For example:

```
!      1                              2     3                    4
!!!
```

```
OBJECT-RESOLUTION       L Low Resolution          low
OBJECT-RESOLUTION       L Medium Resolution     medium

OBJECT-TYPE             L Table of Contents       toc
OBJECT-TYPE              L Cover                    cov

OBJECT-COLOUR           L  Black & White (Text)   bw

SCAN-JOB-STATUS          L New                      NEW
```

- edit_doc_999.lng

To display the tag "SCJ" in the record's full view, a definition for SCJ tag has to be added to the edit_doc_999.lng table of the BIB library. For example:

```
## SCJ              D LScan Job          Y      E
```

- pc_display_lng/scan-job-expand

Scan Job Details are displayed in a separate window which can be opened from the Search Tab’s Full Record display, by double-clicking the tag SCJ. The display is based on XML format. Definitions are in the form "scan-job-expand".

- tab_expand

Information about the scan job must not be provided in internal format, but rather in external format, for example to replace the codes of the sublibrary into normal text. The new expand routine "expand_doc_bib_scj" must be used to create an external field (e.g. 076) which contains the following subfields:

```
$$a      Owner of the scan job (normal text)
$$b      Order Number
$$c      Object type
$$d      Color
$$e      Resolution
```

The "expand_doc_bib_scj" routine must be defined in the tab_expand table of the BIB library, for expand menu "WWW-X". The parameter CODE= can be used to define the external field which will contain the scan job details. For example:

```
!   1              2                              3
!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
```

9 Create Thumbnails

Thumbnails are created in the GUI by the ImageMagick software (which is OpenSource). This software is included in the Aleph Installation Kit.

System Librarian's Guide – Aleph ADM

Version 23

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 ALEPHADM CLIENT (ALEPHADM.INI) 4

1.1 ALEPHADM.INI Settings 4

2 TABLE_KEY 5

3 TAB_ALEPHADM.LNG 5

4 COLUMN HEADINGS (PC_TAB_COL.LNG) 6

1 ALEPHADM Client (ALEPHADM.INI)

The alephadm.ini file defines settings for the ALEPHADM client. This chapter presents and explains the following sections of the Alephadm.ini file.

1.1 ALEPHADM.INI Settings

[HelpWnd]
[ShowDifferences]
[TableMarkedLines]
[TablesNavigator]
[Tables]

Explanations of other setting definitions, relating to more than one module, can be found in the *.ini Files chapter of the General module of the ALEPH User Guide.

[HelpWnd]

```
[HelpWnd]
FontSize=14
```

This option determines the size of the font in the F1 Help window. The allowed range is 6-24. If the value is not within this range, the font size will default to 14.

[ShowDifferences]

```
[ShowDifferences]

ChangedLines="Changed Lines"
DeletedLines="Deleted Lines"
InsertedLines="Inserted Lines"

FgColorChangedLines=255,000,000
FgColorDeletedLines=000,000,255
FgColorInsertedLines=000,128,000
FgColorIdenticalLines=000,000,000

BkColorIdentical=255,255,255
BkColorDif=192,192,192
```

`ChangedLines="Changed Lines"`
The text between the quotation marks will appear at the bottom of the Show Differences window to refer to changed lines.

`DeletedLines="Deleted Lines"`
The text between the quotation marks will appear at the bottom of the Show Differences window to refer to deleted lines.

`InsertedLines="Inserted Lines"`
The text between the quotation marks will appear at the bottom of the Show Differences window to refer to inserted lines.

`FgColorChangedLines=255,000,000`
Sets the color for the changed lines that are displayed in the Show Differences window and for the text defined in `ChangedLines`.

`FgColorDeletedLines=000,000,255`
Sets the color for the deleted lines that are displayed in the Show Differences window and for the text defined in `DeletedLines`.

`FgColorInsertedLines=000,128,000`
Sets the color for the inserted lines that are displayed in the Show Differences window and for the text defined in `InsertedLines`.

`FgColorIdenticalLines=000,000,000`
Sets the color for identical lines that are displayed in the Show Differences window.

`BkColorIdentical=255,255,255`
Sets the background color for identical lines that are displayed in the Show Differences window.

`BkColorDif=192,192,192`
Sets the background color for changed, deleted and inserted lines that are displayed in the Show Differences window.

[TableMarkedLines]

```
[TableMarkedLines]
FgColor=000,128,000
BkColor=255,255,128
```

`FgColor=000,128,000`
Sets the color for marked lines that are displayed in the Tables Editing window.

`BkColor=255,255,128`
Sets the background color for marked lines that are displayed in the Tables Editing window.

[TablesNavigator]

```
[TablesNavigator]
WarningTimeOut=1
WarningTimeOut=1
```

Sets how many minutes before the end of the allowed editing time the green light should turn yellow in editing mode in the Tables Editing window.

2 TABLE_KEY

The TABLE_KEY line defines the columns which make up the "key" of a line in the table. The TABLE_KEY is used by the ALEPHADM client for display in FORM mode. List the columns that make up the key, separated by commas, no spaces.

For example, tab16 TABLE_KEY 1,2,3,4

3 tab_alephadm.lng

For files to appear in the ALEPHADM Navigator, they have to be listed in the `tab_alephadm.lng` table. This table is located in `xxxnn/tab/tab_alephadm.lng`.

For every library, there is a separate table. The table has a language suffix and is therefore language-sensitive.

```
! COL    1.  20; TEXT; ;
!            Directory;
!            Directory;
! COL    2.  30; TEXT; ;
!            Filename;
!            Filename;
! COL    3.   1; ALPHA {Y,N}; ;
!            Language extension;
!            Language extension:
!            (Y/N);
! COL    4.   1; ALPHA; ;
!            Alpha;
!            Alpha;
! COL    5.  50; TEXT; ;
!            Description;
!            Description;
```

Key to Table:

- Column 1: Denotes the directory under which the table files are stored.
- Column 2: Denotes the filename of the table files. This will be displayed in the ALEPHADM Navigator.
- Column 3: This is a flag that describes whether or not the table in the line has a language extension.
- Column 4: The standard value is "L". However, for CJK languages set to "C".
- Column 5: Contains a short description of the function of the table in the system. This description is displayed in the ALEPHADM Navigator.

The following setup is displayed in the Navigator as shown below:

```
!     1              2        3 4        5
!!!! ... !!!!!-!!!! ... !!!!!-!-!-!!!!! ... !!!!!!!!!!
tab            tab14          N L Item Reshelving Time
tab            tab15          Y L Item Statuses
```

(The table length has been shortened for display in the ALEPH User Guide.)

File Name	Description
tab14	Item Reshelving Time
tab15.eng	Item Statuses

4 Column Headings (pc_tab_col.lng)

Column headings for tables of information in the GUI ALEPHADM module are built in. You can change the width of a column from the GUI by dragging the column

border with the mouse pointer. The server file `pc_tab_col.lng` has no influence on the column headings of the ALEPHADM module.

System Librarian's Guide - Cataloging

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: December 24, 2017

Web address: http://www.exlibrisgroup.com

Table of Contents

1 RECORD FORMATS 9

2 TEMPLATES 9

2.1 Creating Local Templates 10

2.2 Creating Library-dependent Templates 10

3 VALID FIELDS 11

4 FORMS 13

5 LIST OF VALUES IN FIXED-LENGTH FIELDS FORMS 19

5.1 Defining Lists of Valid Values and Description 21

5.2 Load the Lists of Valid Values into Aleph Data 22

5.3 Setting up the "GUI Fixed-Length Fields Forms" 24

6 DEFAULT SUBFIELDS 25

7 DEFAULT FIELDS FOR NEW RECORD 26

8 TAG INFORMATION 27

9 SEARCH HEADINGS 29

10 SEARCH SUBFIELD OPTIONS 31

11 CHECK FIELD 33

11.1 AL Section 33

11.2 D section 35

12 FIX RECORD 36

12.1 tab_fix 36

12.2 fix_doc.lng 74

12.3 fix_doc_track 75

13 LOCATE FUNCTION 75

14 DUPLICATE RECORD FUNCTION 77

15 IMPORTING UPDATED TABLES 78

16 FLOATING KEYBOARD 78

17 AUTHORIZATIONS 82

17.1 Allowed and Denied Tags 82

17.2 Cataloging "OWN" Permissions 83

17.3 Holdings Filter 84

18 MERGING RECORDS 84

19 UPDATING THE TABLES PACKAGE 87

20 SUBFIELD PUNCTUATION 87

21 VALIDATION OF CONTENTS OF A FIELD 88

22 CHECK FIELD OCCURRENCES AND DEPENDENCY BETWEEN FIELDS 89

23 FORBIDDEN ERRORS AND TRIGGERS 91

24 CHECKING ROUTINES FOR NEW HEADINGS IN THE HEADINGS LIST 93

25 CHECKING ROUTINES FOR NEW HEADINGS IN THE BIBLIOGRAPHIC AND AUTHORITY HEADINGS LIST 93

26 CHECKING ROUTINES FOR NEW DIRECT INDEXES (IND) 94

27 LOCKING RECORDS 95

27.1 Locking Period for Locked Records 95

27.2 Lock Status Message 95

28 CHECK ROUTINES FOR CHECK RECORD 95

28.1 Check Types Available for Column 1 of the check_doc Table: 96

28.2 Check Programs Available for Column 2 of the check_doc Table 96

28.3 Check Programs For Document Deletion 100

29 FIXED-LENGTH FIELDS CHECKING ROUTINES 101

30 VALIDATION MESSAGES (TABLE-DEPENDENT) 105

31 VALIDATION MESSAGES (SYSTEM-DRIVEN) 105

32 CATALOGING PRODUCTIVITY REPORT 106

32.1 HOL Records tab of Records Editor 106

33 COLUMN HEADINGS (PC_TAB_COL.LNG AND TAB_COL.DAT) 107

34 DEFAULT VALUES FOR FIXED FIELDS IN NEW RECORDS 107

35 IMPORTING RECORDS 108

35.1 Remote Conversions 109

36 COMBINING DIACRITICS 109

37 RECORD LENGTH LIMITS 110

38 HIDDEN FIELDS 110

39 RECORD MANAGER 111

40 OVERVIEW TREE 111

41 SETTING UP A SCRIPT FOR THE CORRECTION OF RECORDS IN ALEPH SEQUENTIAL FORMAT 116

41.1 Generic Fix Doc Script Specification 116

41.2 Script Flow 116

41.3 Generic Fix Doc Operations 117

CONCATENATE-FIELDS 119

41.4 Generic Fix Doc (p_file_08) Script Examples 124

42 CLIENT SETUP (CATALOG.INI) 126

42.1 Catalog.ini Settings 126

42.1.1 [ConvertFile] 126

42.1.2 [Form] 128
42.1.3 [Editor] 128
42.1.4 [ExpandTemplate] 130
42.1.5 [DuplicateRecord] 130
42.1.6 [OffLine] 131
42.1.7 [Locate] 131
42.1.8 [Scan] 131
42.1.9 [HolOwnTextDefaults] 131
42.1.10 [General] 132
42.1.11 [RecordBar] 132
42.1.12 [RecordTree] 132
42.1.13 [RfidMedia] 133
42.1.14 [LOW] 133

43 CATALOGING TABLES 134

43.1 Library Tables 134

44 SETTING UP THE LKR FIELD 138

44.1 tab_fix_z103 138

45 SUPPORTING ADDITIONAL FILTERS IN LKR FIELD 140

46 LKR UPDATING UPON ITEM ENUMERATION AND CHRONOLOGY MODIFICATION 141

47 TAB100-RELATED ENTRIES IN CATALOGING 143

48 SETUP OF ADM LIBRARIES 147

49 MATCHING RECORDS 147

50 SETTING UP SERVICES 151

50.1 Retrieve Catalog Records (ret-01) 151

51 CJK UNICODE CHARACTERS 151

52 PUBLISHING 152

52.1 Initial Extract Process 152

52.2 Ongoing Extract Process 153

52.3 Name Spacing in Publishing 155

53 UPLOAD BIB AND HOLDING INFORMATION FROM ALEPH TO KERIS 155

53.1 Tables Set-Up Configuration 156
53.1.1 KERIS Z39.50 Gate Configuration 156
53.1.2 Expand and Fix Routines Setup 157
53.1.3 Manual Upload Using the Remote Menu of Cataloging Module 161
The Remote menu of the Cataloging module uploads a single document to KERIS. It supports the following updates: 161
53.1.4 Upload New and Updated Document Manually 161
53.1.5 Upload Deleted Document Manually 163
53.1.6 Upload Suppressed Document Manually 163
53.1.7 Upload Repaired Document Manually 164
53.1.8 Bulk Upload Using Batch Service 165
53.1.9 The Batch Output Reports 167

54 OUF LOADER 167

54.1 Instructions for Running the OUF Loader 168
54.1.1 Running from the UNIX Prompt 168
54.1.2 Parameters Description 168

55 PREVENTING THE AUTOMATIC CREATION OF PAR RECIPROCAL LINKS BETWEEN RECORDS 170

56 GENERATING A LOCALLY ASSIGNED CALL NUMBER IN BIBLIOGRAPHIC AND ITEM RECORDS 170

56.1 Creating the Call Number in the BIB Record 170

56.2 Check Routine for Call Number Prefix 171

57 AUTOMATIC CREATION OF 6XX FIELDS 172

57.1 The Batch Service: Create Additional Subject Heading(s) from Authority (manage-46). 172

57.2 Defining the AUT Index code for Detecting the AUT Headings - tab_bib_aut_match 174

57.3 Manage-46 Service Functionality - Workflow and Example 175

57.4 Match Algorithm and Translate 176
57.4.1 Match and Translate for Create 6XX Using 1XX 176
57.4.2 Match and Translate for Create 6XX Using 7XX 179

58 AUTOMATIC TRANSLATION OF BIBLIOGRAPHIC NOTE FIELDS 181

58.1 Fix Routine for Translation - fix_doc_notes 181

58.2 Setting Up a List of Translations - tab_fix_notes 181

58.3 Automatic Translations – Functionality and Examples 182
58.3.1 Compare Action 182
58.3.2 Replace Action 182

59 LINK TO RDA TOOLKIT.....183

60 UPDATE HOL RECORD BASED ON ITEM ARRIVAL.....184

1 Record Formats

In ALEPH, every record in the system must be assigned a Record Format. This information is kept in the FMT field which is an ALEPH-specific field. In the Cataloging module, the content of the FMT field (in other words the record format) of the record in the Catalog Editor is displayed in the Cataloging bar. The record format is consulted by various functions in the system, such as the display of online tag information, checking procedures, input forms, templates, and so on. For example, the values in the MARC 21 008 field depend on the record's format, and therefore there are separate input forms according to the format of the record (a form for BK, a form for SE, and so on). Although formats can be added, this is not recommended because of the amount of setup and upkeep this entails. Note that for more granularity in "format type", you can use the expand_doc_type expand program to create a TYP field with values which can be used for indexing and display in addition to or instead of the FMT.

The list of the available record formats is defined in the `formats.lng` table located in the library's `pc_tab/catalog` directory.

Following is a sample of the `formats.lng` 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
```

Key to Table:

Column 1 - Code
This is the unique code by which the system identifies the format. The code must be two characters long. This code is displayed in the Cataloging bar of the Cataloging module for the record in the Catalog Editor.

Column 2 - ALPHA
ALPHA code. Must always be L.

Column 3 - Description
Enter a description for the format. This can be up to 20 characters long.

2 Templates

There are two types of cataloging templates: local templates and library-dependent templates. Local templates are only available for the station on which they are created. Library-dependent templates are available to all librarians cataloging in the library.

2.1 Creating Local Templates

To create a **local template**, perform the following steps:

Open a cataloging record to serve as the basis for your local template.

Open the Cataloging menu and choose Create Template on Local Drive.

Enter the template file name and click OK. A message appears confirming that the template has been saved. The template is now available in the list of templates, and can be chosen from the Cataloging menu using the Open Template option. This template can now be used as a basis to catalog a new record.

Note that local templates are saved on the PC in the following folder of the GUI: *../Catalog/Template*.

2.2 Creating Library-dependent Templates

To create a **library-dependent** template, perform the following steps:

Add a file containing the template to the `$data_root/pc_tab/catalog` directory. Give the file the extension `.mrc`.

Make sure that the file contains the following values in the following columns:

Column 1: Field tag code and indicators
In addition, there are three codes for the use of the system: DB and SYSID and FMT. FMT is for the code of the record format.

Column 2 - ALPHA
ALPHA code. Must always be L.

Column 3 - Subfield codes and contents
Subfield codes are prefixed by two dollar signs ($$). The system codes must be defined as follows:

- The value of DB is always LOCAL.
- The value of SYSID is always 0.
- The value of FMT is the code of the record format.

Following is a MARC 21 sample template for books:

```
DB    L LOCAL
SYSID L 0
FMT   L BK
LDR   L ^^^^^nam^^22^^^^^^u^4500
008   L ^^^^^^s2000^^^^^^^^^^^^r^^^^^000^0^eng^d
020   L $$a
040   L $$a
080   L $$a
1001  L $$a $$b $$c $$d
2401  L $$a
2451  L $$a $$b $$c $$h
24611 L $$i $$a$$b
250   L $$a $$b
260   L $$a $$b $$c
300   L $$a
440   L $$a $$n $$p $$v
500   L $$a
```

```
502   L $$a
5050  L $$a
650 2 L $$a
690   L $$a
7001  L $$a $$b $$c $$d
7101  L $$a $$b
7102  L $$a $$b
7112  L $$a $$n $$c $$d
740   L $$a $$h
```

An alternative easy way to create a library-dependent template is to create a template locally, and then to make it available for the entire library. First create an appropriate template on the PC (local template) and then FTP the file to the `$data_root/pc_tab/catalog` directory. This ensures the proper placement of values and the functionality of the template.

Library-dependant templates are listed, together with the local templates, in the List of Templates window. Note that only those templates defined for the home library to which the cataloger is connected are available. For example, if the librarian is connected to a bibliographic library - XXX01 - the templates defined for this library are displayed in the window. If the librarian is connected to an authority library - XXX10 - templates defined for this library are listed in the List of Templates window.

3 Valid Fields

The system librarian is in charge of defining the valid tags and aliases for the Cataloging client. The list of valid tags and aliases is activated in the Cataloging module by using the hotkey F5 or by selecting the New field (choose from list) option from the Edit menu. Valid tags and aliases are defined by editing the `codes.lng` table, located in the library's `pc_tab/catalog` directory.

The "codes" table enables you to define the following three aspects:

- Defines whether the tag is displayed in the list of tags available in the Cataloging module.
- Defines whether the tag can be edited only through a form or through a form and directly on the catalog draft.
- Defines whether or not the tag can have subfields.

Following is an example from the table:

```
!1    2 3 4 5      6          7                    8
!!!!!-!-!-!-!-!!!!!!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

LDR   N Y Y L Leader          L Leader
001   Y Y Y L Control No.     L Control Number
003   Y N Y L Control No. ID  L Control Number Identifier
005   Y N Y L Date and Time   L Date and Time of Last Transaction
006   Y Y Y L Linking Field   L Linking Field
007   Y Y Y L Phys.Descrip.   L Physical Description Fixed Field
008   Y Y Y L Fixed Data      L Fixed Length Data Elements
010   Y N N L LC Control No.  L Library of Congress Control Number
013   Y N N L Patent Info.    L Patent Control Information
```

```
016   Y N N L Nat.Bib. No.    L National Bibliography Number
```

Key to Table:

Column 1 - Field tag
Enter a field tag up to 5 characters long. The # character can be used as a placeholder for indicators in positions 4 and 5. For example, entering *245##* includes *2451*, *2452* or *24514*. Fields in the "codes" table must first be defined in the "Codes and names of MARC and ALEPH fields" table (this is done by editing the tab01.lng table of the library's tab directory).

Column 2 - Display
This column defines whether the tag can be chosen from the list of tags available in the Cataloging module. Values are *Y* and *N*. If you choose *Y*, the tag is displayed in the list. If you choose *N*, the tag will not be displayed in the list but it is still possible to add manually to the catalog draft.

Column 3 - Edit
This column defines whether the tag can be edited only through an editing form or if it is possible to edit the tag both from a form and directly from the catalog draft. Values are *Y* and *N*. If you choose *Y*, you can only edit the tag through a form. If you choose *N*, you are able to edit the tag either from a form or directly in the cataloging draft.

If Column 3 is set to 'N', then the setup of Column 4 influences whether or not the form is editable. If Column 4 is set to 'N' (that is, the field can have subfields), then form editing is possible. If it is set to 'Y' (that is, the field cannot have subfields), then the form will not be available for editing.

Note that this option should be used in conjunction with the setup of the forms. If the field is a fixed-length field and the form for the field is set as a form for fixed-length fields, enter *Y*. If you want the cataloger to edit a fixed-length field from the catalog draft, then the form for the field should be defined as a form for a non fixed-length field, and this column should be set to *N*.

If a form is not available for a field, the option to edit the field through a form will not be available.

Column 4 - Subfields
This column defines whether or not the tag can have subfields. Values are *Y* and *N*. If you choose *Y*, the cataloger will not be able to add subfields to the field. If you choose *N*, the cataloger is able to add subfields to the field.

Column 5 - ALPHA of name tag
ALPHA code. Must always be L.

Column 6 - Catalog name tag
The catalog name tag (also called an alias) can be up to 16 characters long. It can be the same name tag defined for the field in the tab01.lng table, although it does not have to be. Name tags are displayed in the catalog record in addition to the (usually numeric) field tags ONLY if the DisplayTagInfo field of the PC's catalog.ini file is set to *Y*.

Column 7 - ALPHA of description
ALPHA code. Must always be L. The description is displayed in the list of valid tags available in the Cataloging module.

Column 8 - Description
Enter a description of the field, up to 38 characters long. The description is displayed in the list of valid tags available in the Cataloging module by using the hotkey F5 or choosing the New field (choose from list) option from the Edit menu.

4 Forms

You can define the forms that catalogers use to enter data for a field. The files that define forms are located in the library's $data_root/pc_tab/catalog directory.

The file format for cataloging forms can include up to 5 characters for defining the tag and its indicators. The possible formats for the form are the following:

nnn_xx.lng is in use for undefined indicators.

nnny_xx.lng or **nnnyy_xx.lng** are in use for specific indicators.

y can be used to define a specific first indicator.

yy can be used to define specific indicators.

xx is the code for the record format, for example, BK for book. Refer to Record Formats on page 9 for more information on record formats.

lng is the code for the language (for example, *eng* for English).

The **exceptions** are for forms for the MARC 21 field 007 and the MARC 21 field 006. They are in the format **007_x.lng** and **006_x.lng**, where x is the code for the material type, as defined in the files scr_007.lng and scr_006.lng (which are also located in the library's $data_root/pc_tab/ catalog directory.

Sample Form #1
Following is a sample form for MARC 21 field 260:

```
^1^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

    ^2^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^&2
    ^3^^^^^^^^^^^^^^^^^^
    ^4^^^^^^^^^^^^^^^^^^

    ^5^^^^^^^^^^^ _30_____________________________

    ^6^^^^^^^^^^^ _30_____________________________

    ^7^^^^^^^^^^^ _30_____________________________

###
F abc

^1=Imprint (NR)
^2=Indicators
^3=First indicator (blank,2,3)
^4=Second indicator (blank)
```

```
^5=Place of publication (a)
^6=Name of publisher (b)
^7=Date of publication (c)
```

Sample Form #2

Following is a sample form for MARC 21 fixed-length field 008:

```
^1^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

 ^2^^^^^^^^^^^ $6$$$$           ^3^^^^^^^^^^^ _1
 ^4^^^^^^^^^^^ _4___            ^5^^^^^^^^^^^ _4___
 ^6^^^^^^^^^^^ _3__             ^7^^^^^^^^^^^ _1 _1 _1 _1
 ^8^^^^^^^^^^^ _1               ^9^^^^^^^^^^^ _1
 ^10^^^^^^^^^^ _1 _1 _1 _1      ^11^^^^^^^^^^ _1
 ^12^^^^^^^^^^ _1               ^13^^^^^^^^^^ _1
 ^14^^^^^^^^^^ _1               ^15^^^^^^^^^^ _1
 ^16^^^^^^^^^^ _1               ^17^^^^^^^^^^ _1
 ^18^^^^^^^^^^ _3__             ^19^^^^^^^^^^ _1

 ^20^^^^^^^^^^ _1

###
D ^^^^^^s2003^^^^^^^^^^^^^^^^^^000^^^eng^d
V

^1=008 Fixed length data elements (BOOKS)
^2=Date entered on file (00-05)
^3=Type of date (06)
^4=Date 1 (07-10)
^5=Date 2 (11-14)
^6=Publication Place (15-17)
^7=Illustration codes (18-21)
^8=Target audience (22)
^9=Form of item (23)
^10=Nature of contents (24-27)
^11=Govt.publication (28)
^12=Conference publ. (29)
^13=Festschrift (30)
^14=Index (31)
^15=Unspecified (32)
^16=Literary form (33)
^17=Biography (34)
^18=Language (35-37)
^19=Modified record (38)
^20=Cataloging source (39)
```

Key to Form

In general terms, the files for the forms are divided into three main sections:

The upper section of the file. This section is used to define the display of the various elements of the form and to define the actual length for the input fields. The following example is from the form for the 260 field:

```
^1^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

    ^2^^^^^^^^^^^^^^^^^^^
```

```
^4^^^^^^^^^^^^^^^^^^

^5^^^^^^^^^^ _30_____________________________

^6^^^^^^^^^^ _30_____________________________

^7^^^^^^^^^^ _30_____________________________

  ###
```

The additional values section. The following example is from the form for the 260 field:

```
F abc
```

The lower section of the file. This section is used to define the actual text that is displayed. The following example is from the form for the 260 field:

```
^1=Imprint (NR)
^2=Indicators
^3=First indicator (blank,2,3)
^4=Second indicator (blank)
^5=Place of publication (a)
^6=Name of publisher (b)
^7=Date of publication (c)
```

Visual Text Definitions (Upper and Lower Sections)

Each line in the lower section of the file should have a matching line in the upper section. In both sections, each line must begin with the caret (^) sign followed by a number. The match between the lines is performed according to this number as shown in the examples below. Note that in the lower section, an equal sign (=) is added after the matching number. To summarize, the lower section of the forms determines the text to be displayed in the position determined by the matching line in the upper section.

Based on the form for the 260 field, the text *Imprint (NR)* is displayed at the top of the form. The following line from the lower section:

```
^1=Imprint (NR)
```

matches the following line in the upper section of the file:

```
^1^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

The text *Indicators* is displayed below *Imprint (NR)*. The line:

```
^2=Indicators
```

matches the following line in the upper section:

```
^2^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^&2
```

The upper section not only determines the position of the text (in other words, of the lines in the lower section); it is also used to determine the length of the display text. This is done by the caret signs that follow the matching number. For this reason, when creating a new form, it is important to include enough caret signs in the line of the upper section, or the text will be cut when displayed in the Cataloging module. Note

that the caret signs are not in a relationship of one-to-one with the characters in the text line of the lower section. This occurs especially with proportional fonts, where the character width varies and the characters do not occupy the same amount of horizontal space. For example, the first line in the upper section of our sample file contains two caret signs after the matching number:

```
^1^^
```

Since the width of each letter is proportional to its shape and an *i* is narrower than a *w* in the Cataloging module, the two caret signs display up to six *w*'s and up to twenty-four *i*'s for the Tahoma font. For Courier New fixed font, up to five *w*'s and five *i*'s are displayed.

When creating new forms, it is best to proceed on a trial-and-error basis. The font for the text displayed in the form is defined in the font.ini file of the `ALEPHCOM/TAB` directory under the WindowControls section. Note that this section also determines the font for many other instances in the client such as buttons, edit fields, static fields and more.

Input Fields Definitions (Upper Section)

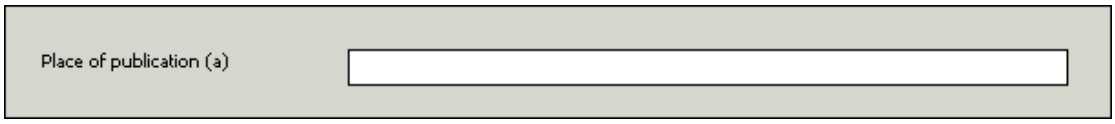
In addition to visual text definitions, the input fields for the forms are defined through the upper section of the file. The following line from the form for the 260 field, includes the text defined in line 5 of the lower section and the input field definitions:

```
^5^^^^^^^^^^ _30_____________________________
```

The lower section of the file contains the following line:

```
^5=Place of publication (a)
```

This is displayed as follows in the Cataloging module:



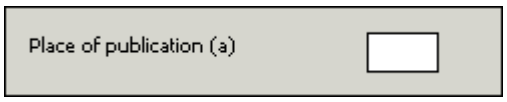
The input field is achieved by the following line:

```
_30_____________________________
```

The line must begin with an underscore (_) character. A number that determines the actual length assigned to the subfield (in this case, to subfield $a of the 260 field) must follow the underscore (in this example, up to 30 characters can be entered in the input field for subfield $$a). After the number - as with the caret sign for text - it is necessary to define the visual definitions for the input field. This is done by adding underscore characters that represent the visual display of the field. Note that the visual length includes the positions defined by the first underscore and by the following number. For example, if the line is defined as follows:

```
_30
```

then the input field is displayed in the Cataloging module as follows:

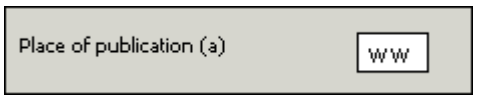


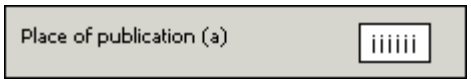
The cataloger can still type up to 30 characters but only some of the characters will be displayed (the number of characters displayed is relative to the font).

The relationship between the characters displayed in the input field and the number of underscores is not one-to-one (as with the caret sign and the characters for the text). Again, when creating new forms, it is best to proceed on a trial-and-error basis. For example, if the line is defined as follows and the font used is Tahoma:

```
_30
```

then in the Cataloging module, only up to two *w*'s will be displayed at a time, but up to four *i*'s can be displayed at a time:





The font for the input fields of the form is defined in the font.ini file of the `ALEPHCOM/TAB` directory under the **WindowControls** section. Note that this section also determines the font for many other instances in the client such as the Browse and Find edit fields.

Upper Section - Additional Information

Indicators: For fields that have indicators, the line for indicators must appear before the lines for text input. The ampersand (&) sign and the number 2 are used to indicate the two positions for the indicators. The following example is from the form for the 260 field:

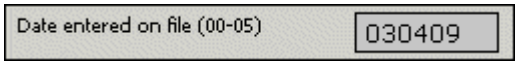
```
^2^^^^^^^^^^^^^^^^^^^^^^^^&2
```

Dollar ($) sign: Indicates that the information is displayed on the screen, but the user cannot change it. The following example is from the form for the 008 field:

```
^2^^^^^^^^^^^ $6$$$$
```

This line is used for positions 00 to 05 of the 008 field and contains the date entered on file. This information is added automatically by the fix_doc_tag_008_open_date fix program. Before applying the fix, this input field is displayed as follows in the form:

Following is the display of this input field in the form after the fix program inserted the date:



Similar to the specifications for a standard field, the first character is a dollar sign (instead of an underscore). Following this first dollar sign, the line includes the actual length of the positions that are in display-mode only - in this case 6 (positions 00 to 05). After the number, it is necessary to define the visual definitions for the input field. This is done by adding additional dollar signs that represent the visual display of the field. Note that in order for the dollar signs to work, an underscore has to exist in the form for another character in another position. In other words, the form must include a character with an underscore in order for the dollar signs to work.

###: Three hash signs are used to specify the end of the upper section of the field.

Additional Values Section

The additional values section divides the upper and lower sections of the file. The following example is from the form for the 260 field:

```
F abc
```

The following example is from the form for the 008 field:

```
D ^^^^^^s2003^^^^^^^^^^^^^^^^^^000^^^eng^d
V
```

Following are the available options for this section:

F - The *F* line is used for non-fixed-length fields to indicate which subfields are defined in the form. For example, the following line:

```
F abc
```

specifies that the form includes subfields $a, $b and $c. Make sure that you leave a space between the *F* and the beginning of the first subfield. Do not leave a space between subfields.

D - The *D* line is used for fixed-length fields to indicate the length of the field and the default values for the form. For example, in the following line:

```
D ^^^^^^s2003^^^^^^^^^^^^^^^^^^000^^^eng^d
```

the form for the 008 field is set to 40 character positions and, as an example, the default language in the form (positions 35 to 37) is set to "eng" (English).

Note that the whole field length needs to be specified. For this reason, positions that are set as undefined, must be marked by using the character that is used to denote a blank in fixed-length fields. In the above example, the caret (^) is used. This is the

standard for MARC 21 libraries. The following example is from the 100 UNIMARC field:

```
D --------d--------km-y0enga0103----ba
```

In the above example, the character used to denote blanks is the hyphen (-). The character used is defined in the DOC-BLANK-CHAR variable in the tab100 table of the library's tab directory.

V - The *V* indicates that the form should be verified for correctness before letting the user leave the form. If errors were found, the error messages are shown in the Messages tab of the lower pane and the system displays a prompt informing the cataloger that checking the field reported warnings and asking if he still wants to continue with the closure of the form.

S - The *S* line is used for fixed-length fields that have subfields (for example, the 100 UNIMARC field). This type of line is used to define the subfields for the form and the length of each subfield. The following is a sample line from the form for the 100 UNIMARC field:

```
S a(36)
```

The above line indicates that the form contains subfield $a of size 36. The following is an additional example:

```
S a(11)b(7)c(12)d(9)
```

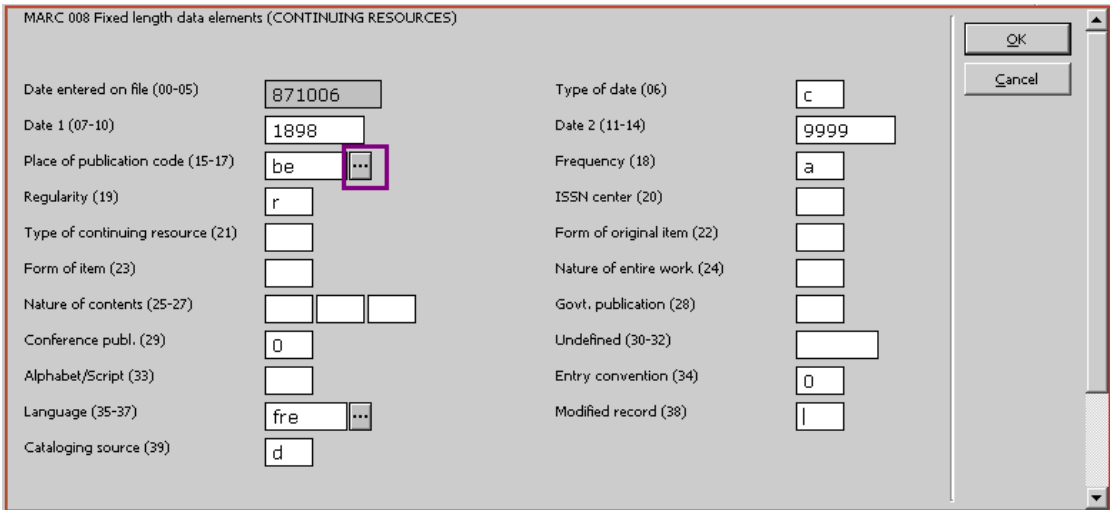
The above line indicates that the form contains subfields $a, $b, $c, and $d. Subfield $a is of size 11, subfield $b is of size 7 and so on.

5 List of Values in Fixed-Length Fields Forms

Depending on your system set-up, the cataloging forms of fixed-length fields can contain an option to display a list of valid values for certain positions.

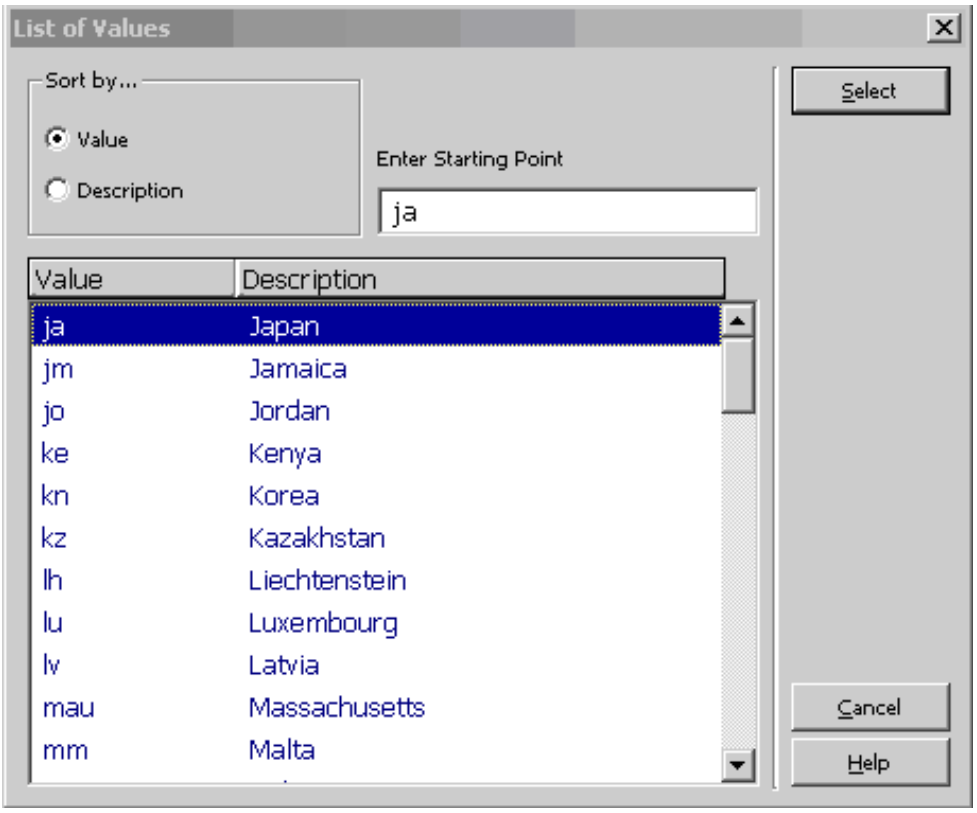
Each library decides for which forms and for which positions the Valid Value dialog box should be available.

Below is a sample of the MARC 008 Fixed Length Form with an expand button next to the Place of Publication Code field (008 positions 15-17):



Clicking the expand button displays the List of Valid Values dialog box, which contains a list of pre-defined values and descriptions for certain tags and positions.

Below is a sample of List of Values for tag 008 positions 15-17: Place of Publication:



The content of the List of Values is determined by the library's setup. The staff user can navigate the list and select an entry to be populated in the relevant position of the fixed length form.

Perform the following in order to configure the display of the List of Values dialog box for fixed-length fields.

1) Define lists of valid values and descriptions.
2) Load the lists of valid values into Aleph data.
3) Set up the "GUI Fixed-Length Fields Forms" to call up the relevant list.

5.1 Defining Lists of Valid Values and Description

The list of valid values and their description is defined in configuration tables.

The table file names are determined by the library. The language extension <lng> must be part of the file name. There should be separate tables for each language.

The Exlibris sample demo table is located in ./usm01/tab directory and is named: tag_values.eng.

Table structure:

Col.1 – Identifier for the list (upper case string, up to 30 characters). The identifier represents a list of values and description for certain material type/tag/position. This identifier should be set in the 'Fixed-Length Form' file. Each field in the 'Fixed-Length Form' that calls the specific identifier lists all Values and Descriptions of the quoted identifier. Each library determines its own identifier codes.

Col. 2 – Value that is valid for the identifier (up to 10 characters)

Col.3 – Description of the value (up to 50 characters)

The following are a few samples of possible identifiers:

Identifier: PLACE-OF-PUBLICATION – Used for setting values and descriptions for tag 008 positions 15-17 (all material types).

Identifier: ILLUSTRATION-CODE-BK – Used for setting values and descriptions for tag 008 positions 18-21 (material type: BK).

Identifier: FORM-OF-COMPOSITION-MU – Used for setting values and descriptions for tag 008 positions 18-19 (material type: MU).

Below is a partial sample from Exlibris demo tables ./usm01/tab/tag_values.eng. This sample defines valid values and descriptions for the above identifiers.

```
! 1                              2                    3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
PLACE-OF-PUBLICATION           aa         Albania
PLACE-OF-PUBLICATION           abc        Alberta
PLACE-OF-PUBLICATION           -ac        Ashmore and Cartier
PLACE-OF-PUBLICATION           ae         Algeria
!
ILLUSTRATION-CODE-BK           a          Illustrations
ILLUSTRATION-CODE-BK           b          Maps
ILLUSTRATION-CODE-BK           c          Portraits
ILLUSTRATION-CODE-BK           d          Charts
```

```
!
FORM-OF-COMPOSITION-MU          an          Anthems
FORM-OF-COMPOSITION-MU          bd          Ballads
FORM-OF-COMPOSITION-MU          bg          Bluegrass music
FORM-OF-COMPOSITION-MU          bl          Blues
```

The content of the tag value tables must be loaded into Aleph data (Z112 Oracle table). For details, see the Load the lists of valid values into Aleph data section.

NOTES:

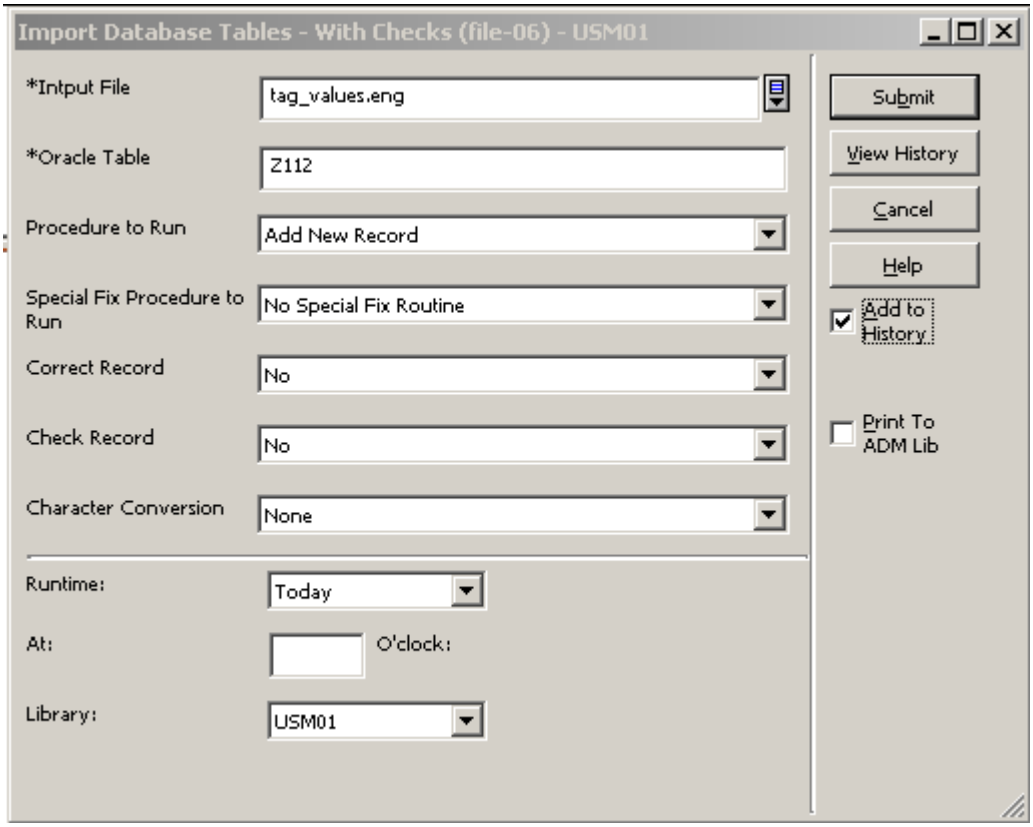
- The configuration table file names can be decided per site. The demo suggested file name (tag_values.eng) is not mandatory. A library may name the files differently. The language extension <.lng> must be part of the file name.
- A library may decide either to have a single table for all valid values and descriptions (e.g. file name: ./xxx01/tab/tag_values.eng), to have a set of tables per tag (e.g.: 006_values.eng, 007_values.eng and 008_values.eng), or to have tables per tag and format (e.g.: 006_se_ values.eng, 006_bk_valuers.eng, 008_se_values.eng, etc).
- Different MARC formats (like MARC21 and KORMARC) hold their set of tables at the relevant BIB library so that different values can be set to different formats.

- The valid values and the description may contain characters with diacritics, such as the Ä, É, Ö, Ü letters (with umlaut or accent mark) in the words: Ägypten, Égypte, Österreich, and Türkiye. In order to sort these letters in the right order, for example, "Ä" after "A" and not after "Z", add in `./alephe/unicode/tab_character_conversion_line` a line with type "`SORT-Z112`", using a table in which the diacritics have fallback characters without diacritics. For example:

```
SORT-Z112           ##### # line_utf2line_utf            unicode_to_word_gen
```

5.2 Load the Lists of Valid Values into Aleph Data

Once the identifiers are set in the valid values configuration tables (e.g. ./usm01/tab/tag_values.eng), their content should be loaded into the BIB XXX01 Aleph Oracle table: Z112 (Fixed Length Tag Values table). The load is performed by using the batch service: "Import Database Tables - With Checks (file-06)".



Input file:
The file-06 service requires that the input file is located in the library's DATA_FILES directory (e.g.: /xxx01/files). This mean that the valid value tables should be copied to DATA_FILES before submitting file-06. Alternatively, the user may place the exact path from DATA_FILES to the file location (.e.g.: ../tab/tag_values.eng) in the "Input File" field.

Oracle table:
Should be : "Z112".

Procedure to Run
Must be: "Add New Record".

Special Fix Procedure to Run:
Should be: "No Special Fix routine".

Correct Record and **Check Record**
Should be: "No".

Character Conversion
For a non-Latin input file (e.g.: tag_values.kor), select the required Character Conversion routine to be applied on the imported Z112 table.

Note: Each time a change is made in the valid values configuration table/s, it is recommended to drop Z112 data (using util A/17/1) and re-load the content of the configuration table/s.

5.3 Setting up the "GUI Fixed-Length Fields Forms"

Once the list of identifiers is set in the valid values tables, and the data is loaded into Z112 Oracle table, the GUI "Fixed Length Forms" should be amended to call the relevant identifiers. This causes the relevant list to be called-up when a "Fixed-length Form" is opened in the Cataloging module.

For this purpose, each of the forms that should contain list of valid values and descriptions must be adjusted to contain the relevant identifier. The cataloging form directory is: ./xxx01/pc_tab/catalog.

For example:

To enable the values and description list in 008 form of BK format, the file ./xxx01/pc_tab/catalog/008_bk.eng must be set with relevant identifiers. The identifier should be set next to the fields. Two percent sign (%%) should precede the identifier code as in the sample below of 008_bk.eng (the add-on elements are in **bold** text).

```
 ^1^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

 ^2^^^^^^^^^^^ $6$$$$           ^3^^^^^^^^^^^ _1
 ^4^^^^^^^^^^^ _4___            ^5^^^^^^^^^^^ _4___
 ^6^^^^^^^^^^^ _3__             ^7^^^^^^^^^^^ _1 _1 _1 _1
 ^8^^^^^^^^^^^ _1               ^9^^^^^^^^^^^ _1
 ^10^^^^^^^^^^ _1 _1 _1 _1      ^11^^^^^^^^^^ _1
 ^12^^^^^^^^^^ _1               ^13^^^^^^^^^^ _1
 ^14^^^^^^^^^^ _1               ^15^^^^^^^^^^ _1
 ^16^^^^^^^^^^ _1               ^17^^^^^^^^^^ _1
 ^18^^^^^^^^^^ _3__             ^19^^^^^^^^^^ _1
 ^20^^^^^^^^^^ _1
D ^^^^^^s2003^^^^^^^^^^^^^^^^^^000^^^eng^d
V

^1=008 Fixed length data elements (BOOKS)
^2=Date entered on file (00-05)
^3=Type of date (06)
^4=Date 1 (07-10)
^5=Date 2 (11-14
^6=Publication Place (15-17)%%PLACE-OF-PUBLICATION
^7=Illustration codes (18-21)%%ILLUSTRATION-CODE-BK
```

```
^8=Target audience (22)
^9=Form of item (23)
^10=Nature of contents (24-27)
^11=Govt.publication (28)
^12=Conference publ. (29)
^13=Festschrift (30)
^14=Index (31)
^15=Unspecified (32)
^16=Literary form (33)
^17=Biography (34)
^18=Language (35-37)
^19=Modified record (38)
^20=Cataloging source (39)
```

The result of the above implementation is that the GUI Fixed-length form for format BK tag 008 has an expand button next to the fields: "Publication Place" (position 15-17) and "Illustration Code" (positions 18-21). When the expand button is clicked, the relevant list of values is displayed.

Note: After updating the cataloging forms in ./xxx01/pc_tab/catalog, run util M/7 utility to update the PC forms package.

6 Default Subfields

You can define the default subfields for fields by editing the `marc_exp.dat` table located in the library's `pc_tab/catalog` directory. The subfields defined in this table are displayed in the following circumstances:

When a field is selected from the list of valid fields - available by using the F5 shortcut key or by selecting the New field (choose from list) option from the Edit menu.

When the Open form option from the Edit menu is chosen for a field for which no form is available.

Note that you do not have to define all subfields, just the most common ones. This is because the cataloger can manually add to the catalog record subfields that are not included in the list of defaults.

Following is an example from the table:

```
1  2  3      4
!!!-!!-!!-!!!!!!!!!!
011    XX a
017    XX ab
024 #  MU adz
025    XX a
027    XX a
028 ## MU ab
```

```
032    SE ab
033 ## XX abc
```

Key to Table:

Column 1 - Field tag
Enter a 3-character field tag.

Column 2 - Indicators
Enter specific indicators, or use the "#" character as a wildcard to indicate any indicator.

Column 3 - Record Format
Enter a specific record format, or use *XX* as a wildcard to indicate that the field is appropriate for any format. Refer to Record Formats on page 9 for more information on record formats.

Column 4 - Subfields
Enter the subfields that you want to be displayed as defaults. Enter them one after the other, without spaces between them. For example: acd.

Remember that you do not have to define ALL subfields, just the most common ones. This is because the cataloger can manually add subfields to the catalog record that are not included in the list of defaults.

7 Default Fields for New Record

You can define the fields that will appear automatically in a new record when the cataloger chooses New Record from the Cataloging menu.

Default fields for new records are defined in the `tagonnew.dat` table located in the library's `pc_tab/catalog` directory.

Following is an example from the table:

```
!!!!!!
L008
L007
```

Enter each desired field on a separate line, in the order in which you want the fields to appear.

If the field has a form, then the form will open automatically when the cataloger creates the new record. If no form is available, the tag without indicators or subfields is displayed ready for the cataloger to edit.

Note that the LDR field is inserted automatically when a new record is created, there is no need to define it in the `tagonnew.dat` table.

8 Tag Information

The Tag Information tab (lower pane of the Cataloging tab) provides a guide to the use of valid indicators, subfield codes and values for the field selected in the Catalog Editor (upper pane). The help displayed in the Tag Information tab is defined in a separate HTML file for each field. These help files are located in the library's /pc_tab/catalog/html directory.

The following are the naming conventions for these files:

nnn_xx_lng.html is in use for undefined indicators (for example, 100_xx_eng.html)

nnny_xx_lng.html or **nnnyy_xx_lng.html** are in use for specific indicators (for example, 853x_xx_eng.html)

y can be used to define a specific first indicator.

yy can be used to define specific indicators.

xx is the code for the record format, for example, BK for book. Refer to Record Formats on page 9 for more information on record formats. The notation *xx* is used to specify that the help is accurate for any record type.

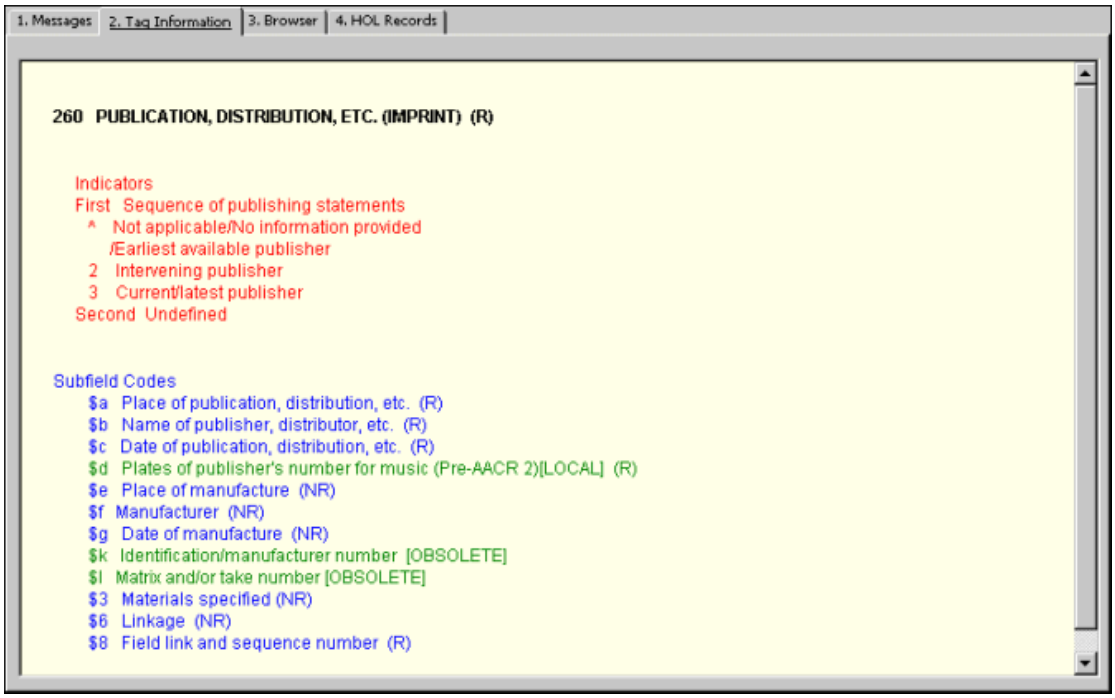
lng is the code for the language (for example, *eng* for English).

The following is the help file for the 260 MARC 21 field (260_xx_eng.html):

```
<html>
<head>
<include>style
</head>
<body>
<pre>
<div class=ct1>
    260   PUBLICATION, DISTRIBUTION, ETC. (IMPRINT)  (R)</div>
<div class=ct2>
         Indicators
         First   Sequence of publishing statements
            ^    Not applicable/No information provided
                 /Earliest available publisher
            2    Intervening publisher
            3    Current/latest publisher
         Second  Undefined
</div>
<div class=ct3>
    Subfield Codes
             $a   Place of publication, distribution, etc.  (R)
             $b   Name of publisher, distributor, etc.  (R)
             $c   Date of publication, distribution, etc.  (R)
             <span class=ct4>$d   Plates of publisher's number for
music (Pre-AAC
R 2)[LOCAL]  (R)</span>
             $e   Place of manufacture  (NR)
             $f   Manufacturer  (NR)
             $g   Date of manufacture  (NR)
             <span class=ct4>$k   Identification/manufacturer number
[OBSOLETE]
             $l   Matrix and/or take number [OBSOLETE]</span>
             $3   Materials specified (NR)
             $6   Linkage  (NR)
```

```
            $8   Field link and sequence number   (R)
</div>
</pre>
</body>
</html>
```

This is displayed as follows in the Cataloging module:



The different display elements - such as background color, text color and fonts - are defined in a file named *style* located in the same directory. Following is a sample of a section from the style file:

```
.ct1
{
  background-color:#FFFFE7;
  font-size:12;
  font-weight:bold;
  font-family:Arial, Helvetica, serif;
  color:#000000;
  text-decoration:none;
}
```

The above section controls the display of the title from the 260 example:

```
<div class=ct1>
    260   PUBLICATION, DISTRIBUTION, ETC. (IMPRINT)   (R)</div>
```

260 PUBLICATION, DISTRIBUTION, ETC. (IMPRINT) (R)

9 Search Headings

You can define the headings file that should be used when the cataloger chooses one of the Search Headings functions by editing the `scancode.dat` table located in the library's pc_tab/catalog directory. The following is a sample of the table:

```
!  1   2                    3                    4
5
!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!->
LOCAL                       USM01                USM10
USM12
050##                       LCC
086##                       SUD
100##  d                    AUT                  PER
110##                       AUT                  COR
111##                       AUT                  MET
130##                       TIT                  TIT
245##                       TIT
260##a                      PLA
260##b                      PUBKey to Table:
```

Key to table

Column 1 - Local

The Local column lists the field tags, indicators, and subfields for which the cataloger can search a headings list. The # character can be used as a placeholder for indicators in positions 4 and 5.

For example:
245## includes *2451, 2452, 24515*, and so on.

When using specific indicators make sure that specific lines are listed before general lines, since the first match found is always taken. For example, 24510 must be listed before 245##. If the table is defined in the following order:

```
245##
24510
```

then the 24510 field in the record will match the first line (245##), and not the second line.

If the lines are not sorted, and therefore *245##* is listed first, the search looks for matches to *245##*.

You can define subfields for the cataloger to search by placing the subfield code next to the field tag and indicator (see MARC 21 field *260* in the above sample table).

In column 12, you can set the subfields that should be ignored when the search heading string is built (such as control subfields: 6805w). The definition set in column

12 will work for the Search Field Headings tool (F3 / Ctrl + F3), not for Search subfield headings.

Column 2 - Optional

You can specify up to 20 subfields that are not to be overwritten. When a heading is selected from a list of headings displayed by any of the Search Headings options, the selected heading replaces the entire field (therefore, all subfields) in the record unless specified in this column.

Columns 3-11
At the top of each column is the code of a searchable base. Below that, for each field tag, is the code of the headings list that is displayed for the base. The code must be a valid headings index code.

For the Search Field Headings of other Base and the Search Subfield Headings of other Base options, it is possible to specify a base more than once in the scancode.dat table to enable the user to define more than one scan code for the field or subfield option. For example, if the `scancode.dat` table is defined as follows:

```
!  1    2                       3                       4
5
                6
!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!->
LOCAL                          USM01                  USM10
USM01
           USM01
100##  d                       AUT                    PER
AUT
           SUB
```

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

- Scan USM10 with PER
- Scan USM01 with AUT
- Scan USM01 with SUB

Note that the first base specified in the table is the local library/base and it is not displayed when selecting to scan the headings from other bases. For this reason, to enable the user to select this option when using this function, the column must be repeated.

In order to be able to search the base in the first line (Local), it has to be defined in the searbase file which resides in alephcom\tab.

Note that the maximum number of lines that the table can contain is 200.

In addition, note that the variable USE-ACC-TEXT in the tab100 table of the Authority library is used to define whether the preferred heading from the authority record is taken when selecting a heading that is a "See From" reference from the list of authority headings displayed when the "Search field/subfield headings of other base" options are used. If the variable is left blank or is set to N, then when selecting a "See From" heading from the "Search field/subfield headings of other base" (Ctrl + F3/F4) options, the preferred form of the heading is inserted automatically into the

catalog draft. If the flag is set to Y, the selected heading (even if it is the non-preferred form of the heading) is inserted into the cataloging draft.

When the "Search field/subfield headings of current base" options are used, if the heading is not connected to an authority record, then the heading is taken as is and inserted into the cataloging draft.

If the order of the subfields in the cataloging draft is different than the order of the heading subfield, the subfield order in the cataloging draft remains and the additional subfields are appended at the end of the cataloging draft record.

Note that this situation can result in a new index entry where the subfield content is the same as a pre-existing index entry, but the subfields are ordered differently. In such a situation, you may need to update the cataloging draft.

If the heading is connected to an authority record, the system checks the USE-ACC-TEXT variable:

If the flag is set to N (or is left blank), the system automatically inserts the preferred form of the heading from the associated authority record.

If the flag is set to Y and the heading is a "See From" heading pointing to another heading (to the preferred form of the heading) in the bibliographic list of headings, then the system takes the heading to which the "See From" is pointing (from the bibliographic database and not from the authority record).

If the flag is set to Y and the heading does not point to another heading in the database (in other words, the heading is connected to an authority record but it is the preferred heading), then the system takes the heading as is from the bibliographic database and not from the authority record.

When you run one of the Search Headings functions, the text taken from the cataloging draft undergoes non-filing procedures to strip non-filing text before the search is performed. The stripping is performed according to the non-filing indicator specified in the tab01.lng table of the library's tab directory.

Col.12 Sub-Field Code

Up to 20 sub-field codes that should be ignored when building the string for the Search Field Headings' tool (F3 / Ctrl + F3). Not relevant for search Subfield Headings (F4 / Ctrl + F4).

10 Search Subfield Options

The `tag_text.dat` table located in the library's `pc_tab/catalog` directory is used to define the menu options that are displayed when the cataloger chooses Search Subfield Options. Following is a sample of the table:

```
!1  2  3 4      5
!!!-!!-!-!-!!!!!!!!!!!!!!!!!!!->
655 ## L a Biographies
655 ## L a Bird's eye views
655 ## L a Cartoons
655 ## L a Catechisms
655 ## L a Daybooks
655 ## L a Diaries
655 ## L a Directories
```

```
655 ## L a Essays
655 ## L a Hymns
655 ## L a Journals
655 ## L a Memoranda
655 ## L a Questionnaires
655 ## L a Reviews
655 ## L a Syllabi
655 ## L a Time sheets

LKR ## L a UP//Up link
LKR ## L a DN//Down link
LKR ## L a ANA//Analytical link
LKR ## L a PAR//Parallel link
LKR ## L a ITM//Item link
```

Key to Table:

Column 1 - Field tag
Enter a 3-character field tag.

Column 2 - Indicators
You can enter a specific indicator, or use the # character as a wildcard to indicate any indicator.

Column 3 - ALPHA
ALPHA code. Must always be L.

Column 4 - Subfield
Enter the subfield for which you are providing a menu option. You can use the # character as a wildcard to indicate any subfield.

Column 5 - Text for menu option
Enter the text as you want it to appear in the Choose Subfield Text window displayed when the Search Subfield Options function is used. You can enter up to 45 characters. You can use two slashes // to separate the actual value from additional text that appears in the menu but will not be entered in the catalog record. See the LKR field in the above section, where "UP" is the value that is entered in the catalog record, and "Up link" is additional text that is displayed in the menu.

In addition to defining menu options, you can have the system check the validity of text entered into subfields when the user chooses the Check Record function, accessible from the Edit menu. To do so, follow these steps:

Edit the check_doc_tag_text table located in the library's tab directory. This table lists valid text for each field that you want to be checked.

The structure of the table is similar to the table used to define the text options; the following is a sample of the table:

```
Partial example of the table:


!  1   2                    3                    4
5                    6                    7                    8
9                    10                   11                   12

!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-
```

```
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
!  1   2                     3                     4
5                    6                    7                    8
9                    10                   11                   12
!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!


LOCAL                        USM01                USM10
USM12                USM01                USM11


050##                        LCC

086##                        SUD

100##  d                     AUT                  PER
AUT

110##                        AUT                  COR
6805w

700##  v                     AUT                  PER
6805w
```

Note that this table contains the valid values to be checked and should not include the description that can be added to the `tag_text.dat` table.

The `check_doc_tag_text` table can be used independently from the `tag_text.dat` table to check text validity for subfields without enabling the Search Subfield Options.

Ensure that the "`check_doc_tag_text`" program is listed in the `check_doc` table of the library's tab directory. The `check_doc` table lists all available checking programs.

11 Check Field

You can define the checks that are made when the cataloger chooses the Check Field function. To define them, edit the `check_doc_line` table located in the library's `tab` directory. There are two sections in this table:

- AL
- D

11.1 AL Section

This section enables you to define the following checks:

- Valid indicators and/or subfield codes for the tag.
- Presence of mandatory subfields.
- Non-repeatability of non-repeatable subfields.

Following is a sample of the section:

```
!!-!!-!!!!-!!!!!-!-!-!
AL XX      260   -
AL XX      260   a 0 -
AL XX      260   b 0 -
AL XX      260   c 0 -
AL XX      260   d 0 -
AL XX      260   e 0 1
AL XX      260   f 0 1
AL XX      260   g 0 1
AL XX      260   6 0 1
```

Key to the AL Section

Column 1 - Section ID
Enter *AL* for each line in this section of the table.

Column 2 - Record Format
Enter a specific record format, or use *XX* as a wildcard to indicate that the field is appropriate for any format. Refer to Record Formats on page 9 for more information on record formats.

Column 3
Not in use.

Column 4 - Field tag
Enter a field tag.

Column 5 - Subfield/Indicators
Enter the subfield that you want to be included in the check.
To define valid indicators for the tag, enter a hyphen (-) in this column.

Note that the indicator portion (for all formats) must be listed before the subfield portion, for each field.

Column 6 - Mandatory - Non-mandatory subfield / Valid first indicator
If column 5 contains a subfield code, this column is used to define whether the subfield is a mandatory subfield of the field. Values are *0* and *1*. If the subfield is mandatory, enter *1*. If the subfield is optional, enter *0*.

If column 5 contains a hyphen, this column is used to define possible values for the first indicator of the field.

Column 7 - Repeatable subfield / Valid second indicator
If column 5 contains a subfield code, this column is used to define the repeatability of the subfield. Values are *1 - 9* and hyphen (-). If the subfield is not repeatable, enter *1*. If the subfield can be repeated unlimitedly, enter hyphen (-). You can use values *2 - 9* to determine that the subfield can be repeated up to the number of times represented by the selected value.

If column 5 contains a hyphen, this column is used to define possible values for the second indicator of the field.

Example:

```
AL XX      020   -
AL XX      020   a 0 1
AL XX      020   c 0 1
AL XX      020   z 0 -
AL XX      020   6 0 1

AL XX      022   -
AL XX      022   - 0
AL XX      022   - 1
AL XX      022   a 0 1
AL XX      022   y 0 -
AL XX      022   z 0 9
AL XX      022   6 0 1
```

In the above example:

For tag 020:

- Both first indicator and second indicator are undefined (blank)
- All subfields $a, $c, $z, and $6 are optional
- Subfield $a, $c and $6 are not repeatable
- Subfield $z is repeatable an unlimited number of times

For tag 022:

- First indicator can be blank, 0 or 1
- Second indicator is blank
- All subfields $a, $y, $z, and $6 are optional
- Subfields $a and $6 are not repeatable
- Subfield $y is repeatable an unlimited number of times
- Subfield $z can be repeated up to 9 times

11.2 D section

This section enables you to determine the rules for checking dependencies among subfields of a single field. Following is a sample of the section:

```
1  2  3    4     5 6              7 8       9        1011     12
13
!!-!!-!!!!-!!!---!-!!!!!!!!!!!!!!-!-!-!!!!!!!!!!!!!!-!-!-
!!!!!!!!!!!!!!-!-

D  XX 9036 260   a                Y b               Y c
N
D  XX 9036 300   a                Y c               N
D  XX 9036 300   a                Y                   b
N
```

Key to D section of table:

Column 1 - Section ID
Enter *D* for each line in this section of the table.

Column 2 - Record Format
Enter a specific record format, or use *XX* as a wildcard to indicate that the field is appropriate for any format. Refer to Record Formats on page 9 for more information on record formats.

Column 3 - ID # of error message
Choose the ID # of the error message that is appropriate for the check. The message is displayed to the cataloger when the system performs a check and finds an error.

User-defined messages can be defined in the library's `check_doc.lng` table located in the library's `tab` directory.

Column 4 - Field tag
Enter a field tag.

Column 5 - Subfield
Enter the subfield that you want to be included in the check. See also column 8 below.

Column 6 - Subfield contents
Optional. Enter any contents that you want the system to check for.

Column 7 - Type of dependency
Enter *Y* if the subfield should be present. Enter *N* if the subfield must not be present.

Columns 8, 9, 10 and 11, 12, 13
The subfield, contents, and dependency columns are repeated. This enables you to create *if, then* statements. For example, you can say that if subfield $a appears (defined in columns 5, 6, 7), then subfield $b must appear (defined in columns 8, 9, 10), and subfield $c must not appear (defined in columns 11, 12, 13).

12 Fix Record

Fix routines are standard library-defined procedures that automatically "fix" or make changes to cataloging records. The system librarian is in charge of defining which fix programs are available and when they are run. Two tables are involved in the setup of fix procedures:

`tab_fix` (located in the library's `tab` directory)

`fix_doc` (located in the library's `pc_tab/catalog` directory)

12.1 tab_fix

The `tab_fix` table defines three aspects:

The fix program that defines the type of change that is performed on the cataloging record

The fix routine in which the fix program runs

If required, additional parameters for the fix program

Following is a sample of the `tab_fix` table:

```
1                 2                                   3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS   fix_doc_tag_008
INS2  fix_doc_001
MERGE fix_doc_merge                 OVERLAY-01
AUT   fix_doc_new_aut_2
```

Key to the `tab_fix` Table:

Column 1 - Routine name

Fix routines are "logical names" for defining a group of fix programs. 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.

The following are ALEPH's reserved routines:

AUT

Run when the new derived authority record is created using the "derived new record" function in GUI Cataloging module.

BNA

Fix programs linked to the BNA routine are automatically activated when the Load BNA Records (file-98) batch service is used.

DEL

Fix programs linked to the DEL routine are activated automatically when a record is deleted via the GUI or during a batch process or by any other program.

HOL

Fix programs linked to the HOL routine are automatically activated when HOL records are created in the Items or Serials functions > Items list > Tab 6. (HOL Links) > "Create New" button. Note that this routine must be used in the holdings library (xxx60).

ILL-L

Fix programs linked to the ILL-L routine are automatically activated when the Locate function is activated from the ILL module (Locate button in the BIB Info tab of the ILL request).

INS

Fix programs linked to the INS routine are automatically activated when a record is sent to the server.

INS2

Fix programs linked to the INS2 routine are automatically activated when a record is sent to the server. The difference between INS and INS2 is that this routine is executed just before the record is updated in the database, and therefore it can make use of the document's system number even if it is a new document. Note that INS2 programs are run after check_doc procedures, therefore the outcome of INS2 programs is not checked before update.

INSFS
Fix programs linked to the INSFS routine are automatically activated when fast cataloging from the administrative modules (Circulation and Acquisitions). This routine is also performed when bibliographic records are created using the Special Request option in the Web OPAC and when bibliographic records are created in the Course Reading module.

LDMRG

Run automatically on a bibliographic record from MARCIVE when it is merged with a matching record in the database.

LOCAT
Fix programs linked to the LOCAT routine are automatically activated when the Locate Record function is used.

M-36
Fix programs linked to the M-36 routine are automatically performed on the records in the input file for the Check Input File Against Database (manage-36) service.

MERGE
Fix programs linked to the MERGE routine are automatically activated when the Paste Record function is used in the Cataloging module.

MNG50
Fix programs linked to the MNG50 routine are automatically activated when the Create Holdings and Item Records Using Bibliographic Data (manage-50) service is used. The programs are performed after the creation of the holdings and ADM records and can be used to modify them.

P-31
Fix programs linked to the P-31 routine are automatically activated when the Load Authority Records batch process (manage-31) is used. Currently, the fix_doc_preferred and fix_doc_aut_mesh programs should be defined in the tab_fix table of the authority library under the P-31 routine.

REF
Fix programs linked to the REF routine are automatically activated when the Trigger Z07 Records (manage-103) service and the Load MAB Authority Records (manage-20) are used.

UE_01
Fix programs linked to the UE-01 routine are automatically activated when the update doc daemon is activated.

The system librarian can add user-defined routine names to the lower section of the table.

Column 2 - Program name
This is the name of the program that will perform a particular fix. Each routine can have up to 20 program names assigned to it, so that a number of different fixes can be performed together. In order to assign more than one program to a routine, open a separate line for each program and repeat the routine name in column 1. For example:

FIX2 fix_doc_tag_008

FIX2 fix_doc_tag_100

FIX2 fix_doc_tag_250

In this example, whenever FIX2 is selected, three programs are run.

Column 3 - Parameters
Certain fix programs require additional information, such as table names. This column is used to define additional parameters for fix programs. Note that the documentation for each fix program indicates whether or not parameters are needed.

Following are the available fix programs:

fix_ced_uid
Creates UID fields from the 020 field or the 022 field for loading purposes (for example, previous versions of the p_manage_18 service).

fix_doc_001
Inserts a 001 field with the system number of the record into the cataloging draft (for example, $$1000010091). This fix program must be attached to INS2 and not to INS, since it needs the system number of the document.

Column 3 of the tab_fix table of the library's tab directory must be used to define whether or not the program should overwrite existing 001 fields. Following are the available options:

OVERWRITE (always replaces existing 001 fields)

NO-OVERWRITE (does not replace existing 001 fields)

OVERWRITE-NON-NUMERIC (replaces only 001 field where there is at least one non-numeric character)

If no parameter is defined in the parameters column, then the default value is OVERWRITE.

Note that when the update_z103_uni linking program (UNIMARC links - Italian version) is used in the tab_z103 table of the library's tab directory, this program must be used.

fix_doc_001_prefix_sysno
This fix program automatically creates a 001 field containing a prefix defined in column 3 of the library's tab_fix table and in the system number of the record with leading zeros (for example, $$USM01-000003526 - in this example, USM01 has been defined in column 3 of the tab_fix table as the prefix). If a 001 field already exists, this program overrides the field and adds the new 001 field with the defined prefix.

Note that this fix program must be attached to INS2 and not INS, as it needs the system number of the document.

fix_doc_001_sysno
Automatically creates a 001 field with the library name and the system number of the record (for example, $$aUSM01000000000000000111142). This fix program must be attached to INS2 and not INS, for it needs the system number of the document.

fix_doc_001_sysno_inv
This routine is used to keep the original system number of records that are uploaded to the system (p_manage_18) after being exported into, for example, MARC format. When the records are downloaded into MARC format, the original system number of the records is deleted. The fix_doc_001_sysno_inv takes the system number stored in the 001 field (previously inserted by the fix_doc_sysno routine) and uses it to correct the values of the ALEPH sequential file used for uploading the records.

fix_doc_004_lkr
This routine adds the MARC 21 004 field to holdings records. The 004 field contains the system control number of the bibliographic record for which the holdings record was created.

fix_doc_005
This routine inserts a 005 field with the current date and time into the cataloging draft record. Note that it is recommended to run this program under the INS2 fix routine. This ensures that the field is created just before updating and not after reading and overriding the errors/warnings displayed in the Record Check Warnings window.
FIX_DOC_099

This routine creates:

- a 997 field with one subfield $a for each 099 field which has two occurrences of subfield $a, and
- a 998 field with one subfield $a for each 098 field which has two occurrences of subfield $a.

This can be seen in the example below:

```
099   L $$aaa $$abb   ->  997   L $$aaa bb
098   L $$aaa $$abb   ->  998   L $$aaa bb
```

fix_doc_1xx_240

The fix_doc_1xx_240 routine enables the cataloger to choose an authority heading composed from subfields $$a and $$t and to create the required 1XX/240 bibliographic combination.

If the selected 1XX field has subfield $$t:

- A 240 field is opened.
- The data from the subfield $$t of the 1XX field is copied from subfield $$t up to the end of the field (in other words, it includes all subsequent subfields).

- The copied data is pasted into the new 240 field with first indicator 1, changing subfield $$t to subfield $$a. All other subfields remain as they are.
- All subfields from subfield $$t up to the end of the field are removed from the 1XX field.

The second indicator of the created 240 field is left blank. Note that this can be automatically corrected to the correct value by using the fix_doc_non_filing_ind fix program which can be set, for example, to be run upon update of the record or as part of the fix routine to which the fix_doc_1XX_240 program is attached.

Note in addition that if the record already contains a 240 field, a new field will be created and the original field will be retained in order to enable the librarian to select the preferred field.

fix_doc_1xx_243

The fix_doc_1xx_243 routine enables the cataloger to choose an authority heading composed from subfields $$a and $$t and to create the required 1XX/243 bibliographic combination.

If the selected 1XX field has subfield $$t:

- A 243 field is opened.
- The data from the subfield $$t of the 1XX field is copied from subfield $$t up to the end of the field (in other words, it includes all subsequent subfields).
- The copied data is pasted into the new 243 field with first indicator 1, changing subfield $$t to subfield $$a. All other subfields remain as they are.
- All subfields from subfield $$t up to the end of the field are removed from the 1XX field.

The second indicator of the created 243 field is left blank. Note that this can be automatically corrected to the correct value by using the fix_doc_non_filing_ind fix program which can be set, for example, to be run upon update of the record or as part of the fix routine to which the fix_doc_1XX_243 program is attached.

Note in addition that if the record already contains a 243 field, a new field will be created and the original field will be retained in order to enable the librarian to select the preferred field.

fix_doc_880

This routine replaces the tag number of the alternate graphic representation field (880) by the associated tag registered in subfield $6 of the field. In addition, the tag number of the associated field is removed from subfield $6 but the occurrence number is retained. For non-880 fields, the tag number of the associated field (for example, 880) is removed from subfield $6 and the occurrence number is retained.

This program creates parallel fields, both containing the same tag number. For example:

```
1001  L $$6880-01$$a[Name in Chinese script].
```

```
8801  L $$6100-01/(B$$aShen, Wei-pin.
```

Is changed to:

```
1001  L $$601$$a[Name in Chinese script].
1001  L $$601$$aShen, Wei-pin.
```

Note that for this fix to work, subfield $6 must be the first subfield in both linked fields.

fix_doc_add_order_info

This routine adds to bibliographic records information from Order records (Z68) and Vendor records (Z70) attached to them. It is mostly intended for use in the UE_01 section of `tab_fix`.

For each Z68 record attached to the BIB record, fields 541 and 590 are added (or updated) as follows:

```
541##$$a - Z70-VENDOR-NAME
541##$$d - Z68-OPEN-DATE
541##$$e - Z68-ORDER-NUMBER

590##$$a - Z68-OPEN-DATE
590##$$b - Z68-APPROVER-ID
590##$$e - Z68-ORDER-NUMBER
```

Example:

```
541  L$$aJerry Books$$d20030903$$e123-1
590  L$$a20030903$$bID511260$$e123-1
```

If the bibliographic record does not contain fields 541 or 590 with Z68-ORDER-NUMBER of the processed Order record, these two fields are added. If it does, the fields are updated if necessary: 541##$$a is updated with Z70-VENDOR-NAME and 590##$$b is updated with Z68-APPROVER-ID.

Note that a bibliographic record, to which more than one Order record is attached, will have multiple occurrences of fields 541 and 590.

fix_doc_add_pinyin_check_sub9

The fix_doc_add_pinyin_check_sub9 routine runs on fields defined in tab_pinyin if their content is CJK. The program takes the content of $$a and creates a parallel $$9 in pinyin, using chi_segmentation (Z113) and pinyin translation (Z114). The 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.

fix_doc_add_pinyin_insert_sub9

The fix_doc_add_pinyin_insert_sub9 routine runs on fields defined in tab_pinyin if their content is CJK. This routine takes the content of $$a and creates a parallel $$9 in pinyin, using chi_segmentation (Z113) and pinyin translation (Z114). In cases where a character has more than one pinyin option, the created subfield contains <option1>.

fix_doc_008_han_1
The fix_doc_008_han_1 routine handles BIB record year information based on Chinese and Korean letters. It performs the following actions:

1. Calculates the Christian year based on the special name of the era and populates 260$$c/264$$c (publication date). If 264$$c is applied, it is done based on the following priority order: 2nd indicator 1,0.3,7.
2. Simultaneously updates 008/7-10 (Date 1) and 008/11-14 (Date2) according to the special name of the era in 260$$c. If 260$$c is missing, 264$$c is applied (based on the following priority order: 2nd indicator 1,0.3,7).

Once fix_doc_008_han_1 is called by the system, the above actions occur and the fields 260$$c/264$$c, 008/7-10 (date 1) and 007/11-14 (date 2) are updated.

fix_doc_arabic
The fix_doc_arabic program can be used to translate Arabic characters into the correct form (presentation forms) according to its position in the word. This fix program uses the arabic_form table in the $alephe_unicode directory for the purpose of deciding which characters to translate and the various forms. The four possible shapes of an Arabic character are:

Isolated:
The character is not linked to either the preceding or the following character.

Initial:
The character is linked to the following character but not to the preceding one.

Middle:
The character is linked to both the preceding and the following character.

Final:
The character is linked to the preceding character but not to the following one.

Following is a sample of the `arabic_form` table:

```
! 1   2    3    4    5                      6
!!!!-!!!!-!!!!-!!!!-!!!!---!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
FE80                0621   ARABIC LETTER HAMZA
FE81           FE82 0622   ARABIC LETTER ALEF WITH MADDA ABOVE
FE87           FE88 0625   ARABIC LETTER ALEF WITH HAMZA BELOW
FE89 FE8B FE8C FE8A 0626   ARABIC LETTER YEH WITH HAMZA ABOVE
FEF1 FEF3 FEF4 FEF2 064A   ARABIC LETTER YEH
```

fix_doc_assign_issn

The `fix_doc_`assign_issn routine can be used by ISSN national centers. The routine creates and assigns ISSNs (022 $a) and Centre codes (022$2) to bibliographic records. The ISSN is composed of eight characters. The first seven characters can be the numbers 0 to 9 and the last character, which is a check character, can be the numbers 0 to 9 or an uppercase X. The first seven characters are assigned by the International ISSN Centre and the last character is calculated by the routine.

The following is an example of ./usm01/tab/tab_fix:

```
! 1                 2                                   3
```

```
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>

ISSN    fix_doc_assign_issn                $$2=9 limit=1885000
```

The fix_doc_assign_issn routine uses two parameters in column 3. The first parameter ($$2) contains the Centre code, which indicates the ISSN National Centre responsible for assigning the ISSN. The second parameter (limit) contains the upper limit of the ISSN counter.

For example, if `limit=1885000`, then the highest possible ISSN number is 1884-9997. The first 7 characters are the numbers of the counter and the last character is the check number.

Note
If the parameters are not defined, the fix_doc_assign_issn routine does not work.

Activate this mechanism by creating the create-issn counter using util G/2. The starting point of the counter is the first possible ISSN, as determined by the International ISSN Centre.

Note that if the record already contains an ISSN, the routine does not run.

fix_doc_aut_lc
This program adds subfield $$2 [LC] to fields 1XX, 4XX, 5XX and COR of authority records from the LC authority database.

fix_doc_aut_mesh
This program adds subfield $$2[MeSH] to fields 1XX, 4XX, 5XX and COR of authority records from the MeSH authority database.

fix_doc_aut_duplicate
This fix routine, which is applied in the Authority Library, deals with ambiguous headings that arise from having the same 4XX in more than one record. It does not take established headings (such as topical heading, personal name, uniform title, and so on) into account.

fix_doc_aut_duplicate performs the following:

- When a 4XX field is found to be duplicate to the 4XX or 1xx of another record, the 1xx field is appended to the 4XX field, using $$7; that is, the 4XX field is changed as follows:

```
4XX [original subfields and text] $$7 [text of 1xx (subfields are
stripped].
```

The fix_doc_aut_duplicate routine is based on checking the 4XX field against the ACC (Z01 headings) lists named GEN and DUP. Accordingly, these lists must be present in the authority library.s tab00 and tab01. GEN is required for authority

control, and should already be present in the library.s tables. The 4#### field is sent to the DUP list, with the 7 subfield stripped. The configuration should be as follows:

```
./xxx10/tab/tab11_acc
4####       7 *          DUP   -7

./xxx10/tab/tab00.<lng>
H DUP   ACC     11 00     0000       Duplicates

./xxx01/tab/edit_field.<lng>
1 # AUT## H   -69
2                         7 A ^[         ]
2                         # A ^
```

If the non-preferred heading (4XX) exists in the DUP list, this means that the field already exists in other records in the Authority library. In this case, the 4XX field will be modified to include the 1XX content in the $$7 subfield .

If the 4XX does not exist in the DUP list, but does exist in the GEN list (meaning there is one previously existing record with the same field content), the 4XX fields of both the record being checked and of the record that is already in the database will be modified to include the 1XX content in subfield $$7.

If the 4XX does not exist in DUP or in GEN, it is a new heading, and there is no special treatment.

Your library can choose to use another subfield number instead of $$7. In order to re-configure this, change column 3 of `tab_fix` of the authorities library. `Tab11_acc` and `edit_field.<lng>` must be updated accordingly. If column 3 of `tab_fix` is empty, the default subfield is $$7.

The INS2 routine allows you to preview a record after the fix, before sending it to the server.

Note that it updates the existing records in the database.

The INS routine updates the record only after sending it to the server, but does not allow this preview, so that no existing records are updated before being sent to the server.

Accordingly, use the INS routine for the actual fix, and the INS2 routine for previewing fixes.

This is relevant only for the fix_doc_aut_duplicate routine, but not for other INS2 functions.

fix_doc_bnb
Adds to the cataloging record a 015 $$a field with a unique BNB (British National Bibliography) number and prefix and subfield $$2bnb in the same tag (for example, 015 $$aABC1001$$2bnb). The values (last BNB number and prefix) are taken from the *last-bnb-number* sequence in UTIL G/2. If the 015 field already exists, its content is not overridden.

The first place of the sequence is alphanumeric. This way it can represent higher numbers. It is numeric up to running number 99,999 and then it is changed to alphabetic (uppercase, like A, B, C and so on).

‘A’ will represent 100,000 and ‘B’ will represent 110,000 and so on.

In any one year the number assigned should run from 00001 to 99999 and then from A0000 to Z9999.

This fix program must be assigned to INS in the tab_fix table.

fix_doc_char_conv_z
This routine carries out character conversion using a user-defined conversion script. Column 3 can contain a conversion instance from `$alephe_unicode/tab_character_ conversion`. If Column 3 is empty, the default instance is Z. In the following example from `tab_character_conversion`, the Z instance uses the `line_sb2line_sb` procedure :

```
Z   ##### # line_sb2line_sb       conversion_z
```

Here is a section from the tab_fix table:

```
CHAR   fix_doc_char_conv_z
```

fix_doc_create_035
This program moves the MARC 21 001 field (Control number) and the MARC 21 003 field (Control number identifier) to the MARC 21 035 field (System control number) deleting the original fields. The new 035 field is added in the following format:

```
035## $a(003)001
```

Additionally, this program also automatically adds to the record a new 001 field. The value for the new field is taken from the sequence "last-001-number" in UTIL G/2.

fix_doc_create_035_1
The fix_doc_create_035_1 program is similar to fix_doc_create_035. The difference is that this program does not create a 001 field (based on the last-001-number Z52 sequence) after creating the new 035 field. In short, this program can be used when you only want to create the 035 (from 001 and 003) without adding a new 001 field automatically.

fix_doc_create_066
The fix_doc_create_066 program creates the 066 field that is used to indicate the character sets present in the record. The field is created with $c for each alternate character set. The fix_doc_create_066 program can only be used on records in the MARC-8 character set. This program is used mainly for export purposes.

fix_doc_create_fmt
This program, according to the definitions of the LDR field, adds the FMT field with the record format (for example, BK for books). The program can be used to add the FMT field to records imported through the Z39.50 server that do not have the FMT field. If the field is present, the program does nothing. This program is for use with records imported through Z39.50. Define the program in the tab_fix of the EXT

library and in the $alephe_gate/<base name>.conf file. For more information, see the *Universal Gateway* document.

fix_doc_create_hol_local_notes
This program is based on the information stored in local tags cataloged in the bibliographic record. It creates holdings records and moves the defined local tags from the bibliographic record to the appropriate holdings record. These tags can then be indexed and displayed as if they were part of the bibliographic record. The application for storing local tags in a holdings record is in a consortial environment where a single bibliographic record is shared by multiple institutions and an institution would like to include local tags that not everyone sees. Which local tags are moved to the holding record from the bibliographic record and which are displayed in the OPAC can be configured by the local institution.

In the event that a single institution has more than one holdings record, a "super holdings" record can be created which stores local tags but not call numbers or any location information.

The following are the tables involved:

The tab_fix table in the tab directory of the bibliographic library (XXX01):

The parameters column, column 3, of the tab_fix table must contain the section in the `tab_fix.conf` table that specifies the local tags, the indicators, and so on. The following is a sample of the setup needed in the tab_fix table to use the program:

```
  ! 1                2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
HOLD  fix_doc_create_hol_local_notes LOCAL-NOTES
```

The tab_fix.conf table in the tab directory of the bibliographic library (XXX01):

This table is used to specify the local tags, the indicators, the subfield that contains the owner of the record, the merge section used in the `tab_merge_overlay` table of the holdings library tab directory and the relevant section from the `tab_mapping` table of the tab directory of the bibliographic library. The following is a sample of the setup needed in the tab_fix table to use the program:

```
[LOCAL-NOTES]
local notes = 590##,690##
owners subfield = 9
owners list = AA,BB,LIN
merge section = 98
mapping section = LCN-2-HOL
```

The `tab_merge_overlay` table in the `tab` directory of the holdings library (XXX60):

This table is used to define how the holdings record is updated (merged) if it already exists. Following is a sample of the setup needed in the `tab_merge_overlay` table to use the program:

```
98 1 Y #####
98 2 Y 590##
98 2 Y 690##
```

The `tab_mapping` table in the tab directory of the bibliographic library (XXX01):

This table is used to define how to map the original tags from the bibliographic record

into the new tags in the holdings record. The following is a sample of the setup needed in the `tab_mapping` table to use the program:

```
LCN-2-HOL       541## abcde 541   abcde                    Y Y
LCN-2-HOL       541## fho39 541   fho39                    Y Y
LCN-2-HOL       561## ab39  561   ab39                     Y Y
LCN-2-HOL       590## ab9   590   ab9                      Y Y
LCN-2-HOL       690## ab9   690   ab9                      Y Y
```

Note that the original bibliographic record has the local tags taken off, but these changes do not take effect until you save the record to the server.

fix_doc_create_hol_sid

Creates SID field in the Holding record with the HOL system number in $$c and the related BIB in $$b.

fix_doc_delete_empty

This program is used to delete fields or/and subfields that do not have any content. Note that in the Cataloging module, when a record is saved on the server, empty fields/subfields are deleted automatically. This fix program can be used, for example, when cataloging records are loaded by other services (for example, cataloging loaders) that do not perform this deletion automatically.

fix_doc_delete_chi_spaces

This program is used to delete spaces between two Chinese letters.

fix_doc_field_200

This program can be used by UNIMARC libraries to create additional fields based on the 200 field (title and statement of responsibility).

If there is more than one $a subfield in field 200, the system creates 423 fields, one for each occurrence of subfield $a, except for the first one. In the 423 field, the first indicator will be blank and its second indicator will be zero. The $a subfield of the 423 field will contain the embedded content of the 200 field's $a subfield. The $c subfield of the 423 field will contain the embedded content of the 200 field's $c subfield that follows $a in the 200 field. The 423 field will also include a $1 subfield containing a 200## tag.

Here is an example:

```
423_0 $a (The second $a in field 200)
      $c (Optional, contains the second $c of field 200)
      $1 (200##)
```

The program also creates a new 510 field $a for each occurrence of $d in the 200 field. The first indicator of field 510 is 1 and the second one is blank.

Here is an example:

```
5101_ $a ($d in field 200)
```

This fix program does not delete previously created 423 and 510 fields.

fix_doc_field_410
This program can be used by UNIMARC libraries to create an additional field based on field 410#1 (Series).

If a UNIMARC 410#1 field contains $a and $1, the program will create a 2252# field. $a of the 2252# field will contain the embedded content of the 410 $a field.

Here is an example:

```
2252# $a<$a in field 410#1>
```

The fix program does not delete previously created 2252# fields.

fix_doc_field_lower / fix_doc_field_upper
The **fix_doc_field_upper** and **fix_doc_field_lower** programs change the case of all occurrences of a field given as a parameter to the routine. The fix_doc_field_upper program changes the case to uppercase and the fix_doc_field_lower program changes the case to lowercase.

The parameter has the following format: *<field><tag><subfield>*

Here is an example from the tab_fix table of a UNIMARC library:

```
! 1                2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
LOWER fix_doc_field_lower            701#0a
```

As shown in this example, the # sign must be used if no character is defined for the field or for the indicator.

In order to change every subfield in a field, you do not have to add a subfield parameter. If a subfield parameter is present, the case of all its occurrences will change accordingly.

fix_doc_do_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.

Note that the maximum number of lines that the file can contain is 500 lines.

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 41 Setting Up a Script for the Correction of Records in Aleph Sequential Format chapter on page 116 in this document. The following is a sample 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,^
3 949##                    CHANGE-SUBFIELD                 h c
3 949##                    CHANGE-SUBFIELD                 b h
```

fix_doc_fixed_fields
This program replaces the hyphens ("-") in fixed-length fields (such as MARC 21 LDR, 006, 007 and 008 fields) by a caret ("^"). This program can be used to correct MARC 21 records in which blanks in fixed-length fields are marked by hyphens (hyphen is a valid value in MARC 21 and should not be used to mark spaces).

fix_doc_hld_stmt
This program performs the same procedures as the expand_doc_hld_stmt program. The difference between the two programs is that when running the expand_doc_hld_stmt program the 863/4/5 enumeration and chronology data are created virtually, whereas when running fix_doc_hld_stmt, the 863/4/5 enumeration and chronology data are added to the HOL record.

For the fix_doc_hld_stmt program, the tab_fix table of the HOL library (XXX60) should include the following line:

```
! 1                  2                          3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!>
HSTMT fix_doc_hld_stmt
```

To enable the generation of summary holdings statements for item records that are not linked to a subscription record or do not have a value in the Copy ID field in the subscription record, the parameter:

```
GET_Z30_BY=HOL
```

should be defined in col.3 of tab_fix as follows:

```
! 1                  2                                  3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
```

```
HSTMT fix_doc_hld_stmt              GET_Z30_BY=HOL
```

For item records which are linked to a subscription record and which hold a value in the Copy ID field in the subscription record and the item record, the parameter:

```
GET_Z30_BY=COPY_ID
```

can be defined in col.3 of tab_fix as follows:

```
!   1                   2                           3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
HSTMT      fix_doc                      GET_Z30_BY=COPY_ID
```

Note: If no parameter is defined in col.3 of tab_fix the system will use the default parameter:

```
GET_Z30_BY=COPY ID
```

For full details of this program refer to expand_doc_hld_stmt in the Expand Routines section of the Indexing chapter.

fix_doc_hol_852_1

This program inserts a space between the main class and the first cutter of the call number, and was required in former versions for filing consistency. The fix procedure is no longer required, since the "lc_call_no" filing procedure creates the same filing key, whether or not the space is present. The space is inserted if the first cutter is held in subfield $h of the MARC 21 852 field and if there is not already a space.

For example, this program changes:

```
LOC0  L $$bMAINX$$cLIB$$hE183.8.B7$$iW45
1993$$oBOOK$$4MAIN$$5LIB$$3Book
```

to:

```
LOC0  L $$bMAINX$$cLIB$$hE183.8 .B7$$iW45
1993$$oBOOK$$4MAIN$$5LIB$$3Book
```

fix_doc_ldr_05_d

fix_doc_ldr_05_d sets the value of position 05 of the LDR field of the record to *d*. It can be listed under the DEL section.

fix_doc_ldr_sta_delete

The fix_doc_ldr_sta_delete program changes the record status in position 05 of the LDR field to 'd' if the STA $$aDELETED field is present in the record.

fix_doc_lkr_up

This program adds subfield $n (up link note) and subfield $m (down link note) to the LKR field of types UP (up link), ANA (analytic) and PAR (parallel). Subfield $m is built based on the 245 field of the record with the LKR field. Subfield $n is built based on the 245 field of the record to which the LKR points. If the records are for a serial (FMT = SE), then subfield $n and subfield $m are built from the corresponding 222. If the 222 field is not present, then the subfields are built from field 245.

It is possible to exclude punctuation suffixes by using the DELSUF parameter in column 3 of tab_fix. This parameter specifies the punctuation marks to be excluded.

The ./xxx01/tab/tab_fix setup example below specifies which puncuation suffixes (slash, space + period, and space+ colon) to exclude from subfields $m or $n of the LKR field when text is copied by fix_doc_lkr_up from 245$a or 245$$b:

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS   fix_doc_lkr_up                 DELSUF=|/|^.|^:|
```

Notes:

- Within the parameter (key=value) no space is allowed, because spaces are often used as delimiters for different parameters.
- If a space should be part of the suffix, use a caret (^).
- The first sign of the value is the delimiter of different suffixes. Use a character that is not used as any suffix of those listed. In the example above, the delimiter is the pipe sign (|).

fix_doc_7xx_lkr

The fix_doc_7xx_lkr fix routine supports the 7XX field update based on information from the LKR field.

For example: ./usm01/tab/tab_fix

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
7XX   fix_doc_7xx_lkr
```

For example: . /usm01/pc_tab/catalog/fix_doc.eng

```
! 1   2 3           4
!!!!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
7XX   N L Create 7XX based on LKR field
```

fix_doc_7xx_lkr creates the 7XX field based on the LKR field.

The following is a description of the fix activation flow, including the cataloger's manual actions and the fix's automatic actions:

The cataloger performs the following:

1. The cataloger catalogs LKR $$b (the system number of the linked record) and $$r (the 7XX tag that contains the information of the linked record). If necessary, the

cataloger also catalogs subfield $$z (to indicate the type of relationship) and $$g (to indicate the enumeration and chronology of the issue in which the absorption or split occurred).

2. The cataloger activates the new fix either manually by selecting the relevant fix routine or automatically when saving the record.

The fix performs the following:

1. The fix checks the FMT of the record.

    - If FMT is not SE or 902$$b = 631, 632, 633, or 634– no activity is performed.
    - Otherwise, the fix continues with the next step.

2. The fix first adds subfield $$a with the value PAR and subfield $$l with the value <BIB library> to the LKR field. (<BIB library> represents the BIB library code – for example – USM01.)

3. If LKR$$aPAR has been added previously, the fix enriches the LKR fields by creating the $$x, $$n, and $$m subfields.
    - Subfield $$m is created from the concatenation of subfields $$a, $$n, $$p, $$h, and $$c of the 245 field of the record being edited. If a subfield $$m already exists in the LKR field, it is not replaced.
    - Subfield $$n is created from the concatenation of subfields $$a, $$n, $$p, $$h, and $$c of the 245 field of the linked record. If a subfield $$n already exists in the LKR field, it is not replaced.
    - Subfield $$x is created from the value of the first 022$$a of the linked record. If a $$x subfield already exists in the LKR field, it is not replaced. If subfield $$x includes only a hyphen (-) (indicating that it should be treated like an empty subfield), the value is not copied from the linked record.

4. The fix deletes the following 7XX fields that have a subfield $$w: 765, 767, 775, 776, 777, 780, 785, 786, and 787. A 7XX field without a $$w subfield is left unchanged.

5. If the LKR contains subfield r, the fix automatically creates one 7XX field for each LKR of type PAR.
    - The tag of the 7XX field is the content of LKR $$r. (The field name is created from the first three digits of the value of LKR $$r).
        - The first digit of the indicator is created from the fourth digit of the value of LKR $$r.
        - The second digit of the indicator is created from the fifth digit of the value of LKR $$r. If the fifth digit of the value of LKR $$r is blank, the second digit of the indicator is also blank.

- $$w contains the 001 number of the linked record (not the system number)
- $$t is created by concatenating the $$a, $$n, $$p, $$h, and $$c of the 245 field of the linked record.
- $$x is copied from LKR $$x
- $$i is copied from LKR $$z
- $$g is copied from LKR $$g

fix_doc_lng_from_bib
The fix_doc_lng_from_bib program can be used to update the language code in the 008 field of the holdings record and/or ADM record according to the language code in positions 35-37 of the 008 field of the associated bibliographic record. In the ADM record, positions 35-37 of the 008 field are updated with the language code. In the holdings record, positions 22-24 are updated with the language code.

Note that this fix is not required for the online update of the language code in the 008 field of the holdings records. Holdings records created in the system are updated according to the HOL-008-LNG variable of the `tab100` table of the holdings library (for example, USM60). This variable is used to determine the default language code for the MARC 21 008 field in holdings records:

If the variable is set to 0, then the language code of the 008 field of the holdings record is set to the defaults specified in the `tab_tag_text` table.

If the variable is set to 1, the language code of the 008 field of the holdings record is taken from the bibliographic record based on standard system rules (that is 008, 041, and so on).

fix_doc_marc21_spaces
This program converts contiguous runs of two or more blanks (spaces) into the blank character specified by the DOC-BLANK-CHAR variable in the library's tab100 table. The following MARC 21 fields are affected by the fix: 010##, 260##, 310##, 321##, 362##, 515##, 525##, 533##, 76###, 77###, and 78###.

fix_doc_merge
This program merges or overlays cataloging records according to the merging program defined in the tab_merge table located in the library's tab directory. Column 3 of the tab_fix table is used to define the merging routine that matches the relevant section in the tab_merge table. Refer to Merging Records on page 84 for more information.

fix_doc_ndl_nb_br
This fix automatically creates the Japanese National number and the BR number in bibliographic records. The fix is based on the information stored in local field 900$a cataloged in the bibliographic record. There are three types of numbers that are differentiated by their prefix. Each type has its own counter. Util G/2 defines the ndl-nb-2, ndl-nb-0, and ndl-br-1 counters. Column 3 of tab_fix is used to define the upper limit of the Japanese National number and the BR number:

The following is an example of ./ndl01/tab/tab_fix

```
! 1                  2                                   3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
NBBR   fix_doc_ndl_nb_br                  nb2=9999999 nb0=9999999 br1=9999999
```

- nb2 – indicates the upper limit of Japanese National Bibliography number. with a prefix of 2
- nb0 – indicates the upper limit of Japanese National Bibliography number. with a prefix of 0
- br1 – indicates the upper limit of BR number with a prefix of 1.

Note that this fix must be manually activated.

fix_doc_new

This fix routine provides a generic platform for deriving a new record, based on a previously cataloged record. A companion configuration table, `tab_fix_new`, is used to define which tags should be generated for the new derived record. It is possible to define:

- tags based on tags of the source record; a single new tag can be composed of several tags from the source record
- tags with content, including fixed length fields
- empty (template) tags - and similar tags
- a program in column 7 of `tab_fix_new`
- manipulate tags

`tab_fix_new` can have several sections; a specific line in `tab_fix` is linked to the relevant `tab_section` by defining "TYPE=" in col. 3. For example, if the relevant section of `tab_fix_new` is called "son", then col. 3 of tab_fix will have TYPE=son.

Refer to the table's header and Configuration Guide documentation for more information on the `tab_fix_new` table.

fix_doc_new_ana

This program creates an analytic record from the current record. The new record is created as follows:

LDR: Position 07 is set to b. Other default values are defined according to the specifications of the `tab_tag_text` table of the library's tab directory.

008: Default values are defined according to the specifications of the `tab_tag_text` table of the library's tab directory.

050: If the field exists in the parent record, it is taken to the new record.

080: If the field exists in the parent record it is taken to the new record.

245: The field is opened but remains empty.

260: The field is taken from the parent record.

300: Subfield c is taken from the parent record.

LKR: Points to the parent record.

fix_doc_new_aut_2
This program is used by MARC 21 libraries to create an authority record from the current bibliographic record. The new record is created as follows:

If the 100 field is present in the bibliographic record, then the authority record will be derived with a 100 field.

If the 110 field is present in the bibliographic record, then the authority record will be derived with a 110 field.

If the 111 field is present in the bibliographic record, then the authority record will be derived with a 111 field.

If the 130 field is present in the bibliographic record, then the authority record will be derived with a 130 field.

The 670 field of the authority record is created from the 245 (subfield $a) and 260 (subfield $c) fields of the bibliographic record. This program also adds text to the new 670 field. The text can be configured via a message file in $aleph_root/error_lng called: fix_doc_new_aut_2.

The record created by the program is by default set to the authority library defined under the AUT section of the `library_relations` table. If this section is not present, you can define the authority library by using the parameters column of 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 (this is done by setting the last two digits in the active library as 10).

fix_doc_new_aut_3
This program is used by UNIMARC libraries to create an authority record from the current bibliographic record. The new record is now created as follows:

If the 70# field is present in the bibliographic record, then the authority record is derived with a 200 field.

If the 71# field is present in the bibliographic record, then the authority record is derived with a 210 field.

If the 72# field is present in the bibliographic record, then the authority record is derived with a 220 field.

If the 73# field is present in the bibliographic record, then the authority record is derived with a 200 field.

The 810 field of the authority record is created from the 200 field - subfield $a and from 210 - subfield $c and $d. This program also adds text to the new 810 field. The text can be configured via a message file in `$aleph_root/error_lng` called fix_doc_new_aut_3.

The record created by the program is by default set to the authority library defined under the AUT section of the `library_relations` table. If this section is not present,

you can define the authority library by using the parameters column of 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 (this is done by setting the last two digits in the active library as 10).

fix_doc_new_aut_4
This program is used by UNIMARC libraries to create an authority record from the current bibliographic record. This program is cursor-sensitive and according to the position of the cursor, the new authority record is created either from the relevant 6XX, 5XX, or 7XX fields (see complete list below). The new authority record is created as follows:

The original field (for example, 700) is taken as is and placed in the corresponding 2XX field in the authority record.

A new 810 field is built based on the 200 / 230 / 250 fields (subfield $a) and from the 210 field (subfields $c and $d) of the bibliographic record. The new program also adds text to the new 810 field. The text can be configured via a message file in $aleph_root/error_lng called fix_doc_new_aut_4.

For example, if the bibliographic record contains the following fields:

```
200 l#$Steam locomotives of Germany and Austria
210 ##$a[Cambridge, Mass.]$cHarvard Univ. P.$dl981
```

Then the new 810 field is added as follows in the derived authority record:

```
810 ##$aSteam locomotives of Germany and Austria, Harvard Univ., 1981
```

In addition, the fields 152 (with subfields $a and $b) and 801 (with subfields $a, $b and $c) are added to the record without contents.

The record created by the program is by default set to the authority library defined under the AUT section of the `library_relations` table. If this section is not present, you can define the authority library by using the parameters column of 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 (this is done by setting the last two digits in the active library as 10).

Note that if the cursor is not placed on one of the relevant tags (6XX, 5XX, or 7XX fields), then the authority record derived from the bibliographic record is created with the default LDR and 008 fields and without the 2XX, 152, 801 and 810 fields.

The following is the list of relevant fields for the creation of the authority record:

500, 600, 601, 606, 700, 701, 702, 710, 711, 712, 720, 721, 722 and 730.

fix_doc_new_aut_5
This program is used by MARC 21 libraries to create an authority record from the current bibliographic record. This program is cursor-sensitive and according to the position of the cursor, the new authority record is created either from the relevant 1XX, 4XX, 6XX or 7XX fields (see complete list below). The new authority record is created as follows:

The original field (for example, 600) is taken as is and placed in the corresponding 1XX field in the authority record.

A new 670 field is built based on the 245 (subfield $a) and 260 (subfield $c) fields of the bibliographic record. The new program also adds text to the new 670 field. The text can be configured via a message file in $aleph_root/error_lng called fix_doc_new_aut_5.

The record created by the program is by default set to the authority library defined under the AUT section of the `library_relations` table. If this section is not present, you can define the authority library by using the parameters column of 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 (this is done by setting the last two digits in the active library as 10).

Note that if the cursor is not placed on one of the relevant tags (1XX, 4XX, 6XX or 7XX fields), then the authority record derived from the bibliographic record is created with the default LDR and 008 fields and without the 1XX and 670 fields.

List of relevant field for the creation of the authority record:

100, 110, 111, 130, 440, 490, 600, 610, 611, 630, 650, 651, 700, 710, 711, 730 and 740.

fix_doc_new_aut_6
This program is used by MARC 21 libraries to create an authority record from the current bibliographic record. This program is cursor-sensitive, and is equivalent to the fix_doc_new_aut_5 program, except for the following changes:

- No additional $aleph_root/error_lng/fix_doc_new_aut_5 text is inserted in the 670 field.
- The title and date in the $$a subfield of the derived 670 field are separated by a comma and a space.
- The $$a subfield of the derived 670 ends with a colon. If there is a period at the end of the derived 670 field, it is removed.
- The record may be derived also from the 8XX fields. Deriving from the 800, 810, 811, and 830 fields is the same as from the 700, 710, 711, and 730 fields respectively.

fix_doc_new_aut_7
This fix_doc program is used by MARC 21 libraries to create an authority record from the current bibliographic record (using the "derive new record" function). The new authority record is created based on 1xx, 6xx, and 7xx (same as fix_doc_new_aut_5 routine). fix_doc_new_aut_7 works also for subject fields cataloged in 69x fields (Hebrew subjects). With this new program, the authority record created will contain the text from the 69x field in the 159 field.

fix_doc_non_filing_ind
Automatically inserts the relevant non-filing indicators according to the stopwords

defined in the tab02 table of the library's tab directory. Non-filing indicators for each tag are defined in the library's tab01.lng table, column 6.

fix_doc_non_filing_ind2

This program is similar to fix_doc_non_filing_ind but can also process stopwords in tab02 that contain an apostrophe or a space (for example: "'n" or "na h-").

When defining a non-filing indicator in tab02 that contain an apostrophe or a space, use fix_doc_non_filing_ind2 instead of fix_doc_non_filing_ind.

fix_doc_notes

Replace text with an alternative text. Enables the automatic translation of bibliographic note fields (for example, the: translation from English to French). This fix works in conjunction with the tab_fix_notes Aleph table (list of translations per bibliographic tag and subfield). For more information, refer to **Automatic Translations – Functionality and Examples** on page **182**.

fix_doc_oclc

The fix_doc_oclc program is used for the OCLC record loader. The program moves the OCLC 001 and 003 fields to the ALEPH (MARC 21) 035 field, in the following format: (003)001.

fix_doc_oclc_2

This program is similar to fix_doc_oclc, except that it deletes any existing 035 field in the incoming record before writing the 001/OCLC number to a new 035 field. The fix also adds the UPD field (Y or N) to authority records.

fix_doc_oclc_retain_001

This program is similar to fix_doc_oclc and fix_doc_oclc_2, except that this procedure adds the new 035 field without deleting the OCLC 001 field or the 003 field.

fix_doc_overlay

The fix_doc_overlay program allows the user to merge records when they are uploaded to the system. It is used to upload those records that find a unique match in the database when the Check Input File Against Database (manage-36) service is run. The records in the output file produced by this service are given new system numbers that match the system numbers of the corresponding records found in the database. The fix_doc_overlay program can be used to merge the records in the database with the new incoming records using a merge routine from the `tab_merge_overlay` table. Column 3 of the `tab_fix` table is used to define the merging routine that matches the relevant section in the `tab_merge` table. When running the Load Catalog Records (p-manage-18) function, select the fix routine defined for this program.

fix_doc_own_1

The fix_doc_own_1 program inserts the value of the Cat. OWN ID field of the cataloger to the OWN field of new created record.

fix_doc_preferred
Automatically creates a COR field when the preferred term of the authority record is changed. The COR field contains the original term. This is necessary so that the link with bibliographic records (that have the original preferred term) is retained.

fix_doc_punctuation_usm
Adds punctuation to MARC 21 245, 260, 264, and 300 fields. The program only deals with subfields a, b and c of the fields.

If the last subfield ends in ".", "!" or "?", a full stop "." is not added to it. The program can specify punctuation marks in column 3. Here is an example:

```
! 1                 2                                   3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS   fix_doc_punctuation_usm        .!?-
```

If the last subfield of one of the these fields - 245, 260, 264, and 300 fields. - ends with one of the punctuation marks specified in column 3 example, a full stop will not be added to it.

If column 3 is empty, in order to add punctuation marks to the three default ones, the default marks (.!?) must be specified in column 3 together with other marks. For instance, a hyphen can be added as shown in the above example.

fix_doc_qualified_ucs

This fix_doc routine accepts a list of fields as parameters. The routine checks the existence of the fields and updates or creates the QUA field, as follows:

- If the fields exist, then the single subfield $$a of the field is set to Y.
- If the fields exist and the cataloger is identified as UCS staff (the Cataloger's OWN ID is 'OLCC'), the single subfield $$a of the field is set to A
- If at least one of the fields does not exist, the single subfield $$a of the field is set to N

For example, consider the following tab_fix setup:

INS2 fix_doc_qualified_ucs 245##,1001#

In the above example, the QUA field is updated with "Y" only if both fields 245## and 1001# exist. If the updating cataloger is UCS staff, the status is automatically set to "A".

If either field is absent, the QUA field is created or updated with N.

The maximum number of allowed fields in this table is 16.

fix_doc_redo_880
This routine reverses the effects of the fix_doc_880 program. This program restores the tag number of the alternate graphic representation field (880). For example:

```
1001  L $$601$$a[Name in Chinese script].
1001  L $$601$$aShen, Wei-pin.
```

Is changed to:

```
1001  L $$6880-01$$a[Name in Chinese script].
8801  L $$6100-01/(B$$aShen, Wei-pin.
```

Note that the order of the paired fields is important, because the tag of the first of a pair is left as is, and the second of a pair is transferred to 880.

In addition, note that the input for this program must be in MARC8 (not in UTF) encoding. The reason for this is that this fix routine sets the escape sequence and orientation for the language code, and in order to do so, the record must be in MARC8 encoding. The fix_doc_redo_880 program will work correctly on UTF records, but will not set the escape sequence and orientation for the language code.

fix_doc_ref_1

This program is used to update a non-preferred term in the bibliographic record to a preferred term. The correction occurs only if the UPD field in the authority record is set to "Y". The tab_fix table of the bibliographic library must include the fix_doc_ref_1 program under all relevant sections:

```
INS   fix_doc_ref_1
UE_01 fix_doc_ref_1
REF   fix_doc_ref_1
```

The fix_doc_ref_1 program must be included under INS for the update of records from the Cataloging module; under UE_01 for the indexing daemon (UE_01 process); and, if necessary, under REF for the Trigger Z07 Records (manage-103) service.

Column 3 of the tab_fix table may be used to set a Y\N\A parameter.

For example:./usm01/tab/tab_fix

```
! 1                2                              3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
UE_01 fix_doc_ref_1                 Y,A
```

‘Y’ and ‘N’ parameters set the library’s policy in case the bibliographic heading record and the authority heading record are not of the same language script, i.e. one is Latin and one is not. If the parameter is set to Y then the link to the authority record will be created, but the update of the bibliographic record will not take place. The parameter’s default is N.

In addition, if the bibliographic heading contained the non-preferred form of the heading and if the conditions for bibliographic updated exist (UPD field is 'Y' and fix_doc_ref_1 has been defined in the `tab_fix` table), then the bibliographic record is updated as in the following example:

Authority record:

```
150  $$aFighting dogs
450  $$aPit dogs
```

Bibliographic record:

```
65010  $$aPit dogs $$zItaly
```

The bibliographic record is updated as follows:

```
65010  $$aFighting dogs $$zItaly
```

If column 3 of tab_fix is set with 'A'; the flip of the bibliographic heading takes place only if there is a match to the authority preferred term. The update of the bibliographic heading (1XX field) based on authority non-preferred term (4XX field for example) does not occur at all. The link to the authority record is created in any case.

Manual update of bibliographic heading base on the authority 4XX heading is enabled, that is, if a user manually searches for a heading to a bibliographic record (via GUI-Cataloging, CTR+ F3), he/she is able to copy the 4XX non-preferred term to the bibliographic 1XX heading.

If tab_fix is set with both parameters 'Y' and 'A', there is also a check for a match in the language script (the update of the BIB 1XX occurs only of it is in the same language script as the AUT 1XX).

Column 3 of the tab_fix table may be used to set a Y\N parameter that sets the library's policy in case the bibliographic heading record and the authority heading record are not of the same script, that is, one is Latin and one is not. If the parameter is set to Y then the link to the authority record is created, but the update of the bibliographic record dose not take place. The parameter's default is N.

To deactivate the automatic updating of subfield $$6 in the BIB record from the references of the Authority record, set column 3 of the tab_fix table with the parameter J.

fix_doc_rlin_1

This program moves the MARC 21 001 field (Control number) and the MARC 21 003 field (Control number identifier) to the MARC 21 035 field (System control number), deleting the original fields. The new 035 field is added in the following format:

```
035## $a(003)001
```

fix_doc_shelf_mark

Intended for use with HOL records. It appends a counter to the $$j subfield of the 852 field. The counter is based on a prefix that is cataloged in the $$j subfield and the 'bl' prefix. For example, consider that the following subfield is cataloged in the 852 field:

$$j mss

If the HOL library has a counter of ‘bl-mss’ with the value 12, the fix_doc_shelf_mark routine will fix the $$j field to $$j mss.13

fix_doc_sort
Sorts the fields of the current record according to the order defined in the ALEPH table of codes (tab01.lng). However, within the MARC 21 5xx, 6xx and 7xx groups of fields, the order of the fields remains as they were entered by the cataloger. Note that for the sorting of the 5xx, 6xx and 7xx blocks, the 500, 600, 700 and 800 codes must be explicitly listed in the table even if they are not used. Fields that do not have any content are deleted. In an authority library, all fields are sorted according to the order defined in the ALEPH table of codes (`tab01.lng`).

fix_doc_sort_505

This program is similar to fix_doc_sort, except that it sorts the 505 fields as a group after all of the other 5XX fields.

By using this program, the 505 fields are inserted as one block, not interspersed with the existing 5XX fields block. The order of the 505 fields remains as it was when the fields were entered by the cataloger.

fix_doc_sort_lkr
This program sorts the LKR fields of the record in the following order: DN, PAR, UP. Note that for this program to run, it is also necessary to define the fix_doc_sort program that is used to sort the fields of the record.

For the fix_doc_sort_lkr program, the tab_fix table should include the following lines:

```
! 1                 2                                   3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS   fix_doc_sort
INS   fix_doc_sort_lkr
```

* Note that INS is used here as an example. The fix can be attached to any routine name (reserved or user-defined).

fix_doc_sort_sub6
This program should be used after performing the fix_doc_880 program that creates parallel fields that are different script representations of each other. The fix_doc_sort_sub6 program is used to sort the linked-parallel fields. The fields are sorted by the occurrence number stored in subfield $6.

fix_doc_space_char
This program changes a character that has been used as a placeholder for blanks to a blank, in the fixed fields LDR, 001-008 in MARC21 and LDR, 001, 005, 100 in UNIMARC. This can be used, for example, in the Download Machine Readable Records (print_03) service when exporting records in MARC format.

You define the character that acts as a placeholder for blanks in tab100 using the DOC-BLANK-CHAR variable. Only the character defined will be replaced by a blank. For example, in USM01, the caret character is defined as the "BLANK-CHAR".

You can include or exclude specific fields, by using column 3 of tab_fix option (replacing the tab100 variable DOC-BLANK-CHAR with a space).

The parameters in column 3 of tab_fix are comma-separated fields (five characters each, which can include hashes, for example, 10###) to include or exclude (to exclude, prefix the list of fields with a SINGLE dash (-), that is -010##,100##,245##). Note that the total length of the fields list must not exceed the length of column 3 of `tab_fix`, specified in the table's header.

The following is an example of fix_doc_space_char in tab_fix with inclusion (in this example, ONLY the specified fields will be processed):

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
EXPRT fix_doc_space_char             010##,008##,001##
```

The following is an example of fix_doc_space_char in tab_fix with exclusion (in this example, ALL record fields will be processed EXCEPT from the specified fields):

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
EXPRT fix_doc_space_char             -010##,008##,001##
```

If column 3 of tab_fix for fix_doc_space_char is left BLANK, the routine will behave as follows:

If the tab100 variable MARC-TYPE = "1" (USMARC), the following fields are processed: LDR and 001 - 008.

If MARC-TYPE = "2" (UNIMARC), the following fields will be processed: LDR, 001, 005, 100##

Otherwise (when MARC-TYPE is other than "1" or "2"), only LDR will be processed.

fix_doc_sub

This program adds subfield $$2[MeSH] to fields 6XX of bibliographic records. The subfield is only added when the second indicator of the field is 2 (Medical Subject Headings).

fix_doc_suppress

This fix routine checks the bibliographic record according to the check_doc section that is defined in column 3 of the tab_fix line. If the check succeeds, a STA field is added to the record with the value SUPPRESSED. If the check fails, the STA field with the SUPPRESSED value is removed. This routine may be used in the UE_01

section of tab_fix if automatic suppression of records is wanted when a BIB record has no attached ADM information. The following are examples of ADM Information: Items, Orders Subscriptions, Links to other BIB, Links to HOL records, and other deletion checks.

fix_doc_tab04_(01-99)
Translates the field codes of the record into another set of field codes. The translation values are defined in the library's tab04 table. The suffix defines which set of codes is chosen from the table (for example, fix_doc_tab04_01 refers to set 01 in tab04).

fix_doc_tag_008
Automatically adds the date of publication to positions 07-10 (or corrects the existing values) of MARC 21 008 field according to the date entered in MARC 21 260 field, subfield $c. If 260$$c is missing, 264$$c is applied (based on the following priority order: 2nd indicator 1,0.3,7).

fix_doc_tag_100_open_date
Automatically adds the current date (creation date) to positions 00-07 of UNIMARC 100 field. The date is entered in the pattern YYYYMMDD.

fix_doc_tag_008_open_date
Automatically adds the current date (creation date) to positions 00-05 of MARC 21 008 field.

fix_doc_tag_008_update_date

This fix routine enables the adding of the update date to positions 26-31 of an HOL record’s 008 field.

fix_doc_tag_041
This fix program updates positions 35-37 in field 008 with the language that is set in subfield $$a of field 041.

fix_doc_transliteration

If activated in a BIB library, this program creates an additional occurrence of the field for which it was called, entering transliterated contents of the source field into this new parallel field. The program works with routines TRNL1, TRNL2, TRNL3, TRNL4, TRNL5 and TRNL6. Each routine works with program arguments pointing to a transliteration table or program. This routine is currently available for the CJK contents only.

```
TRNL1 fix_doc_transliteration        HANJA_TO_HANGUL
TRNL2 fix_doc_transliteration        HANJA_TO_PINYIN
TRNL3 fix_doc_transliteration        PINYIN_TO_HANGUL_MOE
TRNL4 fix_doc_transliteration        PINYIN_TO_HANGUL_CK
TRNL5 fix_doc_transliteration        KANA_TO_ROMANIZED_KANA
TRNL6 fix_doc_transliteration        KANA_TO_HANGUL
```

- HANJA_TO_HANGUL - transliterates Hanja characters into Hangul
- HANJA_TO_PINYIN - transliterates Hanja characters into Pinyin

  If multiple possible transliterations are detected, the fix routine puts then in subsequent double square brackets, with the possible transliterations of each term separated by commas within the square brackets.

  For example

  700$$a[[zhong,guo]] [[bao,xian]] [[jian,du,guan]] li [[wei,yuan,hui]]
- FIX_HANJA_TO_PINYIN – fixes the transliteration that is created by HANJA_TO_PINYIN

  Removes the double square brackets that are created by the HANJA_TO_PINYIN option and removes all the transliteration options for each term except the first one.
- PINYIN_TO_HANGUL_MOE and PINYIN_TO_HANGUL_CK - both transliterate Pinyin input into Hangul, but the mappings are slightly different.
- KANA_TO_ROMANIZED_KANA - transliterates Japanese Kana entries into Latin-alphabet Kana
- KANA_TO_HANGUL - transliterates Japanese Kana into Hangul

If activated in the AUT library, this fix routine transliterates the contents of $$q of the 1XX entry and enters the resulting string into a new occurrence of field 400, subfield $$a. The settings in tab_fix in the AUT library are the same as in the BIB library.

fix_doc_trans_doc

This fix routine adds translated content of specific fields. It works with a configuration table that specifies:

- Which fields in the document are translated
- Which target field stores the translated content
- Which translation method is used

The configuration table is sent to the fix routine as a parameter.

For example, [BIB library]/tab/tab_fix:

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
TRNL1 fix_doc_trans_doc              tab_doc_trans
```

The configuration table must exist in the library where the fix is performed (the Bibliographic library or the AUT library) and have the following structure:

Column 1: **source field and indicators**
Size: 5
can be used for the fourth and fifth positions to indicate truncation of numeric additions to the field code (for example, 245## for 2451, 2452, 24501)

Column 2: **source subfield(s)**
Size: 10

Column 3: **target field and indicators**
Size: 5
can be used for the fourth and fifth positions to indicate truncation of numeric additions to the field code (e.g., 245## for 2451, 2452, 24501)

Column 4: **target subfield(s)**
Size: 10

Column 5: **translation method**
Size: 25
Options are: HANJA_TO_HANGUL, HANJA_TO_PINYIN, FIX_HANJA_TO_PINYIN
PINYIN_TO_HANGUL_MOE, PINYIN_TO_HANGUL_CK
KANA_TO_ROMANIZED_KANA, KANA_TO_HANGUL

Column 6 (optional): **translation method to perform on the translated data**
Size:25
Options are: HANJA_TO_HANGUL, HANJA_TO_PINYIN, FIX_HANJA_TO_PINYIN, PINYIN_TO_HANGUL_MOE, PINYIN_TO_HANGUL_CK, KANA_TO_ROMANIZED_KANA, and KANA_TO_HANGUL

Use column 6 if you want to perform another translation on the translated content. After translating the field using the translation method as specified in column 5, another translation is done.
The added field the data after both translations.
Note that if the translated content after using the first transliteration method (as specified in column 5) is identical to the source field (no actual translation is applied) – the second transliteration (as specified in column 6) is not performed, and no fields are added to the document.

For example:
1. In <Bibliographic library>/data_tab/tab_doc_trans:

```
!
!1    2          3     4          5                          6
!!!!! !!!!!!!!!-!!!!!-!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!
130## anp        940   anp        HANJA_TO_HANGUL
130## anp        940   anp        HANJA_TO_PINYIN
130## anp        940   anp        HANJA_TO_PINYIN            PINYIN_TO_HANGUL_MOE
130## anp        940   anp        HANJA_TO_PINYIN            PINYIN_TO_HANGUL_CK
!
240## anp        940   anp        HANJA_TO_HANGUL
240## anp        940   anp        HANJA_TO_PINYIN
240## anp        940   anp        HANJA_TO_PINYIN            PINYIN_TO_HANGUL_MOE
240## anp        940   anp        HANJA_TO_PINYIN            PINYIN_TO_HANGUL_CK
```

According to the above setup, four corresponding fields with the transliteration content are created for the 130## and 240## fields, subfields a, n, and p:

- Source data is translated using HANJA_TO_HANGUL
- Source data is translated using HANJA_TO_PINYIN
- Transliteration that is created by HANJA_TO_PINYINis translated using PINYIN_TO_HANGUL_MOE

- Transliteration that is created by HANJA_TO_PINYIN is translated using PINYIN_TO_HANGUL_CK

The translated content will be stored in field 940.

fix_doc_uk_222
This program adds a new 222 UKMARC field to SE (serial) format records. The 222 field is created from fields 245 and 240, subfields $a, $j and $s.

fix_doc_uk_marc21
This routine converts UKMARC to USMARC.

fix_doc_uni_100
Automatically adds date of publication to positions 09-12 (or corrects the existing values) of the UNIMARC 100 field according to the date entered in UNIMARC 210 field, subfield $d.

fix_doc_uni_100_advanced
The fix_doc_uni_100_advanced program is similar to the fix_doc_uni_100 program. This program automatically inserts or corrects the dates in the 09-12 and 13-16 positions of the UNIMARC 100 field according to the date entered in the UNIMARC 210 field, subfield $d. Additionally, the dates of the 210 field (subfield $d) are standardized. For example, for dates like 198? or 19?, the fix program replaces question marks and spaces with "-" (hyphens).

fix_doc_usm_001
Automatically creates a 001 field. The value is taken from the sequence "last-001-number" in UTIL G/2. If the 001 field already exists, a new 001 field is created based on the "last-001-number" and the old field is stored in a new 035 field. In addition, note that if the 003 field is also present, then the program deletes the field and adds its contents to subfield $b of the newly created 035 field.

fix_doc_usm_222
This program adds a new 222 MARC 21 field to SE (serial) format records. The 222 field is created from the 245 field, subfields $a and $b.

fix_doc_usnaf
Adds to the cataloging record 001$$a and 010 $$a fields with a USNAF control number and prefix, (for example, 001$$aABC1001 / 010 $$aABC1001). The values (last USNAF number and prefix) are taken from the *last-usnaf-number* sequence in UTIL G/2.

fix_doc_japanese

This routine is used to communicate with an external product used in the Japanese market (Happiness) to enrich a document with segmented and transliterated versions of the data. The URL address of the external product should be defined as a parameter in column 3 of tab_fix.

Note that fields returned from the external product longer than the maximum number of characters allowed are split into several separate smaller fields. This should be viewed and edited by the cataloger.

fix_doc_zero_ldr_00_04
This program is used to set to zeros the first five character positions of the LDR field (00-04). These positions contain a numeric string that specifies the length of the entire record. The number is right-justified and each unused position contains a zero. The LDR of the records in the system usually contains either spaces - when the record is created through the Cataloging module - or the original logical record length from the imported record. In both cases, these values are incorrect and misleading. This fix should be used during conversion, import and cataloging.

Note that ALEPH's export routines calculate the record's length automatically when the total record is assembled for exchange.

expand_doc_fix_abbreviation
This program can be used both as an expand program and as a fix program. Refer to the Expand Record section in the Indexing module for more details (expand_doc_fix_abbreviation).

expand_doc_type
This program can be used both as an expand program and as a fix program. Refer to the Expand Record in the Indexing module for more details (expand_doc_type).

fix_doc_create_7xx_kor and **fix_doc_create_7xx_marc**
The following two fix routines create full 76X-78X fields based on the existing slim 76X-78X fields, that is: 76X-78X fields with subfield $$w data.

- fix_doc_create_7xx_kor is used for KORMARC BIB record
- fix_doc_create_7xx_marc is used for MARC21 BIB record.

The cataloger should delete the LKR field before updating the $$w subfield in the 76X-78X and recreate the LKR manually.
The structure and content of the created full 76X-78X linking fields are as follows:

KORMARC (fix_doc_create_7xx_kor)
The order of the 76X-78X subfields is as follows:

- $$t (title)
- $$s (uniform title)
- $$a(author)
- $$b(edition)
- $$d(publisher)
- $$r(report number)
- $$u(STRN)
- $$x(ISSN)
- $$y(CODEN)
- $$z(ISBN)
- $$w(system number)

The source information for every subfield is taken from the target record in the BIB library.

7xx subfield	**Source information to be taken from the target record**
$$t (title)	245 $$a, $$f, $$g, $$k, $$n, $$p
$$s (uniform title)	130 $$a or 240$$a If there is no 130, use 240; if there is no 240, do not create $$s
$$a (author)	100 tag (all subfields) -> 110 tag (all subfields) -> 111 tag (all subfields) -> 700 tag (all subfields) -> 710 tag (all subfields) -> 711 tag (all subfields) The system looks first for 100, then for 110, and then 111, and so on, until it finds a matching field. The first field found is used.
$$b (edition)	250 $$a
$$d (publisher)	260 $$a, $$b, $$c
$$r (report number)	088 $$a
$$u (STRN)	027 $$a
$$x (ISSN)	022 $$a
$$y (CODEN)	030 $$a
$$z (ISBN)	020 $$a
$$w (BIB library code) + target record system number, entered by the cataloger	Remains as it was filled by the cataloger.

Note:

Data from input fields that have multiple subfields are created without the input subfields.

The following fields are mapped in the fix_doc routine:

- 760
- 762
- 765
- 767
- 770
- 772
- 773
- 774
- 775
- 776
- 777
- 780

- 785
- 786
- 787

with the following subfields:

- $$t(title)
- $$a(author)
- $$b(edition)
- $$d(publisher)
- $$x(ISSN)
- $$z(ISBN)
- $$w(system number)

MARC21 (fix_doc_create_7xx_marc)
The order of the 7xx subfields should be as follows:

- $$a (author)
- $$t (title)
- $$s (uniform title)
- $$b (edition)
- $$d (publisher)
- $$r (report number)
- $$u (STRN)
- $$x (ISSN)
- $$y (CODEN)
- $$z (ISBN)
- $$w (system number)

Source information for every subfield should be taken from the target record ($$w) in the BIB library (prefix of $$w)

7xx subfields	**Source information to be taken from the target record**
$$a (author)	100 tag (all subfields) -> 110 tag (all subfields) -> 111 tag (all subfields) -> 130 tag (all subfields) -> The system will first look for 100 and then for 110 and then 111, and so on, until it finds a matching field. We should derive it from the first field.
$$t (title)	245 $$a, $$f, $$g, $$k, $$n, $$p
$$s (uniform title)	240$$a the first tag
$$b (edition)	250 $$a
$$d (publisher)	260 $$a, $$b, $$c
$$r (report number)	088 $$a
$$u (STRN)	027 $$a

$$x (ISSN)	022 $$a
$$y (CODEN)	030 $$a
$$z (ISBN)	020 $$a
$$w (BIB library code) + target record system number, entered by the cataloger	Remains as it was filled by the cataloger.

The subfields allocation is same as in above KORMARC information.

fix_doc_create_lkr
Create LKR fields depending on the 76X-78X fields $$w data.

In order to create a functional LKR field, the slim 76X-78X field must have data in $$w that can be used to create LKR $$b and $$l.

The structure of the LKR field is:
$$a - link type (UP, PAR,DN) – see the following section 1.
$$b - system number of the target record – see the following section 3.
$$l - target BIB library code – see the following section 3
$$r- MARC Tag and Indicators – see the following section 4
$$m - content of 245 $$a of the same record
$$n - content of 245 $$a of the target record

LKR fields will be created only when the link type is PAR, UP or DN.

NOTE: subfields $$m and $$n will be generated by the system using fix_doc_lkr_up routine. The procedure should be set in the library's tab_fix table in the INS2 section.

1. Setting type of link (LKR $$a)
   The 76X-78X field tag determines the link type:
   760 – UP
   762 - DN
   765 - PAR
   767 – PAR
   770 - DN
   772 - UP
   773 – UP
   774 - DN
   775 - PAR
   776 - PAR
   777 - PAR
   780 - PAR
   785 - PAR
   786 - PAR
   787 – PAR

2. Setting the document system number($$b) and BIB library code($$l)
   The cataloger enters

- 78002 $$w(YUL02)48923

$$b should be taken from $$w of 76X-78X with deleting the prefix

- $$b48923

$$l should be taken from the prefix of $$w of 76X-78X with deleting the parentheses.

- $$lYUL02

If there is no prefix in the content of $$w of 76X-78X, the same library code as the source record could be set to $$l.

3. Setting the $$r
   3 digits of MARC Tag and 2 digits of Indicators from slim 76X-78X field created by the cataloger will be used.
   - $$r78002

fix_doc_tag_008_heb
This fix routine gets the years in Hebrew letters or in Arabic numbers, (entered in MARC 21 260 field, subfield $c. If 260$$c is missing, 264$$c is applied) converts them to Latin numbers, and places them in field 008, positions 7-10. If 264$$c is applied, it is done based on the following priority order: 2nd indicator 1,0.3,7.

fix_doc_signatura
This fix routine gives a unique inventory number to a specific catalogue entry. It figures a new inventory number for a given year and series and places it in 090 field of the document. In general, every year starts from 1. See the following example:

```
2009 a 12.
```

- 2009 – represents the year 2009
- a – the code for a series
- 12 – the assigned running number that is derived from the system counter util g/2 - last-signa-a.

You can define a separated counter per series code, for example, last-signa-a, last-signa-b, last-signa-c, etc.

fix_doc_aut_008_pos_29
Fix routine for the Authority record. Parameters must be defined in column 3 of tab_fix. The parameters list the MARC fields, separated by a comma, that are checked by the program. Up to 25 MARC fields can be listed.
Sample from ./usm10/tab/tab_fix

```
! 1             2                       3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
```

2. Sets 008/29 to “a” if the record does contain any of the MARC fields listed in column 3.

If no parameters are defined, fix_doc_aut_008_pos_29 is inactive.
If 008/29 is “b” to begin with, none of the above is performed.

fix_doc_aut_008_pos_32
Fix routine for the Authority record. No parameters are defined in column 3. The program performs the following:

1. Sets 008/32 to “a” if the record contains a 100 field, unless it also contains a 670 field whose $a subfield begins with a left square bracket – 670 $a [
2. Sets 008/32 to “b” if the record contains both a 100 field, and a 670 field whose $a subfield begins with a left square bracket – 670 $a [
3. Sets 008/32 to “n” if the record does not contain a 100 field.

12.2 fix_doc.lng

The `fix_doc.lng` table enables you to define the menu options that are displayed when the cataloger chooses the Fix Record function or the Derive New Record function from the Edit menu of the Cataloging module. The following is a sample of the fix_doc.lng table:

```
! 1   2 3           4
!!!!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS   N L Perform fixes as executed when document is updated (INS)
PUNC  N L Fix document's punctuation
04-01 N L Convert UNIMARC records to MARC 21 records
008   N L Update 008 field from 260 field
AUT   Y L Create authority record from current bibliographic record
ANA   C L Create an analytic record from the current record
```

Key to the `fix_doc.lng` table:

Column 1 - Procedure ID
This is the unique code by which the system identifies the procedure. It must be an ID defined in column 1 of the tab_fix table.

Column 2 - Fix Current Record/Derive New Record
This column defines whether a new record is going to be created when performing a fix routine, or if the current record is going to be fixed. The possible values are:

Y - Open as new record (unconditionally).

C - Open as a new record (only if the current record has been previously saved on the server).

N - Fix current record.

Routines that have been set to N appear under the Fix record option in the Edit menu. Routines that have been set to Y or C appear under the Derive New Record option in the Edit menu. Note that if the routine is set to C and the current record for the Derive routine is a local record that has not yet been saved on the server, then this routine is not displayed in the Fix Procedure window.

Column 3 - ALPHA
ALPHA code. Must always be L.

Column 4 - Text of menu option
Enter a description of the procedure up to 45 characters in length. This text will appear under the Fix record option and the Derive new record option in the Edit menu according to the setup of column 2.

It is possible to write external programs for fixing records. External programs can be written in any programming language and can be executed without linkage to ALEPH 500. They are particularly useful for special on-site developments.

The program must reside in `$aleph_exe` and it should have no extension. When the program is compiled, it will be placed in `$aleph_proc`.

12.3 fix_doc_track

This fix routine is used to store the change history of bibliographic and authority records in the new Z00T Oracle table. The fields that are tracked are listed in the Parameters column (Col. 3). For example:

```
! 1             2                    3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS2  fix_doc_track              100##,310##,500##,651##
```

Note that if there are multiple occurrences of a field, only the first one is tracked. In the above example, if the record has a 10010 field and a 10011 field, then only the one that is found first in the document will be tracked.

The change history may be viewed in the Cataloging GUI by selecting the **Edit\View Records History** menu option.

13 Locate Function

The Locate function of the Cataloging module enables you to find records in other databases or in your local database that are similar to the one currently being edited. The System Librarian is responsible for setting up the criteria that the system uses in order to determine which records are similar (for example, you can decide that if the records have the same words in the title and author fields, then the records are "similar"). You can define the criteria by editing the `tab_locate` table located in the library's tab directory.

Note that the criteria defined in this table also affect the Locate function in the Search function.

The `tab_locate` table defines the locate routine that is to be used when searching for a record in other databases. Multiple lines can be set up for one library, in which case

all lines are taken with an AND condition between them. The `tab_locate` table should include both the source and target library.

Following is a sample of the `tab_locate` table:

```
1         2     3             4           5                     6
!!!!!!!!!-!!!!!-!!!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!
USM01     100## ab           wau=       locate_str_3         3
USM01     245## -c           wti=       locate_str_0         90
USM01     008##              wyr=       locate_str_2

UNI01     100## a            wau=       locate_str_1
UNI01     245## a            wti=       locate_str_0
```

Key to the `tab_locate` table:

Column 1 - Library Code
Enter the library code of the library in which you want to locate records.

Column 2 - Tag and indicators
Enter a field tag that is used as a "locate" parameter. You can define specific indicators. Use the # character to indicate any indicator. Note that this is always the local tag (for example, see the tag definitions for locating in UNI01 -UNIMARC type library- from USM01 which is a MARC 21 library).

Column 3 - Subfield
Enter the subfields that will be used to build the locate string. The "-" sign can be used to mean "all subfields except for". For example, if this column is configured with: -ab, then all subfields except for 'a' and 'b' will be used.

Column 4 - Find command
Enter the WRD code that is used with the find command to search for similar records.

Column 5 - Extract function
The extract functions define how the contents of the field are going to be treated.

Extract functions:

locate_str_0:
Takes subfield content as is.

locate_str_1:
Runs "build_filing_key" on a subfield and takes the 2 longest words. A word must be at least two characters in order to be considered a "word". If the subfield has only one word, that word is taken.

locate_str_2:
Takes the year from the 008 field (position 8, length 4).

locate_str_3:
Works similarly to locate_str_1, but takes the number of longest words specified in the Column 6 parameters (for example, the three longest words).

locate_str_sys_no:
Uses a doc number in a specified field to perform an exact match.

Column 6 - Parameters
Enter the parameters that will be used by the extract function defined in column 5. Such parameters can be the number of words that will be used by locate_str_3 or the breaking procedure that will be used by locate_str_0 or locate_str_1.

Bases for the locate function are defined in the ALEPHCOM/TAB/LOCATE.DAT file. This file affects both the Locate function in the Cataloging module and the Locate function in the ILL module. Note that you can define a separate file for the Cataloging module. You do this by adding a locate.dat file to the catalog directory of the library (./pc_tab/catalog). This file must be in the same format of the locate.dat of the alephcom directory. Note that if the file is added, even if it is empty, then the bases in the Locate window of the Cataloging module are taken from the locate.dat file of the catalog directory. If the file is left empty, then no bases are displayed from the Cataloging module even if bases have been defined for the locate.dat file in the alephcom directory.

The "locate" section in the CATALOG/TAB/CATALOG.INI file defines whether or not the record found using the Locate function should be merged automatically with the current record.

14 Duplicate Record Function

The Duplicate Record function enables you to copy the currently displayed record and then edit the copy. The new record is located on your local drive.

It is up to you, the System Librarian, to determine whether the new record should be assigned automatically to the Home Library (the library to which the user is currently connected), assigned automatically to another specific library, or assigned to the library of the cataloger's choice (in which case, a list of all available libraries is displayed for the cataloger to choose from).

In order to determine which of the above is in effect, open the CATALOG.INI file (found in the client's CATALOG/TAB directory). Go to the [DuplicateRecord] section. Following is an example of what you can find there:

```
[DuplicateRecord]
Library=HOME
```

If you want the new record to be assigned automatically to the Home library, type *HOME* to the right of the equal (=) sign. If you want a different library, type the code for the library, for example, *USM01*. If you want the cataloger to choose from a list of all available libraries (that is, all libraries listed in the per_lib.ini file in the CATALOG/TAB directory), type *ALL*. If you want to define the list of libraries that the cataloger can choose from, type the list of libraries. For example:

```
Library=USM01,USM20,ACC01,UBW01
```

15 Importing Updated Tables

You can determine whether or not the system automatically imports updated Catalog tables when the Cataloger opens the Cataloging module. To determine this, go to the ALEPHCOM/TAB directory and open the ALEPHCOM.INI file. Go to the section labeled [Package]. Following is an example of the relevant section:

```
[Package]
AlwaysImportFiles=Y
```

Enter *Y* to the right of the equal sign if you want the system to import updated tables automatically.

Enter *N* to the right of the equal sign if you want the system to ask the Cataloger whether or not he wants to import the updated tables.

Note that this section also determines whether or not the updated printing templates package is automatically downloaded to the client when connecting to any of the modules.

16 Floating Keyboard

The Floating Keyboard enables you to insert characters that are not present in your workstation's standard keyboard. The Floating Keyboard is configured by the System Librarian according to the needs of the library. Following is an example of a floating keyboard.

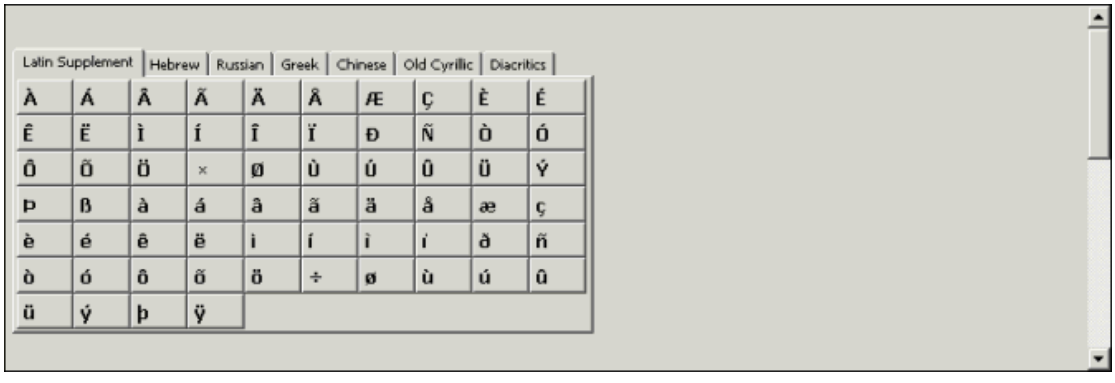
Three files define the Floating Keyboard setup:

- Keyboard.ini
- Keyboard.txt
- Font.ini

All files are located in the ALEPHCOM/TAB directory.

Keyboard.ini

The keyboard.ini defines the configuration settings.



The sample below matches the example of a Floating Keyboard shown above.

```
[Main]
Title=Keyboard

[WindowLocation]

KeyboardWindowPosition=189,267
KeyboardWindowRelocate=Y

[Tabs]
NoTabs=7

[Tab1]
Caption=Latin Supplement
NoCols=10
BtnWidth=40
BtnHeight=25

[Tab2]
Caption=Hebrew
NoCols=10
BtnWidth=40
BtnHeight=25

[Tab3]
Caption=Russian
NoCols=10
BtnWidth=40
BtnHeight=25

[Tab4]
Caption=Greek
NoCols=10
BtnWidth=40
BtnHeight=25

[Tab5]
Caption=Chinese
NoCols=10
BtnWidth=40
BtnHeight=25

[Tab6]
Caption=Old Cyrillic
NoCols=10
BtnWidth=40
BtnHeight=25

[Tab7]
Caption=Diacritics
NoCols=10
BtnWidth=40
BtnHeight=25
```

Table sections:

[WindowLocation]
This section defines the position of the Floating Keyboard and whether or not it is possible to relocate it. Note that in the Cataloging module, this option is not in use. In the Cataloging module, the keyboard is displayed in the lower pane.

[Tabs]
This section defines the number of tabs that appear in the Floating Keyboard.

[Tab(number)]
For example, [Tab3]

This section defines the configuration settings for each tab of the keyboard.

Caption: Defines the caption of the tab (for example, Russian).

NoCols: Defines the number of columns for the tab.

BtnWidth: Defines the width of the character keys for the tab.

BtnHeight: Defines the height of the character keys for the tab.

Keyboard.txt

The Keyboard.txt file defines the characters that are displayed in each tab.

The sample below matches the example of a keyboard shown above.

```
! Unicode code
!!!!!!!!!!!!!

[Latin Supplement]
!!!!!!!!!!!!!
\00C0
\00C1
\00C2
\00C3

etc...

[Hebrew]
!!!!!!!!!!
\05D0
\05D1
\05D2
\05D3

etc...

[Russian]
!!!!!!!!!!
\0410
\0411
\0412
\0413

etc...

[Greek]
!!!!!!!!!!
```

```
\0386
\0388
\0389
\038A

etc...

[Chinese]

!!!!!!!!!!

\4E10
\4E11
\4E12
\4E13

etc...

[Old Cyrillic]

!!!!!!!!!!
\0410
\0411
\0412
\0413

etc...

[Diacritics]

!!!!!!!!!!
\02BB
\0307
\0324
\0310

etc...
```

This file contains one column. This column contains the Unicode value of the character that is inserted in the cataloging draft.

Note that the table is divided according to the tabs for the keyboard. Each section should be entered in the same order in which it is defined in the Keyboard.ini file. The link between a tab in the two files is determined by order and not by the capture.

Font.ini

The Font.ini file contains the font definitions.

Note that is possible to define different fonts for different Unicode ranges (columns 2 and 3 of the file).

The following is an example of the Font.ini file for the floating keyboard:

```
!       1              2     3                   4             5 6 7 8
9
!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-
!!-!!!!!!!!!!!!!!!!!
AlephKeyboard        0000 00FF Tahoma                          Y N N
16 DEFAULT_CHARSET
```

```
AlephKeyboard        0401 045F Tahoma                          Y N N
16 DEFAULT_CHARSET
AlephKeyboard        0384 03CE Tahoma                          Y N N
16 DEFAULT_CHARSET
AlephKeyboard        05D0 05EA Tahoma                          Y N N
16 DEFAULT_CHARSET
AlephKeyboard        0000 FFFF Bitstream Cyberbit              Y N N
16 DEFAULT_CHARSET
```

For more information on the Font.ini file refer to the Font Definitions (Font.ini file) section of the General chapter.

17 Authorizations

17.1 Allowed and Denied Tags

The `permission.dat` table, located in the library's `pc_tab/catalog` directory, defines allowed and denied tags for different catalogers.

Following is a sample of the table:

```
!1         2     3 4
!!!!!!!!!!-!!!!!-!-!!!!!!!!!!
YOHANAN    ##### Y
YOHANAN    650## N
OMRI       ##### Y
OMRI       100## N
TAMI       ##### Y
TAMI       245## N
YIFAT      ##### N
```

Key to `permission.dat`:

Column 1 - User Name
This is the unique string by which the system identifies the cataloger/user.

Column 2 - Tag Code
This column contains the allowed or denied tag and indicators. Use the hash (##) as a placeholder for undefined tags and/or indicators (for example, 100## means tag 100 any indicators; ##### means ALL tags).

Column 3 - Type of Permission
Values are Y and N. Y is used for allowed tags and N for denied tags. In the above sample of the table, the user OMRI is authorized to edit all fields except for the 100 field.

In the Cataloging module, denied tags will appear in a different color.

A user that does not have an entry in the `permission.dat` table is denied permission to edit any tag. If the library does not want to use the `permission.dat` mechanism, the table can be removed and all users will then be allowed to edit any tags.

Note that users that have cataloging proxies do not need to be listed in this table. When the cataloging tables are repacked, users of this type are granted the rights assigned to their cataloger proxy.

17.2 Cataloging "OWN" Permissions

The system librarian can assign a group of allowed OWN values for a cataloger. This can be done by setting up the `tab_own` table in the library's `tab` directory.

Up to five different OWN values of cataloging records can be allowed for a single OWN value of a user.

Following is a sample of the table:

```
!    1          2          3          4          5          6
!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!
CAT         AA         BB
CAT1        BB         CC         DD
```

In this example, any user with the value CAT in its Cat. OWN Permission field has authorization for updating records with OWN values of AA and BB. Those users with the value CAT1 in their Cat. OWN Permission field have authorization for updating records with OWN values of BB, CC and DD.

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:

```
!    1          2          3          4          5          6
!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!
MECAT       ME###
MECAT1      #####
MECAT2      ##########
MECAT3      ME###      GR####

MACA##      MACAT1
```

Based on the sample above:

ME### includes, for example, MEDUC, MELEC, and so on.

includes all OWN values that are up to five characters.

########## includes all possible OWN values (this is equal to the GLOBAL authorization).

Note

If a User's OWN value needs to be assigned more than five different record's own values, (without using the hash (#) character as a wildcard), you can define multiple lines for the same user's OWN. For example:

```
!     1          2          3          4          5          6
! !!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!
! CAT        MED        HYL        HIL        LAM        LAW
```

```
! CAT       LIT        MUS        WID        HILR       BCU
```

Key to the tab_own table:

Column 1 - User's OWN Permission
This column contains the value of the Cat. OWN Permission field assigned to the user(s). Use the hash (#) character as a placeholder for any character. For example, CAT## includes users with OWN Permission CAT, CAT1, CATXX, and so on.

Columns 2 to 6 - Record's OWN value
Columns 2 to 6 contain the record's OWN values which the user with the OWN permission defined in column 1 is allowed to update. Use the hash (#) character as a placeholder for any character. For example, ME### includes, MEDUC, MELEC, and so on.

If a catalog proxy is assigned to the user (see the Staff Privileges - User Information - Password section, then the OWN values for the user are taken from the proxy's record.

17.3 Holdings Filter

The holdings records displayed in the Cataloging navigation tree and in the *HOL records* tab in the Cataloging module lower pane can be filtered based on the holdings record's OWN field. Only users with an OWN Permission value that is the same as the value in the OWN field in the holdings record will be able to see it.

This filter is dependent on the `tab_own` table. In addition, the OWN-FILTER value in tab100 must be Y.

18 Merging Records

The fix_doc_merge program is used to merge or overlay cataloging records according to the merging routines defined in the `tab_merge` table located in the library's tab directory. Column 3 of the tab_fix table is used to define the merging routine that matches the relevant section in the tab_merge table.

The following is a sample of an entry for the fix_doc_merge program in the tab_fix table:

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
FIX   fix_doc_merge                  OVERLAY-01
```

The "OVERLAY-01" routine must match a routine in the `tab_merge` table. The following is a sample of the `tab_merge` table:

```
!   1                    2                    3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!->
OVERLAY-01 merge_doc_overlay              01
OVERLAY-02 merge_doc_overlay              02
OVERLAY-03 merge_doc_replace
```

Key to the tab_merge table:

Column 1 - Routine Name
This column is used to define the merging routine. It matches the routines defined in column 3 of the tab_fix table.

Column 2 - Merging Program
This column contains the merging program. The following are the available options:

merge_doc_replace:
This program replaces the contents of the original record with the contents of the new record, retaining the CAT fields from both records.

merge_doc_overlay:
This program merges/overlays the record according to the overlay specifications defined in the tab_merge_overlay table of the library's tab directory.

merge_doc_adv_overlay:
This program merges/overlays the record according to the overlay specification defined in the tab_merge_adv_overlay table of the library's tab directory. This table shares the same purpose as the tab_merge_overlay table and acts in a similar manner, with an added level of complexity. The additional functionality is based on determining 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 and the accompanying "weights" table that is used to evaluate the two records. This program is usually used when loading records into the system.

Column 3 - Merge Set
This column contains the merge set to be applied when the merge_doc_overlay or the merge_doc_adv_overlay programs are performed. The merge set must match a merge set defined in the `tab_merge_overlay`/`tab_merge_adv_overlay` tables of the library's tab directory.

Note that when the overlay programs are used, the system librarian is in charge of defining which fields are retained or overwritten when merging/overlaying two cataloging records. This is done by editing the library's `tab_merge_overlay/tab_merge_adv_overlay` tables located in the library's tab directory.

The merge_doc_overlay function runs when the Paste record option is selected from the Edit menu of the Cataloging module. The system uses the definitions of the `tab_merge_overlay` table if the following line is defined in the `tab_fix` table of the library's `tab` directory:

```
MERGE fix_doc_merge          (routine name for tab_merge)
```

The merge_doc_overlay function can also be used when the Locate Similar Records option is selected from the Edit menu of the Cataloging module. The system uses the definitions of the `tab_merge_overlay` table if the following line is defined in the tab_fix table of the library's tab directory:

```
LOCAT fix_doc_merge          (routine name for tab_merge)
```

Following is a sample of the `tab_merge_overlay` table:

```
!1 2 3   4
!!-!-!-!!!!!
01 1 Y #####
01 1 N 008##
01 1 C 245##

01 1 Y 245##
```

Key to the tab_merge_overlay table:

Column 1 - Merge set
This column is used to define different merging routine sets. Up to 99 different merging routine sets can be defined; the values are 01 to 99. Note that the routine must match the definitions in the tab_merge table of the library's tab directory. For example, if you want to work with the 03 routine, then the relevant fix_doc_merge section of the tab_fix table must be attached to the routine that in the tab_merge table is set to work with the merge set 03. Additionally, note that the lines of the table are limited to 99.

Column 2 - Merging direction
Values are 1 and 2. 1 defines lines for the original record, that is, the document into which fields are merged/pasted. 2 defines lines for the document from which fields are copied.

Column 3 - Action
Values are Y, N and C:
Y - For the original record (1) - retains the field.
For the copied record (2) - copies the field.
N - Does not retain the field.
C - Retains the field only if it does not appear in the other record.

Column 4 - Tag code
This column contains the field tag and its indicators. Use the hash (#) as a placeholder for undefined tags and/or indicators (for example, 100## means tag 100 any indicators; ##### means ALL tags).

This column can also be used for subfield and subfield contents to use as filters, as shown in the following example:

```
      01 2 Y 590##5,*abc*
```

In the above example, the tag 590 is disregarded if subfield $5 of the field does not contain the string "abc" as part of its contents.

In the example above, all fields are taken from the original document (1), except the 008 field. The 245 field is always taken from the copied record. If the copied record does not have a 245 field, the 245 field of the original record is retained. Otherwise it is overlaid from the second document to the original record.

Note that the search for the code is sequential. For example:

```
01 1 N 008##
01 1 Y #####
```

At first, the system will not take the 008 field because of the N in column 3 for the field. Then, the system continues "reading" the next line that defines that all fields should be taken. The result is that the 008 field is taken, too.

19 Updating the Tables Package

After making changes to any of the tables of the catalog directory (`$data_root/pc_tab/catalog`), the system librarian is in charge of repackaging the cataloging tables. The cataloging tables are repackaged by performing UTIL M/7. This updates the packaged file of tables (pc_cat.pck) in the library's catalog directory. When a user connects to a home library in the Cataloging module, the system compares the tables on the client with the date of the pc_cat.pck package on the server. If the dates are different and the *AlwaysImportFiles* flag in alephcom.ini (under [Package]) is set to "N", then the user is prompted to update the tables on the client. If the dates are different and the *AlwaysImportFiles* flag is set to "Y", then the tables are imported automatically.

Among other tables/files, the catalog directory contains cataloging forms (for example, 008_bk.lng for MARC 21), cataloging templates (for example, 008_bk.lng for MARC 21), help files (in the HTML subdirectory), codes for the FMT field (formats.lng), field contents for fixed text fields (tag_text.dat), list of valid tags and aliases (codes.lng), and so on.

20 Subfield Punctuation

The `tab_subfield_punctuation` table in the library's tab directory is used to define subfield 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 than the authority record. This data should be retained. In MARC, authority records do not have end punctuation while bibliographic records do. The `tab_subfield_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 subfields retained from the bibliographic record (for example, between subfield $a - personal name - and subfield $t - title of MARC 21 600 field). The following is a sample of the `tab_subfield_punctuation` table:

```
!2   345     6
!-!!!!!-!-!-!!!!!!!!!!-!!!!!!!!!!
A 1#### a   .       .
A 1#### d   .           -.
A 100## a 4 ,
A 100## d 4 ,
A 110## b   .           .
```

Key to `tab_subfield_punctuation`:

Column 1 - Program code
Use always "A".

Column 2 - Tag and indicators
Contains the field tag with indicators for which subfield punctuation is being defined. Use the hash (#) as a placeholder for undefined indicators.

Column 3 - Subfield code
Enter the subfield to which the end punctuation is going to be added.

Column 4 - Following subfield code
Enter the subfield that follows after the end punctuation added to the subfield defined in the previous column.

Column 5 - Punctuation to add
Enter the punctuation signs that should be added to the subfield.

Column 6 - If punctuation
This column is used to determine whether or not the punctuation defined in column 5 is added to the field. If the field already ends with the punctuation defined in column 6, punctuation from column 5 is not added.

21 Validation of Contents of a Field

You can set up ALEPH to check the contents of some fields. This is done through the `check_doc_line_contents` table in the library's `tab` directory.

Following is a sample of the `check_doc_line_contents` table:

```
!  2     3         4                       5
!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!
## 020## a isbn
## 022## a issn
## 7#### x issn
## 260## c range                1850 2002
## 260## c number_length        4

## 022## a length               9
```

Key to the `check_doc_line_contents` table:

Column 1 - Record's format
Enter a specific record format (for example, BK), or use ## as a wildcard to indicate that the field is appropriate for any format. Refer to Record Formats on page 9 for more information on record formats.

Column 2 - Field code
Field code to be checked. Use the hash (#) as a placeholder for undefined tags and/or indicators (for example, 020## or 7####).

Column 3 - Subfield code
Enter the subfield code of the subfield to be checked. If the column is left blank, then the field is taken as is.

Column 4 - Name of check program
The existing check programs are **isbn**, **issn**, **issn-isbn, length**, **number_length**, and **range**:

isbn - verifies that the ISBN entered in the field is a valid ISBN (including check digit). This routine is capable of validating both types of ISBN: 10-digit ISBNs and 13-digit ISBNs.

Both 10-digit ISBNs and 13-digit ISBNS are considered valid.

issn - verifies that the ISSN entered in the field is a valid ISSN (including check digit).

issn-isbn – Verifies that the field is either a valid ISSN number or a valid ISBN number. ISSN numbers may be missing the hyphen and ISBN numbers may be 10 or 13 digit numbers.

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.

Column 5 - Values to check
For **length** enter <length> (for example, for subfield $c of MARC 21 field 260, the length is **4** for the year).
For **range** enter <from> <to> (for example, for subfield $c of MARC 21 field 260, enter reasonable values for the range of the year, say, 1850 - 2001).

22 Check Field Occurrences and Dependency between Fields

Definitions for field occurrences and dependency between fields for checking routines are set up in the `check_doc_doc` table in the library's tab directory.

The table contains two sections:

- OC
- D

Following is a sample of the **OC section:**

```
OC BK 5001 00 01 100## 110## 111## 130##
OC XX 5002 01 01 245##
OC BK 5003 01 01 260##
OC SE 5007 01 01 310##
```

This section enables you to define which fields are mandatory and their repeatability.

Key to the OC section of `check_doc_doc`:

Column 1 - Section ID
Enter OC for each line of this section of the table.

Column 2 - Record format
Enter a specific record format (for example, BK), or use XX as a wildcard to indicate that the check is appropriate for any format. Refer to Record Formats on page 9 for more information on record formats.

Column 3 - Error message code
Enter the code of the error message that is displayed in the Cataloging module. The code should match the definitions of the check_doc.lng table located in the library's tab directory.

Column 4 - Minimum of occurrences
00 indicates that the field is not mandatory. 01 indicates that the field must be present.

Column 5 - Maximum number of occurrences
If the field is not repeatable, enter 01. If the field is repeatable, you can use values 02 to 99 to define that the field can be repeated up to a particular number of times according to the selected value.

Column 6 - Field code
Enter the field code of the fields for which occurrences are being defined. Up to 5 field codes can be entered (with "OR" implied).

In the above sample of the table, the first line indicates that a record can have only one occurrence of either MARC 21 field 100, or 110, or 111 or 130. These fields are not mandatory.

Note that repeated fields with the same subfield $6 (this subfield links fields that are different script representations of each other) are considered a single occurrence of the field. This avoids incorrect repeatability messages.

Based on the above sample, if the record contains two occurrences of MARC 21 field 100, as follows:

then no error message is displayed when checking the record; the system considers both 100 fields as a single occurrence.

Following is a sample of the **D section:**

```
D  BK 7003 2450#   Y 1####   N
D  BK 7004 2451#   Y 1####   Y
```

This section of the table enables you to define dependencies between fields, such as if one is present another must be present, or if one is present another must not be present.

Key to the D section of `check_doc_doc`:

Column 1 - Section ID
Enter D for each line of this section of the table.

Column 2 - Record format
Enter a specific record format (for example, BK), or use XX as a wildcard to indicate that the check is appropriate for any format. Refer to Record Formats on page 9 for more information on record formats.

Column 3 - Error message code
Enter the code of the error message that is displayed in the Cataloging module. The code should match the definitions of the `check_doc.lng` table located in the library's tab directory.

Column 4 - Field code
Field code for the first part of the condition. Use the hash (#) as a placeholder for undefined tags and/or indicators (for example, 100## or 1#0##)

Column 5 - Type of dependency
This column defines whether the check relates to the field being present or not. Values are **Y** and **N**. Use **Y** to define that the field is present. Use **N** to define that the field is not present.

Column 6 - Field code
Field code for the second part of the condition. Use the hash (#) as a placeholder for undefined tags and/or indicators (for example, 100## or 1#0##).

Column 7 - Type of dependency
This column defines whether or not the check relates to the field being present. Values are **Y** and **N**. Use **Y** to define that the field is present. Use **N** to define that the field is not present.

In the above sample of the section, if a record has a 245 field with 0 as first indicator, then the 1XX fields must not be present. If the record has a 245 field with 1 as first indicator, then a 1XX field must be present.

23 Forbidden Errors and Triggers

The `check_doc_mandatory` table can be used to define whether error messages produced by cataloging check routines should activate a trigger or be defined as forbidden.
A cataloging error defined as forbidden does not allow the user to save/update the record, while errors that activate triggers allow database update. The triggers are automatically assigned CAT as the department in the Trigger Department field. Triggers can later be retrieved using the Triggers node on the Cataloging tab; the triggers of a specific record can be retrieved through the **Record's Triggers** command from the **Edit** menu of the Cataloging module.

For example, if the `check_doc_doc` table is used to define that a MARC 21 record must have a 245 field (for example, OC XX 5002 01 01 245##), then you can set error message 5002 to activate a trigger or to be defined as forbidden.

In addition, through the check type (column - col.1), you can define that the error message activates a trigger or is forbidden only in particular instances of the system,

such as, for example, when records are updated or created from the Cataloging module.

Following is a sample of the `check_doc_mandatory` table:

```
1           2   3                     4
!!!!!!!!!!!!!!!!!!!-!!!!-!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
CATALOG-DELETE       0011 M ADM record points to current document.
                     0012 M HOL record points to current document.
                     0013 M BIB record points to current document.
CATALOG-INSERT       0101 T Field is a duplicate entry in the INDEX
file.
CATALOG-INSERT       0110 T Field is a new heading in the index list.
CATALOG-INSERT       0161 T ISBN is incorrect.
CATALOG-INSERT       0162 T ISSN is incorrect.
                     5001 M A record cannot have more than 1 main
entry (1XX).
                     5002 M Required 245 field is either missing or
duplicated.
                     5008 M Required 008 field is either missing or
duplicated.
                     9999 T Too many errors (must be less than 40).
```

Key to the check_doc_mandatory table:

Column 1 - Check type
The check type defines when the check program is performed. Check programs are assigned to check types in the check_doc table of the library's tab directory. The 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: Check programs attached to the NAV-MAP-DELETE check type are run when the Total Delete option is selected from the Record Manager in the Cataloging module.

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.

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

Column 2 - Identifying number of the check program
Enter the error code of the check program. User-defined error codes are defined in the `check_doc.lng` table in the library's tab directory. System-defined error codes are defined in the `check_doc` table in the `$aleph_root/error_lng`directory.

Column 3 - Type of error
This column is used to define the type of error. Values are M and T. Errors of type M are considered forbidden errors and do not allow the user to update the record. Errors of type T activate a trigger and allow database update. The record's triggers can be retrieved through the Record's triggers option from the Edit menu of the Cataloging module.

Column 4 - Error message
Optional free-text column. It is non-functional, for information only.

Note that if error code 9999 (Too many errors) is not defined in the table, it is considered by the system as a forbidden error (type 'M').

24 Checking Routines for New Headings in the Headings List

The system librarian can define which fields are ignored for purposes of the check message that informs the cataloger that the heading is a new heading in the headings list (acc file). This is done by including the field in the `check_doc_new_acc` table in the library's tab directory.

Following is a sample of the `check_doc_new_acc` table:

```
! 1
!!!!!
245##
260##
```

Key to the check_doc_new_acc table:

Column 1 - Field code
Enter the 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. In the above sample, the title headings and the imprint headings are ignored by the checking routine for new headings in the list.

Note that for the "New headings" check routine to be performed, the check_doc_new_acc program should be listed in the `check_doc` table of the library's tab directory. The `check_doc` table lists all the checking programs that are run when the user chooses the "Check Record" function.

25 Checking Routines for New Headings in the Bibliographic and Authority Headings List

The `check_doc_new_acc_aut` table in the library's tab directory defines the fields that should be ignored when checking for new headings in the Headings List of the relevant authority library and in the Headings List of the bibliographic library.

Following is a sample of the `check_doc_new_acc_aut` table:

```
! 1
!!!!!
```

```
245##
260##
```

Key to the check_doc_new_acc_aut table:

Column 1 - Field code
Enter the field code of the fields that should be ignored while checking for unique headings in the Headings List (ACC index) of the bibliographic library and in the Headings List of the relevant authority database. Use the hash (#) as a placeholder for undefined tags and/or indicators. In the above sample, the title headings and the imprint headings are ignored by the checking routine for new headings in the bibliographic list of headings and in the list of headings of the relevant authority library.

Note that for the "New headings" check routine to be performed, the check_doc_new_acc_aut program should be listed in the check_doc table of the library's tab directory. The check_doc table lists all the checking programs that are run when the user chooses the "Check Record" function.

26 Checking Routines for New Direct Indexes (IND)

The system librarian can define which fields are ignored when the system checks whether or not a duplicate record is opened in the Direct (Z11) Index. This is done by including the field to be ignored in the check_doc_unique_index table in the library's tab directory.

Following is a sample of the check_doc_unique_index table:

```
! 1
!!!!!
050##
020##
```

Key to the check_doc_unique_index table:

Column 1 - Field code
Enter the field code of the fields that should be ignored while checking for unique headings in the Direct Request Index. Use the hash (#) as a placeholder for undefined tags and/or indicators. In the above sample, the Library of Congress call number and the ISBN are ignored by the checking routine for new headings in the list.

Note that for the "Duplicate Direct Index" check routine to be performed, the check_doc_unique_index program should be listed in the check_doc table of the library's tab directory. The check_doc table lists all the checking programs that are run when the user chooses the "Check Record" function.

27 Locking Records

27.1 Locking Period for Locked Records

Locked records are automatically unlocked after a period defined by the system librarian by defining the setenv doc_lock_period variable in the `pc_server_defaults` table located in the `$alephe_root` directory. The period is defined in seconds. By default, the variable has been set up to lock records for one hour:

```
setenv doc_lock_period          3600
```

27.2 Lock Status Message

When a cataloger locks a record, the phrase "Locked by current user" is displayed in the Cataloging bar, informing the user that the record has been locked. In addition, when a cataloger loads a record locked by another cataloger, the phrase "Locked by another user" is displayed in the Cataloging bar informing the cataloger that the record is locked. This text can be modified by editing the following entries from the `pc_cat_c0203`file of the `$aleph_root/error_lng`directory:

```
2002 0000 L [Locked by another user] System No. $3 - Format $4 - $1
($2)
2003 0000 L [Locked by current user] System No. $3 - Format $4 - $1
($2)
```

28 Check Routines for Check Record

The `check_doc` table is used to define the check programs that are used in the system and in the environment in which these programs are used.

The following is a sample of the table:

```
!        1                          2                            3
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CATALOG-INSERT       check_doc_new_acc
CATALOG-INSERT       check_doc_new_acc_aut
CATALOG-INSERT       check_doc_unique_index

Z39-INSERT           check_doc_line
Z39-INSERT           check_doc_line_contents

Z39-REPLACE          check_doc_new_acc
Z39-REPLACE          check_doc_new_acc_aut

CATALOG-DELETE       check_doc_delete_lkr
CATALOG-DELETE       check_doc_delete_item
```

Note that up to 100 programs can be defined in the `check_doc` table.

Key to the check_doc table:

Column 1 - Check Type
Enter the check type that defines when the check program is performed.

Column 2 - Check Program
Enter the check program(s) that should be performed for the specific check type defined in column 1.

Column 3 - Program Arguments
Contains additional information about the programs, such as table names. This column is used to define additional parameters for the check programs.

28.1 Check Types Available for Column 1 of the check_doc Table:

The following are the available check types:

CATALOG-INSERT: Check programs attached to the CATALOG-INSERT check type are performed when the cataloging record is saved, updated or when the Check Record option is selected from the Cataloging module.

CATALOG-DELETE: Check programs attached to the CATALOG-DELETE check type are performed when the Delete Record from Server option is selected from the Cataloging module.

BATCH-DELETE: Check programs attached to the BATCH-DELETE check type are performed when the Delete Bibliographic Records (p-manage-33) batch process is run.

NAV-MAP-DELETE: Check programs attached to the NAV-MAP-DELETE check type are performed when the Delete Bibliographic record option is selected from the Navigation Window of the Search module.

Z39-INSERT: Check programs attached to the Z39-INSERT check type are performed when a record is inserted via Z39.50 ES Update.

Z39-REPLACE: Check programs attached to the Z39-REPLACE check type are performed when a record is inserted via Z39.50 ES Update.

Z39-DELETE: Check programs attached to the Z39-DELETE check type are performed when a record is inserted via Z39.50 ES Update.

28.2 Check Programs Available for Column 2 of the check_doc Table

The following are the available check programs:

check_doc_852
This program checks whether the sublibrary and collection codes - cataloged in subfields $b and $c of the MARC 21 location field (852) of the holdings record - match the definitions of the tab_sub_library and tab40 tables. Note that this program should be included in the check_doc table of the holdings library (xxx60).

check_doc_853

This routine checks if patterns (853/4/5 tags) share the same subfield $$8 value.

check_doc_853x
This program checks the presence and the validity of mandatory subfields in the MARC 21 853/4/5 and 853X/4X/5X fields. The check_doc_853x program also

checks dependencies between subfields. For example, if the 853 field has a subfield $a, then subfield $a must also be present in the 853X field.

check_doc_adm_lkr
This program checks whether the bibliographic record to which the LKR field in the administrative record is pointing (subfield $b) is already linked to another administrative record. This program should be used only in administrative libraries (XXX50).

check_doc_aut_008
This check routine produces an error if one of the following is detected in the authority record:

- Position 29 in line 008 has the value ‘n’, and a 4#### or a 5#### line exists.
- Position 29 in line 008 has the value ‘a’ or ‘b’, and no 4#### or 5#### line exists.
- Position 32 in line 008 has the value ‘n’, and a 1000# or a 1001# line exists.
- Position 32 in line 008 has the value ‘a or ‘b’, and no 1000# or 1001# line exists.

check_doc_aut_5xx
This program checks whether or not the 5XX field (*See also* from tracing field) cataloged in the authority record has a corresponding entry in the "GEN" index. If there is no matching heading in the "GEN" index, an error message is displayed. If the 5XX field has a corresponding entry, the program also checks whether this entry derives from a 1XX Heading field or from a *See from* tracing field (4XX). If the matching heading derives from a *See from* (4XX), then an error message is displayed.

Note that for the implementation of this program the 5XX fields should be sent to a "GXX" headings index in the authority library. The check_doc_aut_5xx compares the entries in these indexes with the entries in the "GEN" index.

check_doc_aut_duplicate
This program checks whether the authority heading already exists in the GEN index of the authority database. Note that this program should be included in the `check_doc` table of the authority database (xxx10).

check_doc_doc
This program checks field occurrences and dependencies between fields, according to the definitions of the `check_doc_doc` table.

check_doc_line
This program checks the validity of indicators and subfields; the presence of mandatory subfields according to the definitions of the `check_doc_line` table. This program also checks dependencies between subfields.

Checks for fixed fields are also done by this program. To avoid these checks, in column 3 of the `check_doc` table, place the word `-FIXED` as seen in the following example:

```
CATALOG-INSERT       check_doc_line                  -FIXED
```

check_doc_line_contents:
This program checks the contents of some fields according to the definitions of the `check_doc_line_contents` table.

check_doc_lkr
This program checks the validity of the library and document number in the LKR field (subfields $l and $b).

The parameter, PAR ,is set in column 3 of the check_doc table.

The check routine is set in ./<bib_library>/tab/check_doc

```
!        1                          2                              3
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!>
CATALOG-INSERT       check_doc_lkr                  PAR
```

After the PAR is set in the check_doc_lkr routine, the system activates the check_doc_lkr routine and an additional routine to check if LKR $$aPAR has a corresponding LKR field in the linked document. This includes a validity check of LKR$$r. The check algorithm is as follows:

For each LKR $$aPAR with $$b<linked document number> and $$l<BIB Library> fields on document A:

- retrieve the linked document number (from $$b)
- retrieve the 7xx tag (from LKR$$r)
- check the linked document (document B) for a LKR field with the following information:
  - $$b with document A's system number
  - $$r with paired 7xx tags

The following are the possible 7xx tag pairs:

- 78000-78500
- 78005-78504
- 78006-78505
- 78004-78507
- 78007-78501
- 78001-78506
- 7750-7750
- 7870-7870
- 7670-7650
- 7760-7760

For example, if document A's LKR contains 78000 in $$r, then document B's LKR contains 78500 in $$r.

If there is no PAR in the linked record or there is no match in LKR$$r of the linked record (as defined in list of 7xx tag pairs), a check doc message 0217 is displayed. For example: “LKR – There is no PAR and/or invalid paired subfield "r" in the linked document USM01: 000000847”. This message is set in ./aleph/error_lng/check_doc.

```
!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
0217 L LKR – There is no PAR and/or invalid paired subfield "r" in
linked document $1:$2.
```

check_doc_locate
This program checks if there are records in the database that are similar to the record currently being updated. The mechanism used by this program is determined by the definitions in the `tab_locate` table of the library's tab directory.

check_doc_match
This program checks if there are records in the database that are duplicates of the one currently being edited. The mechanism used by this program is determined by the definitions in the `tab_match` table of the library's tab directory. Note that in the `tab_match` table, the match code 'CAT' (column 1 of the table) is used to specify the matching routines performed by the check_doc_match program.

check_doc_new_acc
This program checks whether or not a new record is opened in the headings list of the library.

check_doc_new_acc_aut
This program checks whether or not the cataloged heading is a new entry in the headings list of the bibliographic library or of the authority library.

check_doc_paired_fields This program checks the following two aspects related to fields that are linked by subfield $6 (subfield $6 contains data that links fields that are different script representations of each other), such as:

```
1001  L $$601$$a[Name in Chinese script].
1001  L $$601$$aShen, Wei-pin.
```

If the record contains a field with subfield $$6 <value numeric - other than '00'>, there must be an additional field with the same tag and indicators and the same $$6 value.

If the record contains a field with subfield $$6 <value numeric - other than '00'>, there must be no more than two fields that have the same $$6 value.

check_doc_tag_text
This program checks the validity of text entered into a subfield defined in the `check_doc_tag_text` table.

check_doc_unique_index
This program checks whether or not a duplicate record is opened in the Direct (Z11) index.

check_doc_url
This program checks the validity of an external URI/URL link from subfield $u of fields such as 856, 505, 530, and so on. The default timeout for this program is 10

seconds. In order to change it, use the setenv check_url_timeout parameter in `pc_server_defaults`.

Note that adding the prefix http:// is not mandatory. If no prefix is entered, the program will assume that the http protocol is used.

28.3 Check Programs For Document Deletion

Following are the available check programs for document deletion:

check_doc_delete_lkr
This program checks if there are any links from the record to be deleted to another record. It checks ADM, ITM and HOL links from the record to records in other libraries.

check_doc_delete_lkr_no_ana

This program checks if there are any links from the record to another record in the same library that need to be deleted. It checks UP (up link), ANA (analytic) and PAR (parallel) links.

check_doc_delete_lkr_itm
This program checks if there are any ITM links from the record to be deleted to another record. In a MAB environment, it also checks links via the 090i field.

check_doc_delete_item
This program checks if the record to be deleted has any associated items.

check_doc_delete_item_opac

This program checks if the record to be deleted has any associated items that are viewable in the OPAC, i.e. column 10 of the tab15.lng line is set to Y.

check_doc_delete_order
This program checks if the record to be deleted has any associated order.

check_doc_delete_copies
This program checks if the record to be deleted has any associated subscriptions.

check_doc_delete_loan
This program checks if the record to be deleted has any associated items on loan.

check_doc_delete_hold
This program checks if the record to be deleted has any photocopy requests or/and hold requests associated to the items linked to it.

check_doc_delete_aut_bib
This program checks if the authority record to be deleted has any bibliographic records associated with the heading of the record.

check_doc_delete_object
This program checks if there are digital objects that are associated with the record being deleted. Note that an object is related to an ADM record depending on the value in the object's 'Cat. Sublibrary' field.

Note that in the Messages tab (lower pane) window of the Cataloging module, the View Related button is enabled with the following check routines:

- check_doc_unique_index

- check_doc_delete_lkr
- check_doc_locate
- check_aut_duplicate

The button retrieves the related record associated with the message displayed for the record being checked.

29 Fixed-length Fields Checking Routines

Fixed-length fields checking routines are table-driven. These routines are flexible and can be customized by the system librarian. Each fixed-length field has its own table for defining validation routines; the structure of the table is the same for all fields. Currently, fixed-length validation routine tables have been defined for MARC 21 006, 007, 008 and LDR (leader), and for UNIMARC 100 and LDR (leader).

The fixed-length tables for MARC 21 are the following:

- check_doc_field_006
- check_doc_field_007
- check_doc_field_008
- check_doc_field_ldr

The fixed-length tables for UNIMARC are the following:

- check_doc_field_100
- check_doc_field_ldr

Note that for the fixed-length validation checks to be functional, the check_doc_line program must be listed in the `check_doc` table located in the library's tab directory.

Following is a sample of a `check_doc_field_<tag>` table, (`check_doc_field_006`):

```
1   2  3  4   5  6  7
!!-!!!-!-!!!-!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
XX       000     1 acdefgijkmoprst

XX 000 a 001     1 ^abcdefghijklmop|
XX 000 a 002     1 ^abcdefghijklmop|
XX 000 a 003     1 ^abcdefghijklmop|
XX 000 a 004     1 ^abcdefghijklmop|
XX 000 a 001-004 3 check_val_left_just
3 check_val_alpha_order
XX 000 a 005     1 ^abcdefgj|
2 uv
XX 000 a 006     1 ^abcdfrs|
2 ghiz
XX 000 a 007     1 ^abcdefgijklmnopqrstvwz|
2 34hxy
XX 000 e 005-006 1 ^^,aa,ab,ac,ad,ae,af,ag,am,an,ap,au,az
1 ba,bb,bc,bd,be,bf,bg,bh,bi,bj,bo,br,bs
1 bu,bz,ca,cb,cc,ce,cp,cu,cz,da,db,dc,dd
1 de,df,dg,dh,dl,zz,||
```

Key to the check_doc_field_ tables:

Note that for columns that contain positions of the field a zero is added to the left of the position.

Column 1 - Record Format
Enter a specific record format, or use XX as a wildcard to indicate that the values for the position(s) of the field are appropriate for any format. Refer to for more information on record formats.

Column 2 - Match Offset
If needed, enter the field position used as a matching point for the character specified in column 3 (Start position). For example, in the section above:

```
XX 000 a 001     1 ^abcdefghijklmop|
```

^abcdefghijklmop| are valid values for position 01 of the 006 MARC 21 field when position 00 of the field contains value "a".

Column 3 - Match Character
If needed, enter the field character (value) of the position given in the previous column. For example, in the section above:

```
XX 000 a 001     1 ^abcdefghijklmop|
```

^abcdefghijklmop| are valid values for position 01 of the 006 MARC 21 field when position 00 of the field contains value "a".

Column 4 - (Start) Position
Start of the position range to check. For example, in the section above:

```
XX       000     1 acdefgijkmoprst
```

acdefgijkmoprst are the valid values for position 00 of the 006 MARC 21 field.

Column 5 - End Position
If needed, end of the position range to check. For example, in the section above:

```
XX 000 e 005-006 1 ^^,aa,ab,ac,ad,ae,af,ag,am,an,ap,au,az
1 ba,bb,bc,bd,be,bf,bg,bh,bi,bj,bo,br
1 bs,bu,bz,ca,cb,cc,ce,cp,cu,cz,da,db
1 dc,dd,de,df,dg,dh,dl,zz,||
```

^^,aa,ab,ac,ad,ae,af,ag etc. are the valid values for positions 05 to 06 of the 006 MARC 21 field when position 00 of the field contains value "a".

Column 6 - Check Type
Defines the type of check that should be applied. Values are 1, 2 and 3:

1 = Check for valid values.

2 = Check for obsolete values.

3 = Run an external check program.

Column 7 - Check Values
The check values depend on the check type defined in column 6.

If the check type is 1, then this column contains the list of valid values that are valid for the position range. If the value being checked is present on the list, then no error message is displayed.

Values are separated by commas. If the position range is only one character wide, the commas can be omitted. For example, in the section above:

```
XX       000     1 acdefgijkmoprst
```

acdefgijkmoprst are the valid values for position 00 of the 006 MARC 21 field.

If the check type is 2, then this column contains a list of values that can be present but are obsolete. If the value being checked is present on the list, an error message will be displayed informing the cataloger that the value is obsolete.

Values are separated by commas. If the position range is only one character wide, the commas can be omitted. For example, in the section above:

```
XX 000 a 006     1 ^abcdfrs|
                 2 ghiz
```

^abcdfrs| are valid values for position 06 of the 006 MARC 21 field when position 00 of the field contains value "a". ghiz are obsolete values for the same position.

Note that for values of type 1 and 2, the blank should be indicated using the DOC-BLANK-CHARACTER variable in the tab100 table of the library's tab directory.

If the check type is 3, then this column contains the name of the external check routine that must be performed for the position range. For example, in the section above:

```
XX 000 a 001-004 3 check_val_left_just
```

In this instance, for positions 01 to 04 of the 006 MARC 21 field, when the value of position 00 is "a", the system performs the check_val_left_just checking routine. This program verifies that the values in the position range are left-justified.

Following are ALEPH's external check routines for fixed-length field validation tables:

check_val_left_just
Verifies that the values in the position range are left-aligned.

check_val_alpha_order
Verifies that the values in the position range are in alphabetical order, ignoring spaces.

check_val_run_time
Verifies that the three characters specified constitute a valid running time (that is, 000-999, ---, and nnn).

check_fixed_field_length
Verifies that the field is as long as the start and stop offsets would indicate.

check_val_red_ratio
Verifies positions 06-08 of the 007 MARC 21 field for microforms (reduction ratio).

The specific reduction ratio of the microform, recorded as three digits. The number is right-justified and each unused position contains a zero. A hyphen is used for any unknown portion of the reduction ratio.

check_val_date_6
Verifies positions 17-22 of the 007 MARC 21 field for motion pictures (film

inspection date). Six characters that indicate the most recent film inspection date; the date is recorded in the pattern ccyymm (century/year/month). A hyphen is used for any unknown portion of the date. Six fill characters (||||||) are used if no attempt is made to code these character positions.

check_val_blank
Verifies that the position range consists only of blanks (^).

check_val_date_4
For a four-position range, verifies that it forms a valid date (i.e., 1999, 19uu, ||||, and so on. The program does not permit the first position to be "u", nor does it allow "uuuu". This routine is only relevant for the bibliographic (for example, USM01) library - the 008 fixed-length field, "Date 1" and "Date 2" subfields, positions 07-10 and 11-14.

check_val_all_9
Verifies that the position range consists only of the digit 9.

check_val_date_8
For an eight-position range, verifies that it forms a valid date (yyyymmdd). The program allows the last two positions (corresponding to the day) to be blanks.

check_val_numeric
Verifies that the position range consists only of digits. This routine is only relevant for the holdings (for example, USM60) library - the 008 fixed-length field, "Number of copies reported" subfield, positions 17-19 and "Date of report" subfield, positions 26-31

check_val_all_u
Verifies that the position range consists only of the value "u".

check_val_date_4_or_u
Like check_val_date_4, for a four-position range, verifies that it forms a valid date. The difference is that "uuuu" is permitted.

check_val_country
Verifies that the position range forms a valid MARC country code. Valid country codes are defined in the `marc_country_codes` table in the `alephe/tab` directory. This routine is relevant for a bibliographic library (for example, USM01) - the 008 fixed length field, "Publication" field, positions 15-17

check_val_language
Verifies that the position range forms a valid MARC language code. Valid language codes are defined in the `marc_language_codes` table in the `alephe/tab` directory. This routine relates to the following:

Bibliographic library (for example, USM01) - the 008 fixed length field, "Language" field, positions 35-37

Holdings library (for example, USM60) - the 008 fixed length field, "Language" field, positions 22-24.

check_val_bitdepth
Verifies positions 06-08 of the 007 MARC 21 field for ELECTRONIC RESOURCE (image bit depth).

A three-character number specifying the exact bit depth of the scanned image(s) that comprise(s) the computer file, or a three-character alphabetic code which indicates that the exact bit depth cannot be recorded. Since the exact bit depth is useful, coding should not include missing digits represented by hyphens (-). Three fill characters (|||) are used when no attempt has been made to encode this data element.

check_val_heading_use
Verifies that there is at least one "a" in the headings use codes of MARC 21 008 field (positions 14-16). This routine checks if the heading has been marked valid for any use. This routine should only be used for Authority (for example, USM10) libraries.

30 Validation Messages (Table-dependent)

The `check_doc.lng` table of the library's tab directory contains user-defined validation messages that are table-dependent. For example, this table is used to define the error messages that are displayed when performing the check_doc_doc checks. The following is a sample line from the `check_doc_doc` table:

```
OC XX 5002 01 01 245##
```

The message 5002 for the above example must be defined in the `check_doc.lng` table. Following is a sample of the table that contains the message for the line from the `check_doc_doc` table:

```
! 1  2              3
!!!!-!-!!!!
```

```
!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>

0001 L Document number $1 in library $2 points to current document.
0002 L Document has $3 item(s) attached to ADM record $1 in library
$2
0003 L ADM record has $1 item(s) attached.
0004 L Document has $3 order(s) attached to ADM record $1 in library
$2
0005 L ADM record has $1 order(s) attached.
```

Key to the check_doc table:

Column 1 - Error Message Number
Error message assigned by ALEPH.

Column 2 - ALPHA
ALPHA code. Must always be L.

Column 3 - Error Message Text
Message text that is displayed in the Cataloging module when performing the check routine.

32 Cataloging Productivity Report

A cataloging productivity report can be produced by running the Count of New and Updated Catalog Records - by Cataloger (com-02) batch service from the Services menu.

This service measures the productivity of the catalogers within a specific time period. The report includes the number of new records cataloged and the number of updated records by each cataloger. In addition, this service summarizes the total cataloging activity (total number of new records and total number of updated records) for the library between the given time period. Deleted records are also included in the updated records counts.

Alternatively, the Report of New and Updated Catalog Records by Cataloger (com-03) service can be used as well. This service works in a similar way to the com-02 service. Its enhanced algorithm is based on the oracle table, CAT Fields (Z106).

32.1 HOL Records tab of Records Editor

The HOL Records tab, found in the lower pane of the Records Editor, can be set up to include additional information about the HOL record, based on the 801 paragraph of the edit_paragraph.lng table that is in the HOL library's data_tab directory. For example, set the edit_paragraph table with:

801 OWN## D :^

801 852## 3

And the pc_tab_col.lng table with

PC_COM_HOL_SELECT L HOL Information 04 050 01 C01 HOL edit_doc_paragraph 801

33 Column Headings (pc_tab_col.lng and tab_col.dat)

The `pc_tab_col.lng` table of the library's `tab` directory and the `tab_col.dat` table of the ALEPHCOM/LNG/TAB directory define the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (for example, USM01) table pc_tab_col.lng using the ALEPHADM module. Note that some list window columns are not controlled by this table. They are controlled by the tab_col.dat table on the GUI client.

For more information about pc_tab_col.lng, see the ALEPH User Guide - General chapter - Desktop Customization - GUI and Toolbars section.

The following is a list of the Cataloging windows which use the pc_tab_col.lng table for formatting data and their identifiers (Column 1 in pc_tab_col.lng):

Identifier	**Cataloging GUI Windows**
PC_CAT_SCAN	Headings in Library (Search headings options)

* In this GUI table, an optional color/font can be used by the system for color/font differentiation between values of the same column. The alternative font and color are defined in Columns 8 and 9 of `pc_tab_col.lng`.

The following is a list of the Cataloging windows that use the tab_col.dat table for formatting data and their identifiers:

Identifier	**Cataloging GUI Windows**
CAT_SCAN_LIB_LIST	Choose Library (Search headings options)

34 Default Values for Fixed Fields in New Records

The default values for the 008 and the LDR MARC 21 fixed-fields and for the 100 and the LDR UNIMARC fixed-fields are hard-coded. You can manipulate these values by using the tab_tag_text table of the library's tab directory. This table is used to define for each field, according to the record format, the default value for these fields when records are created in the system.

Following is a sample of the `tab_tag_text` table:

```
!1   2                    3
!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
LDR BK ^^^^^nam^a22^^^^^^a^4500
LDR SE ^^^^^nam^a22^^^^^^a^4500

008 BK ^^^^^^s2000^^^^^^^^^^^^^^^^^000^^^eng^d

008 SE ^^^^^^c19009999^^^^r1^^^^^^^^0^^^^0eng^d
```

Key to the tab_tag_text table:

Column 1 - Field Code
Enter the field code of the field for which the default values are being defined (for example, LDR, 008 or 100).

Column 2 - Record Format
Enter a specific record format to indicate the format for which the default values for the field are appropriate.

Column 3 - Default Values
Enter the default values for the field. Use the blank character specified in the DOC-BLANK-CHAR variable of the tab100 table to represent spaces.

35 Importing Records

The conversion mechanism of the Import records option of the Cataloging module is based on two different types of conversions:

- Local conversions (that is, conversions performed at the level of the client)
- Remote conversions (that is, conversions performed by server routines)

For both types of conversions, you must define the conversion program under the `convert.ini file` in the following format:

```
!    1             2                  3                  4
!---------!-------------------!-------------------!--------------------------------
---->

LOCAL      BIP                 0                  Books in print conversion
LOCAL      CDMARC              0                  CDMARC conversion
LOCAL      2709POR             5                  2709 Portugal conversion
LOCAL      2709OCLC            3                  2709 OCLC conversion
LOCAL      CDMARC              CP1251             CDMARC conversion for Cyrillic
LOCAL      2709OCLC            9                  2709 OCLC conversion with atl fix
LOCAL      RLINPASS            5                  RLIN Pass
LOCAL      2709OCLC             CP936               2709 OCLC USMARC conversion for
Chinese(simp)
LOCAL      2709OCLCUNIMARC      CP936               2709 OCLC UNIMARC conversion for
Chinese(simp)
LOCAL      2709OCLC             CPUTF               2709 OCLC USMARC conversion for
UTF
LOCAL      2709OCLCUNIMARC      CPUTF               2709 OCLC UNIMARC conversion for
UTF
REMOTE     SEQ                                     ALEPH Sequential
REMOTE     SEQ300                                  ALEPH300 Sequential
REMOTE      MARC                                              MARC
```

Column 1 - Conversion Program
Local or Remote

Column 2 - Conversion Type
Among the possible types for local program are: 2709POR, 2709OCLC, 2709OCLCUNIMARC, BIP, CDMARC, RLINPASS

Column 3 - Character Conversion Type

Code Page - Not used in REMOTE program. This is the identifier of the conversion that is being performed

Column 4 - Conversion Name
Conversion Name

35.1 Remote Conversions

For remote conversions, the following are the conversion programs that are currently available:

pc_cat_conv_mab_d: conversion from MAB2 Diskettenformat (ekz).

pc_cat_conv_cdmarc: conversion from CDMARC.

pc_cat_conv_aleph_seq: conversion from ALEPH Sequential format.

pc_cat_conv_aleph300_seq: conversion from ALEPH 300 Sequential format. Note that only single records can be loaded/converted by this program.

pc_cat_conv_marc: conversion of MARC records separated either by a new line character, or by a MARC record separator (ASCII - 29 or Hexadecimal - 1D). It can also be used for the conversion of a single MARC record. We recommend that you use this program instead of the '2709 OCLC conversion' local conversion program as that can be problematic in the case of long fields (longer than 2000 characters).

The conversion specifications are defined in the `pc_tab_cat_conv` table, located in the library's tab directory. The following is a sample of the table:

```
    1                2                                3
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!>
MAB        pc_cat_conv_mab_d             850_TO_UTF
MARC       pc_cat_conv_marc
CDMARC     pc_cat_conv_cdmarc
SEQ        pc_cat_conv_aleph_seq
SEQ_8859_1 pc_cat_conv_aleph_seq         8859_1_TO_UTF
SEQ300     pc_cat_conv_aleph300_seq      ALEPH300_TO_UTF
```

Key to the pc_tab_cat_conv Table:

Column 1 - Conversion Routine
This is the identifier of the conversion that is being performed (free-text).

Column 2 - Conversion Program
Enter the conversion program that should be performed for the specific conversion routine defined in column 1.

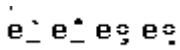
Column 3 - Parameters Certain conversion routines require additional information, such as character conversion routines. This column is used to define additional parameters for conversion programs.

36 Combining Diacritics

The following description is relevant if the combined character functionality is not supported (in alephcom.ini CombinedCharSupported=N). If the combined character functionality is supported (CombinedCharSupported=Y), you can enable the display of hidden characters by selecting the Show hidden characters option in the Edit text menu (Alt+F2).

In order that combining diacritical marks be displayed as clearly as possible in the cataloging draft, they can be displayed in conjunction with a "spacing" character (similar to the way that they are displayed in the Combining Diacritical Marks table in "The Unicode Standard"). The underline (U+005F) has been chosen for diacritics that

are positioned above the character, and the dotted circle (U+25CC) has been chosen for combining diacritics that are positioned below the character.



The spacing characters and the diacritics with which they are used are defined in the Spacer.ini table located in the Alephcom/tab directory. Following is an extract of this file:

```
! 1    2    3
!!!!-!!!!-!!!!
005F 0300 0315
005F 0334 0338
25CC 0316 0333
25CC 0339 033C
```

In the extract above, the underline (U+005F) has been selected as the spacer for the following ranges of Unicode characters:

0300 to 0315

0334 to 0338

In addition, the dotted circle (U+25CC) has been selected as the spacer for the following ranges of Unicode characters:

0316 to 0333

0339 to 033C

37 Record Length Limits

Records in ALEPH are limited to:

5000 subfields.

45000 bytes.

Each field is limited to 2000 bytes.

38 Hidden Fields

Hidden fields are fields that are present in the cataloging record but that are not displayed in the Catalog Editor. In other words, these are fields that cannot be updated directly by the cataloger through the Cataloging module. To define a field as hidden, add the field to the `tab_cat_hidden_fields` table located in the library's tab directory.

Following is a sample of the `tab_cat_hidden_fields` table:

```
! 1
!!!!!
```

650##

39 Record Manager

The Record Manager displays information regarding the record currently being edited in the Catalog Editor (upper pane of the Cataloging tab). The information is displayed in tree structure after opening a record from the server or after saving a new record. Note that the display in the Record Manager is limited to 800 lines.

The system librarian is in charge of defining the following:

The display of the relation between the holdings and the items records. This is determined by the *item_hol_tree_style* variable of the pc_server_defaults file. The following options are available:

1 - Do not display the connection between the items and the linked holdings records
2 - Display the items under the related administrative (ADM) record and also under the linked holdings record
3 - Display the items only under the linked holdings record (in this case the holdings libraries nodes are displayed before administrative - ADM - libraries nodes)

The number of leaves in each node is determined by the *pc_tree_view_max_branch* variable of the pc_server_defaults file. In the following example, nodes stemming from BIB records (in the Record Manager) are limited to 10 leaves and nodes stemming from the Administration/Holdings environment (under the Items, Acquisitions and Circulation_overview trees) are limited to three leaves::

```
setenv pc_tree_view_max_branch 10
```

This variable is limited to 750 nodes.

The way in which the tree is displayed when first opened. This is determined by the *expand_tree_style* variable of the pc_server_defaults file. The following options are available:

1 - Expand only the selected record node.
2 - Expand only record nodes (administrative, bibliographic and holdings)
3 - Expand all existing nodes (administrative, bibliographic, holdings, item, order, and so on.)

Note
The display of the holdings record is also based on the *OWN field*, and the display of the items on the ITEM-SHOW permission in the `user_function.lng` table.

40 Overview Tree

The Overview Tree shows the records in the system that are related to a cataloging record. For example, it displays the holdings records, the administrative record, and the items, subscriptions, orders and loans attached to the selected cataloging record. The information is displayed in tree structure.

Note

The display in the Overview Tree is limited to 800 lines.

The system librarian is in charge of defining the following:

The display of the relation between the holdings and the items records. This is determined by the *item_hol_tree_style* variable of the pc_server_defaults file. The following options are available:

1 - Do not display the connection between the items and the linked holdings records
2 - Display the items under the related administrative (ADM) record and also under the linked holdings record
3 - Display the items only under the linked holdings record (in this case the holdings libraries nodes are displayed before administrative - ADM - libraries nodes)

Note

The display of the holdings record is also based on the OWN field, and the display of the items on the ITEM-SHOW permission in the user_function.lng table.

The number of leaves in each node. This is determined by the pc_tree_view_max_branch and pc_filter_tree_view_max_branch variables of the pc_server_defaults file. In the following sample, nodes of Bibliographic records are limited to 10 leaves and nodes stemming from Administration/Holdings records are limited to three leaves:

```
setenv pc_tree_view_max_branch         10
setenv pc_filter_tree_view_max_branch  3
```

Both variables are limited to 750 nodes.

The way in which the tree is displayed when first opened. This is determined by the *expand_tree_style* variable of the pc_server_defaults file. The following options are available:

1 - Expand only the selected record node.
2 - Expand only record nodes (administrative, bibliographic and holdings)
3 - Expand all existing nodes (administrative, bibliographic, holdings, item, order, and so on.)

You can expand ALL nodes of a record in the Overview Tree, by clicking Expand All Nodes from the right-click menu. This action is the same as the initial state of a record tree when the environment variable expand_tree_style is set to 3.

The moving routines that are performed when records are moved through the Overview Tree. These routines are defined in the tab_move_record table of the bibliographic library's tab directory.

The sort of the items in the tree is dependent on the TREE function in the tab_z30_sort table of the administrative library.

The following is an excerpt of the `tab_move_record` table:

```
! 1         2           3
!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ADM         ADM         move_adm_to_adm
```

```
Z30         ADM         move_z30_to_adm
Z16         ADM         move_z16_to_adm
Z68         ADM         move_z68_to_adm
ITEMS       ADM         move_items_to_adm
COPIES      ADM         move_copies_to_adm
ORDERS      ADM         move_orders_to_adm
HOL         BIB         move_hol_items_to_bib
Z30         HOL         move_z30_to_hol
```

Key to the tab_move_record Table:

Column 1 - Moving From Record Type of record to be moved. The available options are: ADM (administrative) , Z30 (a single item), Z16 (a single subscription), Z68 (a single order), HOL (a single holdings record), ITEMS (all items under the Items node), COPIES (all subscriptions under the Subscriptions node), ORDERS (all orders under the Orders node), HOL-ITEMS, MEX.

Column 2 - Moving To Record Record to which the selected record is being moved. Available options are: ADM (administrative), BIB (bibliographic), HOL (holdings), HOL-ITEMS.

Column 3 - Moving Procedure Moving routines. The following are the available moving programs:

- move_adm_to_adm: moves all instances under an administrative record to another administrative record. This move includes:

Items and item history (Z30 and Z30H)
Holding requests and holdings request history (Z37)
Loans and loan history (Z36)
Photocopy requests (Z38)
Advance booking - time slots (Z320)
Short loans - status (Z321)
Serials claim (Z20)
Maintenance Records (Z328)
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)
Budget transactions (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_adm_to_bib: moves an ADM record from one BIB record to another BIB record. This move includes linked HOL records. It is not performed if an ADM record linked to the target BIB already exists (for example, a new BIB without an ADM).

- **move_bib_to_bib**: moves BIB and AUT records. Linked ADM and/or HOL records will be moved according to move_adm_to_bib and move_hol_to_bib. This procedure works only in MAB. When a holdings record is moved from one bibliographic record to another, all its attached items are moved as well. This program also moves the digital objects attached to the bibliographic record.

- **move_hol_to_bib**: moves a HOL record from one BIB record to another BIB record. It is not performed if an item is linked to the HOL. In this case, the Z30/ADM must be moved first.

This program works only in a multi-HOL, multi-ADM environment and requires the following setup:

```
ADM XXX60 XXX50
```

move_z30_to_adm: moves a selected item to another administrative record. This move includes:

Items and item history (Z30 and Z30H)
Holding requests and holding requests history (Z37)
Loans and loan history (Z36)
Photocopy requests (Z38)
Advance booking - time slots (Z320)
Short loans - status (Z321)
Serials claim (Z20)
Maintenance Records (Z328)

The move is not performed if:

- The item is linked to an incoming or to an outgoing ILL request.
- The item is linked to a holdings 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 administrative record. This move includes:

Subscription information (Z16)
Items and item history (Z30 and Z30H)
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 a holdings 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 and item history (Z30 and Z30H)
Order and subscription log (Z71)
Arrival information (Z78)
Acquisition claims (Z501)
Budget transactions (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 holdings record
- The item is linked to a subscription record

- **move_items_to_adm**: moves all items under the selected items node to another administrative record. The move includes all instances specified under move_z30_to_adm.

- **move_copies_to_adm**: moves all subscriptions under the selected subscriptions node to another administrative record. The move includes all instances specified under move_z16_to_adm.

- **move_orders_to_adm**: moves all orders under the selected orders node to another administrative record. The move includes all instances specified under move_z68_to_adm.

- **move_hol_items_to_bib**: moves a HOL record (together with its items, if any) to another BIB record. Moving of items is performed according to the same guidelines as in move_z30_to_adm; naturally, in this case the restriction preventing the move of an item connected to a HOL record is irrelevant.

- **move_z30_to_hol**: moves an item connected to a HOL record, to another HOL record connected to a different BIB record. Moving of the item is performed according to the same guidelines as in move_z30_to_adm; as in move_hol_items_to_bib, the HOL link restriction is not activated.

Note

If the relevant program for a particular move is not listed in the table, an error message is displayed in the GUI when trying to perform the selected move and the move will not be performed.

The following privileges are related to the functionality of the Overview Tree:

- Global permission to move records: CATALOG (MOVE-TREE-ITEM)
- Permission to move subscription records: CATALOG (MOVE-Z16)
- Permission to move order records: CATALOG (MOVE-Z68)
- Permission to move item records: CATALOG (MOVE-Z30)

No special permission is needed in order to view the Overview Tree.

41 Setting Up a Script for the Correction of Records in Aleph Sequential Format

41.1 Generic Fix Doc Script Specification

The Modify MARC Record File service (p_file_08) modifies records in ALEPH sequential format according to a user-specified processing script. This document describes the format of this script.

The script is in ALEPH table format, with 9 columns. The library can use multiple scripts by creating multiple tables, with different names. One of the parameters of the p_file_08 batch job is the name of the script (table). Blank lines and lines beginning with '!' are ignored.

Col. 1	Iteration	1 digit
Col. 2	Tag	5 characters, tag and indicators. '#' in any position acts as a wildcard)
Col. 3	Format Filter	2 characters, '#' is a wildcard
Col. 4	First Position Filter	1 character
Col. 5	Position Range Start	blank or 3 digits
Col. 6	Position Range End	blank or 3 digits
Col. 7	Occurrence Filter	blank, a 5-digit number, or "FIRST", "LAST", "NOT-F", or "NOT-L")
Col. 8	Operation code	30 characters; the list of valid operation codes is in the *Generic Fix Doc Operations* section
Col. 9	Operation parameters	100 characters max; the list of valid parameters for each operation is listed below

41.2 Script Flow

The script processes every record in the input to p_file_08 as follows:

1. The operations are done in order of Iteration, then in the order within every Iteration. In other words, 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.

2. With the exception of the SORT-FILES operation, each operation is applied only to the fields whose tag matches the pattern in Tag.
   a. If the Format Filter is not blank, the operation is done only if the current format of the record matches (as stored in the FMT field).
   b. If the Occurrence Filter is not blank, only the specified occurrence or occurrences of the field are processed. Note that "NOT-F" means "not first" and "NOT-L" means not last.

c. If the First Position Filter is not blank, the operation is performed only if the first position in the field matches the pattern in Tag.

3. If an operation deletes a field, then that field does not exist for subsequent operations.

4. Certain operations on fixed fields act only on the position range specified by Position Range Start and Position Range End. Note that field positions are counted starting from zero, following the MARC 21 convention but contrary to the UKMARC convention.

41.3 Generic Fix Doc Operations

ADD-FIELD
Adds a field to the record with the coding and contents specified. An occurrence of the new field is added to the record for each existing occurrence of the tag specified in col.2. To add a field once to all records in a file, use a mandatory non-repeatable tag (for example, LDR) in col.2.

Parameters: (comma-separated)
1. field code (5 characters: tag plus indicators)
2. field alpha (1 character)
3. field contents (include subfield codes, for example, $$a)

ADD-FIELD-GENERAL
Similar to the ADD-FIELD operation with the ability to specify the delimiter (rather than using comma). It is not recommended to use a space as a delimiter. The first character in the operation parameter indicates the delimiter.
The following example uses the ^ sign as a delimiter:

```
!   2   3  4  5   6    7                8                          9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!
```

1. subfield code
2. contents

ADD-SUBFIELD-GENERAL
Similar to the ADD-SUBFIELD operation with the ability to specify the delimiter (rather than using comma).
The first character in the operation parameter indicates the delimiter. It is not recommended to use a space as a delimiter.
The following example uses a ^ sign as a delimiter.

```
!
!   2   3  4  5    6    7                8                          9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!
```

Parameters: (comma- or space-separated)
1. value to match; '#' acts as a wildcard (1 character)
2. value to set indicator to upon a match (1 character)

CHANGE-SECOND-IND-MATCH
Changes the value of the second indicator of a variable field. It is similar to CHANGE-FIRST-IND except that it has a third parameter, which is a string to match upon. If there is such a string in the tag's content, then the indicator is changed as specified.

Parameters: (comma- or space-separated)
1. value to match; '#' acts as a wildcard (1 character)
2. value to set indicator to upon a match (1 character)
3. string to match upon. It can include subfield delimiters ($$) and wildcards (#).

CHANGE-SECOND-IND-MATCH-GENERAL
Similar to the CHANGE-FIRST-IND-MATCH operation with the ability to specify the delimiter (rather than using comma or space).
The first character in the operation parameter indicates the delimiter. It is not recommended to use a space as a delimiter.
The following example uses a ^ sign as a delimiter.

```
!   2   3  4  5   6    7                    8                              9
!-!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
```

If “N” and the value of the field position is not in the list, the record is also rejected. In all other cases, the record is accepted. (1 character)

List of values of the field position that determine whether or not to accept the record

COND-LOAD-VAL-FIELD
Determines whether to continue processing the record or reject it, based on the presence or absence of a particular field tag

Parameters:
Condition type; can be “Y” or “N”:

COND-LOAD-VAL-MATCH
Determines whether to continue processing the record or reject it, based on the presence or absence of a particular field tag + subfield + content string

Parameters: (comma-separated)
1. Condition type; can be “Y” or “N”:
2. Text string in the field that determines whether or not to accept the record

COPY-FIELD
Copies the entire contents of each matching field to another field. Any attempt to copy to the same tag, or to a tag that matches the pattern in col.2 is ignored in order to prevent an infinite loop. (If it is necessary to duplicate a field, tag and all, first COPY-FIELD to a temporary tag, then CHANGE-FIELD to the desired tag.)

Parameters: (comma-separated)
field code of new field (5 characters: tag plus indicators)
field alpha (1 character) (defaults to L)

COPY-SYSTEM-NUMBER
Copies the entire contents of a fixed length control field to new variable field and subfield, optionally adding a prefix.

Parameters: (comma-separated)
field code of new field (5 characters: tag plus indicators)
field alpha (1 character)
new subfield (1 character)
optional prefix to add to the contents of the fixed length control field after they are copied to the new variable field

DELETE-FIELD
Unconditionally deletes a fixed or variable field.

Parameters: *none*

DELETE-FIELD-COND
Deletes a variable field if it contains the specified string. Matching is exact and case-sensitive.

Parameters: (comma-separated)
condition type; can be "Y" or "N":
If "Y" and the match string is present in the field, the field is deleted.
If "N" and the match string is not present in the field, the field is also deleted. In all other cases, the field is retained.
match string

DELETE-FIELD-COND-GENERAL
Similar to the DELETE-FIELD-COND operation with the ability to specify the delimiter (rather than using a comma).
The first character in the operation parameter indicates the delimiter. It is not recommended to use a space as a delimiter.
The following example uses the ^ sign as a delimiter.

```
!   2   3  4  5   6    7                8                   9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!
```

Parameters: (comma-separated)
subfield in which to insert the hyphen (1 character)
position within subfield at which to insert the hyphen (3 digits, leading zeroes required)

FIXED-CHANGE-VAL
Changes the value of the specified range of positions (cols.5-6) in a fixed field if the current value matches a pattern.

Parameters: (comma-separated)
pattern; '#' is a wildcard. Note that the pattern must be exactly as long as the specified position range.
replacement values. Must be exactly as long as the specified position range.

FIXED-CHANGE-VAL-RANGE
Replaces every occurrence of a character found anywhere in the specified range of positions (cols.5-6) in a fixed field with another character.

Parameters: (comma-separated)
character to match; '#' is a wildcard
replacement character (use ^ for blank, | for fill character)

FIXED-FIELD-EXTEND
Extends a fixed field if it is at least the specified minimum length but less than the maximum, by appending the specified character to bring it up to the maximum length.

Parameters: (comma-separated)
minimum length of field required for it to be extended (3 digits, leading zeroes required)
length to extend field to (3 digits, leading zeroes required)
character to pad field with (1 character, use ^ for blank, | for fill character)

FIXED-RANGE-OP
Performs the specified operation on a position range (cols.5-6) of a fixed field.

Parameters: 1. field operation. Must be either LOWER or LJ.
LOWER: changes all characters in the range to lowercase
LJ: left-justifies the non-blank values in the range

REPLACE-STRING
Replaces all occurrences of the specified string with another within a variable field. The strings can include subfield codes.

Parameters: (comma-separated)
pattern to match; '#' acts as a positional wildcard; otherwise, matching is exact and case-sensitive.
replacement string; may be the empty string

REPLACE-STRING-GENERAL
Replaces all occurrences of the specified string with another, using a pre-defined delimiter between the source and target strings. The strings can include subfield codes.
The delimiter between the source and target string is set in the position 1 of the operation parameter. From position 2 and onward, set the source and target strings separated by the delimiter character set in position 1.

Parameters: first position in the operation parameter indicates the delimiter. From position 2 and onward, set the source and target strings separated by the delimiter character set in position 1.
pattern to match; '#' acts as a positional wildcard; otherwise, matching is exact and case-sensitive.
replacement string; may be the empty string

SORT-FIELDS
Sorts the fields in the record by tag number, based on the usual sort order for the ALEPH library in which this script is being run. Note that this operation is the only one that does not use tag parameters in col.2. It is recommended to run this as the 9th and last iteration in each script.

Parameters: *none*

STOP-SCRIPT
Stops the script from running. All other operations scheduled after this command are not performed.

Parameters: 1. subfield code and subfield contents for the condition (for example, $$a = *PUB*); can be left blank (see Appendix, entries 11 and 12).

41.4 Generic Fix Doc (p_file_08) Script Examples

1. Create tags 035 and UID from 001 (for Blackwell's MARC with Books files):

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
2 001                     COPY-SYSTEM-NUMBER             035  ,L,a,(OrLoB)
2 001                     COPY-SYSTEM-NUMBER             UID  ,L,a,(OrLoB)
3 001                     DELETE-FIELD
```

2. Create tags 035 and UID from non-standard location of system number (LaserQuest):

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 035##                   REPLACE-STRING                 $$q,$$a(LQUEST)
1 035##                   COPY-FIELD                     UID  ,L
```

3. Change tag 533 – Replace the string ***Wash., D.C.*** with ***Washington, D.C.***
The ^ sign (set in the first position of the operation parameters) defines the delimiter between the source text and the target text.

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
1     533##                       REPLACE-STRING-GENERAL             ^Wash.,
D.C.^Washington, D.C.
```

4. Add a field to all records in a file (for example, 987 to indicate Pinyin processing):

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
1 LDR   ##                ADD-FIELD      987  ,L,$$aPINYIN$$bCaQMM$$c20001130
```

```
                                                $$dc$$e1.0$$fHKUST      w2p
                                                programme
```

5. Change 400 to 800, correcting indicators and removing subfield w:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 4002#                    CHANGE-FIRST-IND               2 1
1 400##                    CHANGE-SECOND-IND              #
1 400##                    DELETE-SUBFIELD                w
1 400##                    CHANGE-FIELD                   800
```

6. Add hyphen to all ISSNs in 022 if not present as 5th character of the subfield:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 022##                    EDIT-SUBFIELD-HYPHEN           a,005
1 022##                    EDIT-SUBFIELD-HYPHEN           y,005
1 022##                    EDIT-SUBFIELD-HYPHEN           z,005
```

7. Extending 007 for computer files to new length with fill character:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 007      c               FIXED-FIELD-EXTEND             006,014,|
```

8. Using fixed field value changing operations with and without ranges of positions:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 007      c 001 013       FIXED-CHANGE-VAL-RANGE         -,|
1 007      h 012           FIXED-CHANGE-VAL               b,i
1 008   SE   030 032       FIXED-CHANGE-VAL               ###,^^^
```

9. Using fixed field range operations (for example, left-justify Nature of Cont. after deleting a value):

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 008   BK   024 027       FIXED-CHANGE-VAL-RANGE         y,^
1 008   BK   024 027       FIXED-RANGE-OP                 LJ
```

10. Do not continue processing the record if the 010 field is not present:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 010##                    COND-LOAD-VAL-FIELD            N
```

11. Do not continue processing the record if the 010 field has a subfield $z with the text DONTLOAD:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 010##                    COND-LOAD-VAL-MATCH            Y,$$zDONTLOAD
```

12. Using a STOP-SCRIPT operation to avoid the performance of scheduled operations when a field with specific contents is present:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 OWN                      STOP-SCRIPT                    $$a = *PUB*
1 LDR                      ADD-FIELD                      OWN  ,L,$$aXPUB
```

13. Using a STOP-SCRIPT operation to avoid the performance of scheduled operations when a specific field is present:

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
1 035                      STOP-SCRIPT
```

```
1 LDR                          ADD-FIELD                      OWN  ,L,$$aXPUB
```

42 Client Setup (catalog.ini)

The catalog.ini file defines settings for the Cataloging client. This chapter presents and explains the following sections of the catalog.ini file:

ConvertFile on page 126

DuplicateRecord on page 130

Editor on page 128

ExpandTemplate on page 130

Form on page 128

General on page 132

HolOwnTextDefaults on page 131

Locate on page 131

OffLine on page 131

RecordBar on page 132

RecordTree on page 132

[RfidMedia] on page 133

Scan on page 131

LOW on page 133

42.1 Catalog.ini Settings

Explanations on settings related to the Items functionality are documented in the Items chapter.

42.1.1 [ConvertFile]

```
[ConvertFile]
```

The ConvertFile section is used for the Import Records subfunction of the Cataloging module.

```
Convert1=L,Books in print conversion,LOCAL,BIP,0
```

The "convert" lines list the available conversion programs. In the example above, the "Books in print" string is the conversion name displayed in the Convert Procedure drop-down menu.

The definitions of the "convert" line include the type of conversion and, if needed, specific parameters (for example, character conversion).

The following is the basic structure for the "convert" lines:

```
ConvertN=TextALPHA,Text,ConversionType,ProgramPath,[Parameter1,Parame
ter2,...,ParameterN]
```

Text: Text that is displayed in the Convert Procedure drop-down menu.

TextALPHA: Alpha of the text.

ConversionType: Type of conversion. The conversion mechanism of the Import Records subfunction is based on two different types of conversion: conversions performed at the level of the client and remote conversions (that is, conversions performed by server routines). For conversions performed at the level of the client, this should be set to LOCAL. For remote conversions, this must be set to REMOTE.

ProgramPath: The path of the program to be executed. For remote conversions, this should always be set to REMOTE.

Parameters: Parameters for the conversion program (optional and program-dependent).For remote conversions, Parameter1 must match a conversion routine from the `pc_tab_cat_conv` table (column 1).

When converting external records into ALEPH format and importing them into your system, you can convert different codepages into UTF-8. To support this, you can add a parameter to the convert lines of the Convert section of the catalog.ini file. The parameter is used to define the input codepage that is to be converted into UTF-8. Following is the convert line for CDMARC records in Cyrillic:

```
Convert5=L,CDMARC conversion for Cyrillic,LOCAL,CDMARC,CP1251
```

This parameter is defined by defining CP + the codepage number (for example, CP1251 for Cyrillic).

If no conversion is needed, this parameter can be set to CPUTF.

```
DefaultInputDir=
```

You can use the DefaultInputDir variable to set the default directory that is opened when the user clicks the button at the right side of the Input File field from the Import Records subfunction. If you leave it blank, the default directory is set to the ConvertIn directory under the Catalog directory.

```
DefaultOutputDir=
```

You can use the DefaultOutputDir variable to define the default directory in which the converted files are stored. If you leave it blank, the default directory is set to the ConvertOut directory under the Catalog directory.

For more information on the Importing Records setup, refer to Importing Records on page 108.

42.1.2 [Form]

```
[Form]
FontSizeX=12
FontSizeY=20
```

The FontSizeX and FontSizeY lines are used to define the grid for the fonts of the cataloging forms.

42.1.3 [Editor]

```
TabCompletion=Y
```

If the TabCompletion flag is set to *Y*, then for subfields that have a list of options defined (tag_text.dat), it is possible to type the beginning of the text and press the Tab key so that the system fills in automatically the complete string.

```
AutoSaveTimeout=1
```

The AutoSaveTimeout variable is used to define the interval - in minutes - between autosaves of local records. If the variable is set to *0* (zero), the records are not saved automatically.

```
UseOldSystemNumber=N
```

The UseOldSystemNumber flag is used to define whether the system number of a record that is being duplicated should be kept as the system number of the new copy of the record. This flag should be set to *N*.

```
DisplayTagInfo=Y
```

The DisplayTagInfo flag determines whether or not the catalog name tags are displayed in the catalog draft in addition to the (usually numeric) field tags. If the flag is set to *Y*, the name tags are displayed.

```
HighLightTag=Y
```

The HighLightTag flag determines whether the tag of the field that is currently being edited appears highlighted or not. If the flag is set to *Y*, the tag is highlighted while the field is edited.

```
EditTag=Y
```

The EditTag flag determines whether the code tag can be edited/changed or not. If the flag is set to *N*, the cataloger will not be able to overwrite tags.

```
ExpandNewTag=Y
```

The ExpandNewTag flag determines whether the subfields defined in the marc_exp.dat are displayed when a field is selected from the list of valid fields - available by using the hotkey F5 or by selecting the New field (choose from list) option from the Edit menu -.

```
SortDeleteEmptyFields=Y
```

The SortDeleteEmptyFields determines whether or not empty fields are deleted when the Sort record option is selected from the Edit menu.

```
FontSizeX=10
FontSizeY=17
```

The FontSizeX and FontSizeY variables are used to define the grid for the fonts of the cataloging draft (for example, tags, indicators, contents).

```
BackGroundColor=255,255,255
InfoColor=128,000,000
TagColor=000,000,255
IndColor=000,000,255
SubColor=192,000,000
FieldColor=000,000,000
DeniedFieldColor=128,128,128
DeniedFieldBackGroundColor=255,255,255

SelectForeGroundColor=255,255,255
SelectBackGroundColor=000,000,128
TagHighLightColor=255,255,255
TagHighLightBackGroundColor=128,000,000
FieldColor1=000,000,000
BackGroundColor1=000,255,000
FieldColor2=000,000,000
BackGroundColor2=255,000,000
```

The above variables are used to define the colors of the different elements of the cataloging draft (for example, the color of highlighted tags, the color of the indicators).

```
DeleteTempDocumentsInterval=7
```

The DeleteTempDocumentsInterval variable is used to define the interval (in days) for NEW* records that have not been updated/created to be deleted automatically from the local drive. If the variable is set to *0* (zero), records will not be deleted automatically.

```
ShowUnicodeValue=Y
```

If the ShowUnicodeValue flag is set to *Y*, when placing the mouse pointer over a character in the catalog record - after about two seconds - a ToolTip appears above the character displaying the character's Unicode value identified by the hexadecimal representation of its Unicode number prefixed with a U, for example, U+0041 for "A". If the flag is set to *N*, this ToolTip is not displayed.

```
RemoteRecordUpdate=Y
```

The RemoteRecordUpdate flag determines whether the system displays a message asking whether or not to update the record in the Remote Catalog whenever the record is saved to the server. The default value is N.

RightClickMenu=EditActions

There are two GUI cataloging Editor menus: Edit Text and Edit Actions.
This flag defines the activation hot keys for those menus.
Possible values: EditActions or EditText

- **EditActions** – (default) Right click to activate the Edit Actions menu. Press shift+right click to activate the Edit Text menu.
- **EditText** – Right click to activate the Edit Text menu. Press shift+right click to activate the Edit Actions menu.

42.1.4 [ExpandTemplate]

```
[ExpandTemplate]
BK=..\template\temp_bk.mrc
```

This section can be used to define a default template for a specific record format. In this case, the default template is selected automatically by the system when the cataloger uses the Expand from template option for a record with the defined format. In the above example, the tem_bk.mrc template - located in the CATALOG/TEMPLATE directory - has been defined as the default template for records of BK (book) format. If no default template is defined, a pop-up dialog box is displayed for the user to select the appropriate template.

42.1.5 [DuplicateRecord]

```
[DuplicateRecord]
Library=ALL
```

This variable is used to define the library/libraries options when duplicating a record. Values are:

HOME - The record is duplicated automatically to the Home Library (this is the library to which the user is currently connected).

ALL - A window listing all libraries defined in the CATALOG/PERLIB.INI file is displayed allowing the user to select the library in which he wants the new record to be saved.

<library code > [, <library code>] - To define specific libraries for selection (for example, USM01, USM10, USM30).

42.1.6 [OffLine]

```
[OffLine]
OffLine=N
```

The OffLine flag determines whether or not the Cataloging module will work with a server connection. If the flag is set to *N*, the client connects automatically to the server when opening the module. If the flag is set to *Y*, no connection is launched and the cataloger can continue working in Offline mode. When working in Offline mode, the user has access to data that has already been downloaded to the local PC (for example, help screens, forms, and so on), but he will not be able to perform functions related to the server (for example, checking procedures, database update, and so on).

42.1.7 [Locate]

```
[Locate]
MergeRecord=Q
```

The MergeRecord variable specifies whether the located similar record should be merged automatically with the current record. If the variable is set to *Y*, then the selected similar record is merged automatically with the current record without a message being displayed. If the variable is set to *N*, then the catalog draft of the selected similar record is displayed. If the variable is set to *Q*, a message is displayed asking the user if the records should be merged.

42.1.8 [Scan]

```
[Scan]

IncludeAUTData=Y
```

The IncludeAUTData flag is used to determine whether or not additional authority information from the authority record should be displayed in the headings list of the bibliographic library together with the linked authority record. If the flag is set to 'Y', then the 260 (Complex See Reference - Subject), 664 (Complex See Reference - Name), 666 (General Explanatory Reference - Name), and 680 (Public General Note) fields from the authority record are displayed together with the linked bibliographic heading.

42.1.9 [HolOwnTextDefaults]

```
[HolOwnTextDefaults]
```

The OWN field is a special ALEPH field that can be used in two different ways:

It can be used to control update access to all types of MARC records (BIB, HOL, ADM, AUT). The user is checked for access/update permission according to the contents of the record's OWN field(s).

It can be used in holdings records to define the "owner" of the record, in other words, the sublibrary to which the record belongs.

This section of the catalog.ini file is used to define in which way the OWN field is used.

```
Activate=Y
```

If the Activate flag is set to Y, when holdings records are created the Enter Owner Information window is displayed. This window enables you to define the owner of the holdings record. If the flag is set to N, then this window is not displayed and users can continue using the OWN field to control update access.

```
SubLibrary=
Note=
```

The SubLibrary and Note parameters can be used to save default owner values. These parameters are automatically filled in by the system after clicking the Save Defaults button from the Enter Owner Information window.

42.1.10[General]

```
[General]
HOLItemSupport=N
```

The HOLItemSupport flag determines whether or not the installation supports HOL Items. In standard applications, this flag should be set to N.

```
LOWSupport=N
```

The LOWSupport flag determines whether or not the installation supports Local Owner functionality (Central-Local Cataloging). In standard applications, this flag should be set to N.

42.1.11[RecordBar]

```
[RecordBar]
FgColorDescript=000,000,255
```

The FgColorDescript instance determines the color of the text displayed in the Catalog bar.

42.1.12[RecordTree]

```
[RecordTree]
BkColor=255,255,255
2=370
ShowDetailedInfo=N
```

BkColor=

The BkColor instance determines the color of the Cataloging tab. Note that all other lines under RecordTree should not be modified.

ShowDetailedInfo=N

ShowDetailedInfo determines whether or not (NEWnnn.MRC) will be presented for every opened record in the record tree or detailed information that includes title, author, system number, format and/or year.

When ShowDetailedInfo=N, the default (NEWnnn.MRC) will appear.

When ShowDetailedInfo=Y, a detailed information will be presented in the record tree.

To enable this, set message number 2004 in./aleph/error_eng/pc_cat_c0203.

Here is an example:

```
!!!!-!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
! Tree node text of record - $1=title, $2=author, $3=system number,
$4=FMT, $5=Year
2004 0000 L $1[$2]
```

42.1.13[RfidMedia]

```
[RfidMedia]
ActivateReader=Y
SuccessMessage=Y
```

The [RfidMedia] section is required when library items are RFID-tagged. For more information on using RFID equipment, please refer to the How-to document in the Ex Libris Documentation Center that is relevant to your RFID vendor (for example, for Bibliotheca, you would refer to *How to set up a BiblioChip® interface in ALEPH® 500 - 18.01*).

ActivateReader=Y

This variable determines whether or not the RFID Reader is updated when the relevant GUI actions are triggered.

SuccessMessage=Y

This variable determines whether or not a message indicating success is issued when the RFID Reader update action succeeds.

42.1.14[LOW]

[LOW]
DefaultOutputDir= C:\Temp

Define the output directory for the "List of Local Owners" window.
Default: C:\Temp.

43 Cataloging Tables

43.1 Library Tables

check_doc
The check_doc table lists all the checking programs that are run when the user chooses the **Check Record** option from the Edit menu or clicks the "Check Record" icon.

check_doc_doc
The `check_doc_doc` table defines field occurrences and dependencies between fields.

check_doc_field_xxx
The `check_doc_field_xxx` tables are used to define valid values for fixed-length fields. For example, the `check_doc_field_008` table is used to define valid MARC 21 values for the MARC 21 008 field.

check_doc_line
The `check_doc_line` table is used when performing tag specific validity checks on a field. The program checks:

- Validity of indicators and subfields.
- Repeatablity and non-repeatability of subfields.
- The presence of mandatory subfields.
- Dependencies between subfields.

check_doc_line_contents
The `check_doc_line_contents` table is used to validate the contents of a field (for example, the ISSN).

check_doc.lng
The `check_doc.lng` table provides validation messages for the check doc programs.

check_doc_mandatory
The `check_doc_mandatory` table is used to define that certain check programs activate triggers or are defined as forbidden. Forbidden errors cannot be overridden and the user is unable to save the record.

check_doc_new_acc
This table defines the fields that should be ignored for purposes of the check messages regarding new acc headings. Up to 1000 codes that should be ignored can be defined. # can be used as a wild card.

check_doc_new_acc_aut
This table defines the fields that should be ignored when checking for new ACC headings, combined with a check in the relevant authority library.

check_doc_tag_text
The `check_doc_tag_text` table validates pre-defined texts for fields.

check_doc_unique_index
The `check_doc_unique_index` table is used to define the field that should be ignored when the system checks whether or not a duplicate record is opened in the Direct (Z11) index.

codes.lng
The `codes.lng` table defines the valid tags and aliases for the database.

fix_doc.lng
The `fix_doc.lng` table contains the text that appears next to fix routines when they are run manually from the Edit menu of the module. The table also determines whether the fix routine appears under the Fix Record option or under the Derive New Record option.

formats.lng
The `formats.lng` table defines the record formats codes (2 characters).

marc_country_codes
The `marc_country_codes` table in the `alephe/tab` directory is used to define the list of valid marc country codes. This table is used by the check_val_country that verifies that the position range of a given fixed-length field forms a valid country code (for example positions 15-17 of the 008 MARC 21 field).

marc_exp.dat
The `marc_exp.dat` table is used to define default subfields. The subfields defined are displayed in the following circumstances:

When a field is selected from the list of valid fields.

When the Open form option from the Edit menu is chosen for a field for which no form is available.

marc_language_codes
The `marc_language_codes` table in the `alephe/tab` directory is used to define the list of valid marc language codes. This table is used by the `check_val_language` that verifies that the position range of a given fixed-length field forms a valid language code (for example, positions 35-37 of the 008 MARC 21 field).

permission.dat
The `permission.dat` table defines allowed and denied tags for different catalogers.

scancode.dat
The `scancode.dat` table defines the heading lists that are used when the cataloger chooses one of the Search Headings functions.

tab00.lng
The `tab00.lng` table defines the system index files. There should be one such table for each language defined.

tab01.lng
The `tab01.lng` table contains the tag codes and names of MARC and ALEPH fields.

tab02
The `tab02` table defines text that is used by the `fix_doc_non_filing_ind` program. The program sets the value of a field's non-filing indicator. Fix programs are defined in the `tab_fix` table.

tab04
The tab04 table converts one set of cataloging tags to another. Different conversion routines can be defined and linked to the fix_doc_tab04_(01_99) program. This can be used when importing records from a database with a different cataloging system.

Note

All tags not defined in this table are deleted from the record when activating the fix routine.

tab05.lng
The `tab05.lng` table defines captions for links between records using subfields in the LKR field.

> In the LKR tag, the MARC tag defining the reason for linking two records is registered in subfield 'r'. tab05.lng defines the caption to display in the OPAC before subfields $$n and $$m.

tab11_acc
The `tab11_acc` table is used to assign fields to headings indexes.

tab11_aut
The `tab11_aut` table is used to define 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.

tab11_ind
The `tab11_ind` table is used to assign fields to direct indexes.

tab11_word
The `tab11_word` table is used to assign fields to word indexes.

tab_aut
The `tab_aut` table establishes which headings indexes in the bibliographic database should be subject to authority control. This table also designates per ACC index which authority database should be checked for a match.

tab_bib_aut_match
Defines 6XX tags and AUT Index codes for the Create Additional Subject Heading(s) from Authority (manage-46)batch service.

tab_cat_hidden_fields
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.

tab_fix_notes
Defines texts used for translation in the fix_doc_notes fix routine.

tab_loader
tab_loader is located in the administrative library's /tab/ directory (./xxx50/tab) and is used by the following services:

- Advanced Generic Vendor Records Loader (file-90)
- Load OCLC Records (file-93)
- Load MARCIVE Records (file-99)
- OCLC server

It defines processing regarding the creation of the holding records, items, orders, budget transactions, and load information.

tab_loader_def
tab_loader_def is located in the administrative library's /tab/ directory (./xxx50/tab) and is used by the following services:

- Advanced Generic Vendor Records Loader (file-90)
- Load OCLC Records (file-93)
- Load MARCIVE Records (file-99)
- OCLC server

It includes default values for fields in the (Z30), orders (Z68), and budget transaction records (Z601) that are created using this service.

tab_own
The `tab_own` table assigns the group of OWN values of a cataloging record (BIB, AUT, ADM or HOL) that are allowed for a particular OWN authorization.

tab_fix
Fix routines are standard library-defined procedures that automatically "fix" or make changes to cataloging records. The `tab_fix` table defines three aspects:

- The fix program that defines the type of "change" performed on the cataloging record.
- The fix routine in which the fix program runs.
- If required, additional parameters for the fix program.

tab_locate
The `tab_locate` table defines the locate routine to be used when searching for a similar record in other databases. Multiple lines can be set up for one library, in which case ALL lines are taken with an AND condition between them. The `tab_locate` table must include both the source and the target library.

tab_match
This table is used to specify the match routines performed by the Check Input File Against Database (manage-36) service and by the `check_doc_match` checking routine.

tab_match_acc
The `tab_match_acc` table is a sample table used to define the fields in the records to be checked against the headings index when the `match_doc_acc` program is used in the `tab_match` table.

tab_merge
The `tab_merge` table lists the merge routines which can be used by the fix_doc_merge program to merge or overlay cataloging records. Column 3 of the `tab_fix` table is used to define the merging routine that matches the relevant section in the `tab_merge` table.

tab_merge_overlay
The `tab_merge_overlay` table defines the fields to be retained, when overlaying cataloging records.

tab_move_record
This table is used to define the moving routines that are performed when records are moved through the Overview Tree in the Cataloging module.

tab_pinyin
This table is consulted to determine the fields on which fix_doc_add_pinyin_check_sub9 and fix_doc_add_pinyin_insert_sub9 routines will run. The fix_doc_add_pinyin programs run on the fields defined, if the content is CJK.

tab_publish
This table contains the specifications for extracting ALEPH records for publishing purposes. The table must be located under the tab directory of the library that contains the records to be extracted (in most cases this is the bibliographic and/or the authority libraries).

tab_subfield_punctuation
The `tab_subfield_punctuation` table is used to define subfield punctuation for fields. Punctuation for fields is necessary when the system automatically updates the bibliographic record from a linked authority record.

tab_z103
Colomn 1 of this `tab_z103` table defines which program runs for the building of links between records. There is an option to define special arguments (like SUDOC) in column 2.

tab_z105
The `tab_z105` table defines messaging between libraries. For example, the update of an authority record should cause an update of a z01 (heading) in the bibliographic library.

tag_text.dat
The `tag_text.dat` table defines fixed values for specific subfields.

tagonnew.dat
The `tagonnew.dat` table defines the default fields when a new record is created.

44 Setting Up the LKR Field

You can control the display of the LKR field by making use of the `tab_fix_z103` table located in the bibliographic library's tab directory.

44.1 tab_fix_z103

The following are the available routines:

GUI-FULL - enables a sorted display in the Full+link tab of the Show node in the Search tab.

GUI-PRINT - enables a sorted display of holdings in printouts.

GUI-TREE - enables a sorted display of holdings in the navigation tree.

HOL-LIST - enables a sorted display in the list of Holding Records in the Cataloging module.

LOCAL-NOTE - allows proper base filtering (expand_doc_bib_local_notes and consults tab_fix_z103 in order to display the tags properly).

WEB-FULL - enables a sorted display in the *Full View of Record* window in the Web OPAC.

WEB-SHORT - enables a sorted display in the *Brief View of Record* window in the Web OPAC.

WEB-SET and WEB-Z103 - enables a sorted display in the *Create set of down-linked records* of the web OPAC.

Following is a sample of the `tab_fix_z103` table:

```
! 1                 2                                3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>

GUI-FULL   fix_z103_sort_existing_key
WEB-FULL   fix_z103_sort_existing_key
WEB-SHORT   fix_z103_sort_doc_no
```

Key to the `tab_fix_z103` table:

Column 1 - routine name
See the routines above

Column 2 - Program name
This is the name of the program that will perform a particular display. The available programs are:

fix_z103_filter_base
The program uses `tab_expand_local_notes.conf` to filter by base. It can use an expand routine in column 3, for example:

WEB-FULL fix_z103_filter_base
FILE=tab_fix_z103_local_notes.conf,EXPAND=WEB-FULL

fix_z103_filter_suppress
The program filters linked documents that are suppressed.

fix_z103_sort_852_b
The program enables the sort by 852 in the printed document.

fix_z103_sort_852_b_item_attr
The program acts like fix_z103_sort_852_b but also looks at `tab_attr_sub_library` type "7" (defines which sublibrary will be the first when sorting items list by sorting routine 06 - preferred sublibrary by IP).

fix_z103_sort_base
The program acts like fix_z103_filter_base, but uses the

tab_fix_z103_local_notes.conf file to control the z103 sort order.

fix_z103_sort_by_my_own

This program sorts records according to the value of the OWN field. All records with OWN tags assigned to the user according to tab_own are positioned at the beginning of the record list.

fix_z103_sort_doc_no
The program enables the sort by z103 doc number.

fix_z103_sort_existing_key
The program enables the sort of Z103 according to z103_sort. The z103_sort is depended on the $s subfield of the LKR field.

fix_z103_sort_lkr_doc_no
The program enables the sort by z103 LKR doc number

Column 3 - Program arguments Contain programs additional information, such as table names. This column is used to define additional parameters for the programs.

45 Supporting additional filters in LKR Field

It is possible to use additional filters in the LKR field to filter items using all the levels of numeration and chronology.

The following are the additional LKR subfields:
- $$d - Fourth level of enumeration
- $$e - Fifth level of enumeration
- $$f - Sixth level of enumeration
- $$g - Alternative first level of enumeration
- $$h - Alternative second level of enumeration
- $$j - Second level of chronology
- $$w - Third level of chronology
- $$o - Forth level of chronology
- $$q - Alternative chronology.

To support the additional filters:

1. Make sure that Z103X table is defined in file_list of the Bibliographic and Administrative libraries.
2. If required, create Z103X table using util/o.
3. Set update_z103_lkr_extended program in tab_z103 of the Bibliographic and Administrative libraries.

For example, ./usm01/tab/tab_z103:

```
!                1                               2
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!->
```

```
update_z103_lkr_extended
```

4. Run manage-12 to re-create the links.

46 LKR Updating Upon Item Enumeration and Chronology Modification

You can set Aleph to update the LKR field of a BIB record upon the modification of the related item's enumeration or chronology information.

The LKR field of type ANA is used to link bibliographic records in order to create a hierarchy between BIB records. This structure supports, for example, maintaining the following three levels of linking between BIB records:

- journal title
- issue title
- article title

The items are created for the journal title. The issue title and article title are linked to the journal title's items via the LKR ANA mechanism.

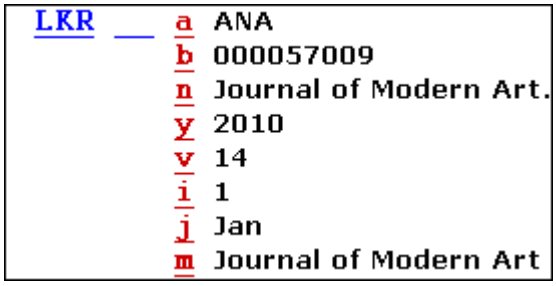
The LKR field is updated when the item enumeration or chronology information is modified using the following sections of the Aleph interface:

- Item Form – Serial Level tab
- Cataloging module – Binding
- Batch service: Update Item Records (manage-62)

For example, there is a bibliographic record with BIB record number 000057009 and Journal Title: Journal of Modern Art. There are 12 items record for this title, one per month (January 2010, February 2010, etc.).

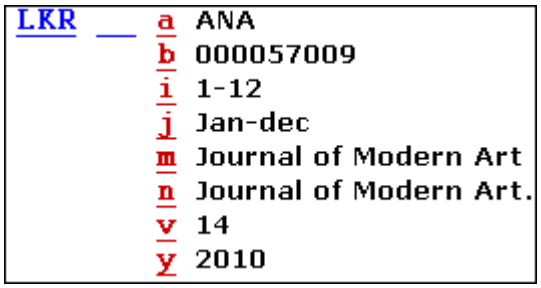
There is a second bibliographic record with BIB Record number 000057010 and Issue Title: Journal of Modern Art – January 2010.

Record 000057010 is linked to record 000057009 in the LKR field:



At the end of the year, the library staff binds all 12 items into a single bound volume, named Vol.14 Jan-Dec 2010. This single bound item replaces the 12 individual items.

After setting up this feature, the system automatically adds information to the LKR field of BIB Record 000057010 that points to the new bound item.



All other Issue Title BIB records (February, March, etc) are also updated with the new LKR field information.

To configure the LKR field updating:

1. Set PERIOIC-INDEX message type to "i" in tab_z105 of the ADM library.
2. In Column 3 (target library), set the BIB library in which to update the LKR field:

For example, ./usm50/tab/tab_z105

```
!      1         2   3     4     5     6     7     8     9    10
!!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!->
PERIODIC-INDEX  i USM01
```

3. In z105_library (for example, USR00), re-activate util E/11 (the message between libraries daemon).

This causes the system to find the bibliographic record that is attached to the LKR field and update the LKR field to reflect the latest enumeration and chronology information set in the item record.

For each bibliographic record that is identified, the following occurs:

- A new LKR field is created, containing filters according to the updated item's information.
- If an identical LKR field already exists, the record is not updated in order to avoid multiple identical LKR fields.
- Previously existing LKR fields are not be deleted from the bibliographic record. There are often several copies of an item and only some of the copies are updated. This means that there are still items linked to the previous LKR fields. The previous LKR fields remain in the record to link to the remaining items that are not updated.
- Previously existing LKR fields remain even if there are no additional items connected to them. This does not cause broken links and is invisible to the end user.

Note that the LKR filters are applied for all LKR subfields only if Z103X (the extended linkage mechanism between documents) is activated. If Z103X is not

activated, the following subfields are not taken into account when linking the bibliographic record to specific items: LKR subfields d, e, f, g, h, w, o, and q.

47 tab100-related Entries in Cataloging

`tab100` is the central configuration table for system-level, server-level and library-level variables. A few lines of the table are shown below:

```
!!!
```

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)

The default value is N.

CREATE-Z00H
Defines whether a deleted BIB record is to be transferred to a history file for statistical purposes.

Possible values are: Y or N.

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.

The default value is N.

CREATE-Z00R
Determines whether or not to create a Z00R record for each Z00 record.

Possible values are: Y or N.

Y = create a Z00R record for each Z00 record. Suitable for BIB, HOL and AUT libraries, but not for an ADM library.

The default value is N.

CREATE-Z106
Determines whether a Z106 record will be created automatically each time a cataloging record is created or updated.

Possible values are: Y or N.

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.

The default value is N.

DOC-BLANK-CHAR
Defines what sign will be used to denote a blank in MARC 21 fixed fields. This should not be confused with the fill character |

Possible values are: ^ or -.

The default value is ^.

EXPAND-ITEM-UPD-TIT

Determines whether or not the BIB's 245$$a is updated when activating the GUI-Cataloging-Expand from Item Barcode tool.

Possible values are: Y or N.
Y = The Expand from Item action replaces the 245$a title field with the item's description (Z30-DESCRIPTION).

N = The Expand from Item action does not update the 245$a title field.

The default value is Y.

FORCE-USE-Z07
Determines how the system behaves when a Z07 record in an ADM or HOL library which is linked to a record in a BIB library is updated.

Possible values are: Y or N.

Y = A Z07 record will be created in the library (for example, ADM or HOL) although the document being updated does NOT belong to it. For example, a Z07 will be created in an ADM library when a record in a BIB library to which it is linked is updated).

The default value is N.

HEADING-SUBFIELDS-ORDER

This flag can be set in ./<bib_library>/tab/tab100.

It defines the order of subfields when replacing the BIB data with BIB/AUTH headings (when using F3 or Ctrl+F3 in GUI-Cataloging and the background processes that update headings).

Possible values:

0 - (default) when replacing the BIB data with BIB/AUTH heading, the order of subfields remains as in the BIB field.

1 - The order of subfields changes to the order of the BIB/AUTH heading.

HOL-008-LNG

The HOL-008-LNG variable is applicable only to the holdings library (for example, USM60) and it is used to determine the default language code for the MARC 21 008 field in holdings records:

If the variable is set to 0, then the language code of the 008 field of the holdings record is set to the defaults specified in the tab_tag_text table.

If the variable is set to 1, the language code of the 008 field of the holdings record is taken from the bibliographic record based on standard system rules (that is, 008, 041, and so on).

The default is 1.

INDEX-ITM-LINK

The ITM link creates a link from one BIB record (for example, record A) to the items that belong to another BIB record (for example record B). If this flag is set to Y then when the items that belong to record B are changed, record A as well as record B will be re-indexed (that is, a Z07 record will be created for both).

For example, if you create logical bases by sublibrary, and the sublibrary of the item that belongs to record B is changed from MAIN to LAW, both records will be re-indexed and will appear under the LAW logical base.

Possible values are: Y or N.

Y = When an item is updated, BIB records linked to the item with an ITM link will be indexed. .

N = When an item is updated, BIB records linked to the item with an ITM link will be not be indexed.

The default value is N.

MARC-EXP-BLANK-CHAR

Determines whether or not to replace the tab100 variable DOC-BLANK-CHAR by a blank in non-fixed fields when exporting records in MARC format.

Y = The character defined by DOC-BLANK-CHAR will be replaced by a blank

N = The character defined by DOC-BLANK-CHAR will remain as is.

MARC-TYPE
Defines the type of the MARC record.

Possible values are: 1, 2, 3 or 4.

1=USMARC, 2=UNIMARC, 3=DANMARC, 4=MAB

The default value is 1.

OWN-FILTER
Determines whether or not the display filter is activated.

Possible values are: Y or N.

Y = The display filter based on `tab_own` is activated (only for HOL or BIB). N = The filter is not active.

The default value is N.

UNION-IGNORE-MATCH

If a value defined here is present in the $$a subfield of STA field in records, any related records will not be found as equivalent to other records.

UPD-NON-FILING-IND

This flag can be set in ./alephe/tab/tab100.

It indicates whether to update the value of the non-filing indicator if the character conversion routine adds or removes characters.

Possible values: Y or N

N or blank (default) = Do not update non-filing values.

Y = Update the tag's non-filing indicator to match the non-filing characters, in case there is a change as a result of character conversion. This action is relevant for

MARC21 BIB tags: 222/240/242/243/245/730 and MARC21 AUTH tags: 130/430/530.

Setting up UPD-NON-FILING-IND=Y impacts batch services that load/create/update MARC21 records into Aleph (e.g.manage-18, file-96, print-03, etc.).

It also affects OCLC server and GUI-Catalog: Import Record action (the latter only for remote server convert procedures).

Possible values are: deleted,circ-created

USE-ACC-TEXT
Determines how the system deals with GEN headings and AUT records when copying into a BIB record.

Possible values are: Y or N.

Y = using CTRL+F3/F4 in cataloging the system copies the contents of the chosen GEN heading into the BIB record.

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 record.

The default value is N.

Z01-TAG-SENSITIVE
Determines whether or not the Z01 record is tag-sensitive.

Possible values are: Y or N.

The default value is N.

48 Setup of ADM Libraries

Note that no ADM libraries should be defined for the **Connect to..** command in the ALEPH menu. In other words, this type of library should not be listed in the catalog/per_lib.ini file. These libraries are accessed from the related BIB library. There is no reason to access them directly.

In addition, note that the **Select ADM Library..** command in the ALEPH menu is used to specify the active ADM environment. This means that all ADM services should be listed in the menu-catalog.xml file and there is no need for the menu-catalog-adm.xml. All jobs that run under an ADM environment must include the <admin_library>Y</admin_library> tag, for example, p-item-03.xml.

49 Matching Records

ALEPH contains a number of matching programs which allow cataloging librarians to match cataloging records according to the matching routines defined in the `tab_match` table located in the library's `tab` directory.

Key to the `tab_match` table:

Column 1 - Match Code
This is the unique match routine code. Each routine performs a particular type of matching operation.

Column 2 - Matching Program
The following are the available match programs:

match_doc_uid: The matching is based on a direct index (Z11). The parameters column (Column 3) must contain either the index name (Column 5 in tab11_ind) or the tag code (Column 1 in tab11_ind). For example, if tab11_ind is defined as follows for the ISBN direct index:

```
1    2    3     4        5             6            7 8
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!-!-!
020                      ISBN  az
```

The parameters for a match based on the ISBN can be defined as follows:

```
XXX   match_doc_uid                 I-ISBN
```

or

```
XXX   match_doc_uid                 T-020
```

Use either I-<index code> or T-<tag code> When using T-<tag code>, there must be an exact match. If tab11_ind col.1 has 020##, this table must have T-020## as well.

match_doc_uid_2: Matching is based on a direct index (Z11). The parameters column (column 3) must contain the index name and the tag code as a unique value. This only works 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:

```
1    2    3     4        5             6            7 8
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!-!-!
035                      035
```

then, the parameters for a match based on the 035 can be defined as follows:

```
XXX   match_doc_uid_2               035
```

match_doc_acc: Matching 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 to be checked against the headings index.

match_doc_script: Uses a table containing a special script for matching records. The table name is defined in Column 3.

match_doc_gen: Contains three 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.

  You can define a specific subfield such as SUBFIELD=a or define matching on any subfield one by one with SUBFIELD=EACH. You can also define matching on any subfield "x" one by one with SUBFIELD=EACHx.

- CODE index name defines the code of the index that is searched in order to find a database record. TRUNCATION=Y can be added to the ACC match type. If this argument is present, the match is performed using a truncated search, that is, the incoming record's field is considered a match if it is contained within the heading.

Column 3 - Program Arguments
For match_doc_uid, this column contains the index code or the tag code used for the direct match.

For match_doc_acc this column contains the table name of the table that contains the tags to be checked against the headings index. Here is a section from the table

```
1          2                      3
F96   match_doc_uid            I-ISBN
RLIN  match_doc_uid            T-020
OCLC  match_doc_script         tab_match_script_oclc
OCLC2 match_doc_uid_2          035
MRCV  match_doc_uid            T-909##
CAT   match_doc_uid            I-ISBN
CAT   match_doc_acc            tab_match_acc
MATCH match_doc_script         tab_match_script.tst
F99   match_doc_gen            TYPE=IND,TAG=909,SUBFIELD=a,CODE=909
```

When match_doc_script is used, a corresponding table must be defined. In the case of the OCLC match code in our example, the system uses values in the `tab_match_script_oclc` table:

```
!1     2                   3      4                5
!!-!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!>
01 match_doc_gen        1    goto 03    TYPE=IND,TAG=035##, CODE=035
01                      0+   goto 02

02 match_doc_gen        20-  goto 03  TYPE=ACC,TAG=245##,
SUBFIELD=abdefgknp,CO
DE=TIT,TRUNCATION=Y
02                      20+  stop
```

Key to the `tab_match_script_oclc` table:

This table contains five columns:

Column 1 - The match set identifier

Column 2 - The name of the match program.

In this example there are three match programs. More programs are available:

match_doc_gen: contains three sections for 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 the database record. TRUNCATION=Y can be added to the ACC match type. If this argument is present, the match is performed using a truncated search, that is, the incoming record's field is considered a match if it is contained within the heading.

match_doc_filter_hvd uses the program arguments SE-TABLE-NAME= and MO-TABLE-NAME=. The matching procedure uses 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.

match_doc_filter_by_weights checks additional parameters existing in `tab_weights`, It can accept two types of Program arguments:

1) TABLE-NAME=<table name>

The following is an example taken from the `tab_match_script` table:

```
!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!>
06  match_doc_filter_by_weights       1       merge           TABLE-
NAME=tab_weights
```

2) <table name> - without the “TABLE-NAME=” words

The following is an example taken from the `tab_match_script` table:

```
!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!>
06 match_doc_filter_by_weights    1    merge      tab_weights
```

Column 3 - Number of Matched Records. Refers to the number of records in the database that match the incoming record. You can specify an exact number, an upper number limit (nn-) or a lower number limit (nn+). 0+ indicates at least one match; 0 indicates no match.

Column 4 - Action. Indicates the action to be taken where the condition of number of matched records is true. Supported actions are: skip (to skip to the next match set);

stop (to stop script execution); goto <xx> (to jump forwards/backwards to a different match set <xxx>; <any text> acts in the same manner as skip. The table above uses the goto and the stop actions.

Column 5 - Arguments. Lists the match program arguments. For the 01 match set, the program arguments are: TYPE, TAG and CODE. For the 02 match set, the program uses the TYPE, TAG + SUBFIELD, CODE and TRUNCATION arguments.

50 Setting Up Services

50.1 Retrieve Catalog Records (ret-01)

You can enlarge the conditions list from two fields to three fields and from one subfield to two subfields within each field.

The third tag field and the second subfield for each tag field are hidden. In order to activate these two options, delete the `<hidden>` XML tag.

For example, the following argument is hidden:

```
<control>
        <hidden>
                <argname>F18</argname>
                <label>Subfield</label>
                <size>1</size>
        </hidden>
</control>
```

To activate it, remove the following two XML tags: `<hidden>` and `</hidden>`.

51 CJK Unicode Characters

There is an option in the Cataloging Record Editor to display and Edit CJK Extension A and Extension B characters. This option includes 4-character Unicode values (U+0000-U+FFFF) and 5-character Unicode values (U+10000-U+FFFFF).

CJK Extension A and Extension B characters can be displayed and entered into cataloging records using Unicode mode (F11):

For Extension B: enter '+' and then enter the 5-character Unicode value (for example, +20000).

For Extension A: act as usual. Enter the 4-character Unicode value (for example, 004C).

Note that CJK Extension A and Extension B characters can be used also in “Open Form” functionality.

ALEPH Keyboard and IME methods work with Extension A values only.

To see SuperCJK Extension A and Extension B characters, install the appropriate fonts (for example: zyksun).

The following lines should be added to ./Alephcom/Tab/Font.ini:

```
EditorField      20000 30000 zyksun   Y N N 16 DEFAULT_CHARSET
ListBox##        20000 30000 zyksun   Y N N 16 DEFAULT_CHARSET
UnicodeEdit      20000 30000 zyksun   N N N 16 DEFAULT_CHARSET
```

52 Publishing

ALEPH publishing is a mechanism which allows sites to extract records from the ALEPH catalog for distributing purposes. (for example, for publishing to search engines and search tools such as Google and Primo).

The extract process has two different flows: initial and ongoing. The initial extract usually includes all records in the catalog, while the ongoing extract mainly deals with new and updated records.

Both publishing processes place the documents into the data repository which is a directory that is locally defined. The data repository consists of Z00p records.

Note that the extract process can be performed on the whole database or on a specific logical base. In addition, extracted records can be modified to include information added by standard ALEPH procedures such as FIX and EXPAND.

52.1 Initial Extract Process

The initial extract process is performed by running the Initial Publishing Process (publish-04). This service can be run from the Publishing submenu of the Services menu in the Cataloging module.

The selected range of records for the specified set will be extracted. (It is possible to choose ‘ALL’ for all the sets specified by the System Librarian).

Note that the service will not run if there is at least one published record in the given range. If you still want to run the service for the selected records, first use the Delete ALEPH Published Records (publish-05) service in order to delete the existing Z00P records in the selected range.

The extraction (initial and ongoing) is performed according to the tab_publish table located under the tab directory of the library that contains the records to be extracted (for example, XXX01).

Here is an explanation for the table’s columns:

Column 1 – Publishing Set

This column contains the code of the set of records to be extracted. For example, if the database needs to be extracted in two separate formats for two separate publishing platforms (such as Google and Primo) then two separate sets should be defined in the table. Note that the code must be in upper case.

Column 2 – Base

A set can be the entire database or a section of the database as defined by a logical base. This column contains the code of the desired logical base from the tab_base.lng table. If the column is left blank, the entire database will be extracted for the set.

Column 3 – De-duplication (for future use)

This column is currently not in use.

Column 4 – Fix and Expand Code

This column contains the fix and expand code of the routines that should be applied before the record is extracted.

Note: To avoid conflicts in tab_fix, do not use the string FULL as a fix and expand code.

Column 5 – Repository Format

This column determines the format of the records in the repository. The supported formats are:

- MARC_XML
- MAB_XML
- HTML
- OAI_MARC21_XML
- OAI_DC_XML

52.2 Ongoing Extract Process

The ongoing extract process is required in order to reflect changes to the database such as the deletion of records and updates to the bibliographic records/holdings records/item records, etc. The ongoing extract process has two main stages:

- The trigger for the extract
- The export of changed/new records to the repository

The triggering mechanism for the extract is based on the ALEPH indexing trigger mechanism. In ALEPH a Z07 record is created for each new or modified record. For the ongoing extract process, when a Z07 record is created for a bibliographic record, the system creates a Z07P record. The Z07P is the trigger for the ongoing extract process.

Note that the creation of z07p is depended on whether tab_publish exists in the Bibliographic and/or Authority library. If the table doesn't exist, no z07p records will be created.

Z07 records are created for bibliographic records in various cases such as changes to the related holdings records, authority records, items, etc. This ensures that bibliographic records are indexed not only according to their own data but also according to associated data. Since the Z07P is based on the Z07, this guarantees that the extracted records, which might contain information derived from FIX and EXPAND procedures, are correctly updated.

Z07p records are also created when an item is loaned, returned, or when a hold request is placed on it.

The timing of the creation of the trigger record (Z07P) differs depending on whether or not the publishing set is created based on a logical base. If the publishing set is not base-dependent, the Z07P is created immediately after the creation of the Z07 record (before it is processed by the UE_01 indexing daemon). If the publishing set is base-sensitive, the record must be indexed before it is extracted. In this case, the Z07P record is created only after the Z07 record has been processed by the UE_01 daemon. The reason for this difference is that in sites where the publishing sets are not base-sensitive, there is no reason to wait for the indexing of the records in order to start the ongoing extract process.

Note that the timing of the creation of the Z07P records explained above is not dependent on the specific publishing set but depends on whether there is at least one entry in the tab_publish table that is base dependent. In other words, if there are four publishing sets defined in the table but only one is base sensitive, then in all cases the Z07P record will be created after the processing of the UE_01 daemon.

The handling of the changed repository records (z00p records) is performed by the Ongoing Publishing utility ue_21. This utility compares the record for which the Z07P was created with the z00p record in the repository. If the records differ (after EXPAND and FIX), the record is handled. When the service finishes processing the triggered documents, the Z07P records are deleted.

In order to prevent unnecessary update because of date change and update of a non important field, a line such as the following can be added to relevant menu in tab_fix:

```
! 1                  2                                  3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
LAM   fix_doc_do_file_08             del_005
```

A file like del_005 should be located in the library's import directory under the tab directory. It can include fields that will always be updated in the bibliographic record such as field 005. These fields will be deleted and the z00p record will not be updated.

Here is an example for the syntax:

```
!   2   3  4  5   6    7                 8                              9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!
```

The Ongoing Publishing utility, ue_21 should run on a regular basis in order to ensure that the repository is up to date. Ue_22 stops ue_21. The performance of ue_21 can be improved by setting the aleph_start/ prof_library variable: `num_ue_21_processes`. This variable enables you to divide the running of the job into several processes. The variable can be set in aleph_start or in the prof_library file of the publishing library. Setting the variable in $alephe_root/aleph_start or aleph_start.private affects all of the publishing libraries. Setting the variable in $data_root/prof_library of the publishing library affects only this library.

If ue_21 tries to upload an invalid xml (containing bibliographic information), a file will be written under $data_scratch directory with the following name convention: util_e_21.xml_err.<YYYYMMDD>.<HHMMSSmm>. The file contains the document number and the library of the document which was not updated due to the invalid xml. The library should look every couple of days for these files and handle them.

The changes triggered by z07p update the z00p records. P-publish-06 service can take the updated z00p records based on dates and record numbers and create a tar file for them. This file can be later transferred to different publishing platforms.

Note that if a change is made to a base or to a base definition in tab_base.lng and this base exist in tab_publish, p-piblish-05 and p-publish-04 should run to create initial load again. Ue_21 should be restarted.

52.3 Name Spacing in Publishing

The BIB library table, ./xxx01/tab/tab_md_ns_info, enables usage of additional or modified name spaces for the Aleph publishing platform.
This optional table can be used to define namespace information for provided formats.

Column 1 – the publishing set code of the set of records to be extracted
Column 2 – defines the namespace information for the publishing set

Example:

```
!        1                      2
!!!!!!!!!!!!!!!!!!!--!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
TEST1                 xmlns="http://www.loc.gov/MARCxx/slim"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.loc.gov/MARC21/slim
http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd"
```

53 Upload BIB and Holding Information from Aleph to KERIS

Aleph uploads BIB documents and holdings from the local Aleph BIB library to KERIS using the Z39.50 protocol.

KERIS is the Union Catalog of the Korean Academic Libraries. The KERIS Server conforms to ANSI/NISO Z39.50-2003(reversion of Z39.50-1995).

KERIS Databases:

- UBIB – BIB documents including local holdings
- UHOL – Holdings linked to UBIB titles

Aleph supports the following update services:

- INSert for UBIB and UHOL
- MODify for UBIB

Aleph uploads to KERIS in two modes:

- Online Upload ("One by one" manual uploading via GUI-Cataloging module) – The user is able to upload new BIB documents, altered BIB documents and BIB documents that were deleted.
  This is available for both BIB documents that were downloaded from KERIS and new cataloged documents.
  The user is able to upload changes in the BIB document's holdings as well.
  This mode can be applied by using the "Remote" menu in the Cataloging module.

- Batch uploading – The user is able to upload group of BIB documents to KERIS using the batch service: Upload Remote Records via Z39.50 (print-30).

Before uploading BIB documents to KERIS, the documents are automatically enriched with additional information using the standard fix and expand mechanism.

53.1 Tables Set-Up Configuration

53.1.1 KERIS Z39.50 Gate Configuration

The KERIS remote database should be configured in Aleph as a Z39.50 gate.

.

The following are instructions for setting KERIS as Z39.50 base. In the sample below, the base code is "KERIS". (You may set the base code to any other 5 characters).

1. Configure the KERIS remote database as an external base of EXT01 libray.

2. Set the following line in ./alephe/tab/tab_base.lng

```
KERIS           KERIS           EXT01           EXT01
```

3. Set a line for KERIS in ./alephe/tab/z39_gate/z39_gate.conf

```
include z39_gate_KERIS.conf
```

4. Set z39.50 gate configuration for KERIS:
   ./alephe/tab/z39_gate/z39_gate_KERIS.conf

5. Set the following lines in KERIS in order to upload to the KERIS z39.50 configuration file: ./alephe/gate/keris.conf file:

```
Z58-ES-DELETE          es_delete_keris
Z58-ES-INSERT          es_insert_keris
Z58-ES-REPLACE         es_replace_keris
```

6. To enable the Remote menu in the Cataloging module, set the BIB library's remote_catalog.dat configuration table. The remote_catalog.dat file is located in the ./<BIB library>/pc_tab/catalog directory.

7. Set the following line in remote_catalog.dat for the KERIS remote base:

```
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!>
KERIS          KERIS          8
```

8. Run util M/7 in bib_library after updating remote_catalog.dat,.

9. To send a repaired document to KERIS (via the "Remote" menu in Cataloging module"), set the EXT01 library's remote_catalog.dat configuration table. The remote_catalog.dat file is located in ./ext01/pc_tab/catalog directory.

10. Set the following line in remote_catalog.dat for the KERIS remote base:

```
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!!>
KERIS          KERIS          9
```

11. Run util M/7 in EXT01 library after updating remote_catalog.dat.

53.1.2 Expand and Fix Routines Setup

Before the document is sent to KERIS, additional information is added to the document which is required to update the KERIS database.

In order to enable document enrichment set tab_fix and tab_expand with the relevant expand/fix routines.

KERSU and KERSD are two special instances that are applied when the KERIS upload is performed. Those instances are set in column 1 of tab_expand and tab_fix. Relevant programs (expand and fix procedures) are defined in column 2 of those tables.

- **KERSU** routine is applied when uploading a record to KERIS (using the Remote function in the GUI Cataloging module or by submitting the batch "Upload remote records via Z39.50").

- **KERSD** routine is applied when uploading a deleted record to KERIS (using the "Remote"" function in GUI Cataloging modules).

Set expand and fix routines in external virtual library (EXT01 in below sample setup), BIB MARC21 library (USM01 in the sample) and BIB KORMARC library (KOR01 in the sample).

Sample of Setup:

Sample Setup in EXT01 (external library)

Fix routines are defined in order to repair an external document and upload it to KERIS. In the following sample, the routine is keris_upload_fix.

tab_fix:

./ext01/tab/tab_fix

```
  1             2                              3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!>
KERSU fix_doc_do_file_08             keris_upload_fix
```

keris_upload_fix:

keris_upload_fix procedure should be set in:

./ext01/tab/import/keris_upload_fix (delete various fields)

```
!   2   3  4  5   6    7               8                 9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!>
1 SID##                   DELETE-FIELD
1 CAT##                   DELETE-FIELD
1 FMT##                   DELETE-FIELD
1 SYS##                   DELETE-FIELD
1 STA##                   DELETE-FIELD
1 LOC##                   DELETE-FIELD
1 OWN##                   DELETE-FIELD
1 KER##                   DELETE-FIELD
```

Sample Setup in USM01 (MARC21 BIB library):

Fix routines should be defined in order to upload updated document or delete document. In the following sample, the routines are keris_upload_fix and keris_delete.

tab_fix:

./usm01/tab/tab_fix

```
  1                      2                     3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!>
KERSU fix_doc_do_file_08             keris_upload_fix
KERSU fix_doc_005
KERSU fix_doc_004_lkr
KERSD fix_doc_do_file_08             keris_delete
KERSD fix_doc_005
KERSD fix_doc_004_lkr
```

keris_upload_fix:

keris_upload_fix procedure should be set in:

./usm01/tab/import/keris_upload_fix (LDR position 10 set to "u" and various fields are deleted):

```
!   2   3  4  5   6    7               8                 9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!>
1 LDR        010 010       FIXED-CHANGE-VAL            #,u
1 SID##                    DELETE-FIELD
1 CAT##                    DELETE-FIELD
1 FMT##                    DELETE-FIELD
1 SYS##                    DELETE-FIELD
1 STA##                    DELETE-FIELD
1 LOC##                    DELETE-FIELD
1 OWN##                    DELETE-FIELD
1 KER##                    DELETE-FIELD
```

keris_delete:

keris_delete procedure should be set in:

./usm01/tab/import/keris_delete (LDR position 5 is set to "d"):

```
!   2   3  4  5   6    7               8                 9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!>
1 LDR        005 005       FIXED-CHANGE-VAL            #,d
```

tab_expand:

Set expand routine to enrich the document with 852 item information. The Aleph sublibrary code should be converted to a code as required by KERIS. In the sample below, the change_sub_library routine converts sublibrary codes ("LAW" and "MED") to KERIS codes (001 and 002 respectively).

./usm01/tab/tab_expand

```
!   1                    2                              3
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!>
KERSU       expand_doc_bib_852_1
KERSU       fix_doc_do_file_08           change_sub_library
```

change_sub_library

change_sub_library procedure should be set in:

./usm01/tab/import/change_sub_library (converts Aleph sublibrary to Keris code):

```
!   2   3  4  5   6    7               8                9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!>

1   852##                                               REPLACE-STRING
$$bLAW,$$b001
1   852##                                               REPLACE-STRING
$$bMED,$$b002
```

Sample Setup in KOR01 (KORMARC BIB LIBRARY):

Fix routines should be defined in order to upload updated document or delete documents. In the following sample, the routines are keris_upload_fix and keris_delete.

tab_fix:

./kor01/tab/tab_fix

```
  1                     2                  3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!>
KERSU fix_doc_do_file_08             keris_upload_fix
KERSU fix_doc_005
KERSU fix_doc_004_lkr
KERSD fix_doc_do_file_08             keris_delete
KERSD fix_doc_005
KERSD fix_doc_004_lkr
```

keris_upload_fix:

keris_upload_fix procedure should be set in:

./kor01/tab/import/keris_upload_fix (LDR position 10 set to "c" and various fields are deleted):

```
!   2   3  4  5   6    7               8                9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!>
1 LDR        010 010       FIXED-CHANGE-VAL            #,c
1 SID##                    DELETE-FIELD
1 CAT##                    DELETE-FIELD
1 FMT##                    DELETE-FIELD
1 SYS##                    DELETE-FIELD
1 STA##                    DELETE-FIELD
1 LOC##                    DELETE-FIELD
1 OWN##                    DELETE-FIELD
1 KER##                    DELETE-FIELD
```

keris_delete:

keris_delete procedure should be set in:

./kor01/tab/import/keris_delete (LDR position 5 is set to "d")::

```
!   2   3  4  5   6    7               8                9
```

```
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!>
1 LDR        005 005       FIXED-CHANGE-VAL            #,d
```

tab_expand:

Set the expand routine to enrich the document with 852 item information. The Aleph sublibrary code should be converted to a code as required by KERIS. In the sample below, change_sub_library routine converts sublibrary codes ("LAW" and "MED") to KERIS codes (001 and 002 respectively).

./kor01/tab/tab_expand

```
!   1                    2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!>
KERSU       expand_doc_bib_852_1
KERSU       fix_doc_do_file_08            change_sub_library
```

change_sub_library:

change_sub_library procedure should be set in:

./kor01/tab/import/change_sub_library (converts Aleph sublibrary to Keris code):

```
!   2   3  4  5   6    7                8                 9
!-!!!!!-!!-!-!!!-!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!>

1   852##                                                 REPLACE-STRING
$$bLAW,$$b001
1   852##                                                 REPLACE-STRING
$$bMED,$$b002
```

53.1.3 Manual Upload Using the Remote Menu of Cataloging Module

The Remote menu of the Cataloging module uploads a single document to KERIS. It supports the following updates:

- Upload new and updated document
- Upload deleted document
- Upload suppressed document
- Send repaired document to KERIS

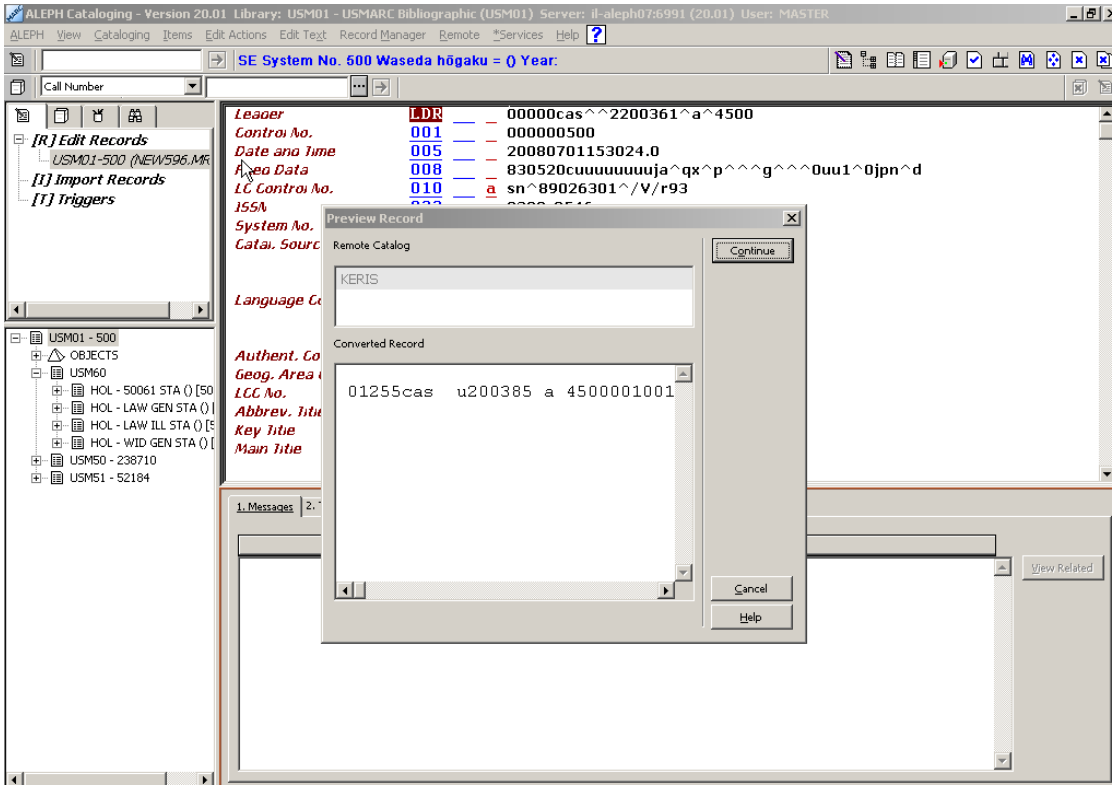
The upload action requires the following staff user privileges: Cataloging Record/Send record to remote server.

53.1.4 Upload New and Updated Document Manually

The users update BIB documents and holdings or create new BIB documents and holdings in the BIB library using the standard workflow supported by Aleph in the cataloging module. The insert and update operations on the BIB document are recorded in the CAT field on the document.

In order to update KERIS union catalog, the user should select the Add Record option in the Remote menu of the cataloging module.

Selecting the **Add Record** option opens the Preview Record dialog which enables the user to view the converted record that is sent to the remote system.



When the **Continue** button is clicked, the BIB document and its holding information is sent to KERIS database using the Z39.50 protocol.

Before sending the BIB document to KERIS, the BIB document is automatically enriched using the standard Aleph fix and expand routines, adding the holdings information and other necessary data.

The job action of the message to KERIS is always INSert and the database target in KERIS is determined using the 035 field on the document:

- BIB documents downloaded from KERIS includes 035 field with subfield $$a starting with "(KERIS)BIB" is uploaded to KERIS UHOL database (example of 035$$a: (KERIS)BIB00001075867).

- New BIB documents that were not downloaded from KERIS and do not includes 035 field with subfield $$a starting with "(KERIS)BIB" are uploaded to KERIS UBIB database.

After the BIB document is sent, the upload date is registered in KER field subfield $$a.

53.1.5 Upload Deleted Document Manually

Deleting the remote BIB document from KERIS must occur before deleting the BIB document from the local BIB library due to the fact that deleting the BIB document from the local library removes most of its data including its 035 field which identifies the source system of the document and the document number in that remote system.

Use the “Delete Record” option in the “Remote” menu in order to delete the BIB document from KERIS.

Select the **Delete Record** option in the Remote menu to send the enriched BIB document to KERIS using the Z39.50 protocol with the value “d” in the position 5 in the LDR field.

The job action of the message to KERIS is INSert and the database target in KERIS is determined using the 035 field on the document:

- BIB documents downloaded from KERIS and includes 035 field with subfield $$a starting with "(KERIS)BIB” is directed to KERIS UHOL database.

- New BIB documents that were not downloaded from KERIS and do not include the 035 field with subfield $$a starting with (KERIS)BIB but were uploaded to KERIS before are directed to the KERIS UBIB database.

The BIB document and its holdings are not deleted from the local BIB library automatically. After completing the delete from the KERIS union catalog, the user needs to delete the BIB document from the local BIB library using the standard delete functionality supported in Aleph in the cataloging module.

To delete the BIB document from the local BIB library, use the **Delete record from server (ctrl + R)** option in the “Edit Text” menu.

53.1.6 Upload Suppressed Document Manually

Adding the field STA with the value $$aSUPPRESSED to a BIB document and saving it to the local BIB library, cause it to be considered suppressed.

Leader	LDR			00656nas^^22002172^^4500
Control No.	001			000000222-4
Date and Time	005			20020418155342.1
Fixed Data	008			770701c19049999ne^|||p||||||||||||dut||
ISSN	022		a	0922-6699
Main Title	245	00	a	Jaarboekje voor geschiedenis en Oudheidkunde van Leiden en Omstreken.
Varying Title	246	13	a	Leids jaarboekje
Status	STA		a	SUPPRESSED
Imprint	260	00	a	Leiden :
			b	<S.N.>,
			c	1904-
Dates of Pub.	362	0	a	jg.1- 1904-
General Note	500		a	Half-title: Leids jaarboekje.
General Note	500		a	Title varies slightly.
General Note	500		a	Issued by Vereeniging Oud-Leiden.
Subject-Geo.Trm	651	0	a	Leiden (Netherlands)
			x	History
			v	Periodicals.
A.E. Corp. Name	710	2	a	Vereeniging Oud-Leiden.

When the user suppresses the BIB document, he should upload the BIB document to KERIS as deleted BIB documents as described is section "Upload Deleted Document Manually"

When the user removes the STA field from the BIB document and saves the BIB document to the local BIB library, he should upload the BIB document to KERIS as described is section "Upload New and Updated Document Manually".

53.1.7 Upload Repaired Document Manually

When a BIB document is downloaded from KERIS, it is saved in a temporary EXT library. The cataloger cannot update and save the document in the EXT library since the EXT library is temporary.

The user can repair the downloaded BIB document in the EXT library via the cataloging module editor and upload it to the KERIS UBIB database using the **Update Record** option in the Remote menu of the cataloging module.

Note that the BIB document data is taken from the cataloging module editor instead of the Aleph database, as in the other remote actions, since the BIB document can not be saved.

The uploaded BIB document from the EXT library is sent to the KERIS UBIB database with a MODify action.

If the user wants to save the downloaded BIB document to Aleph, he should use the standard functionality applied as “Duplicate Record (Ctrl+N)” action under the Cataloging menu in the cataloging module.

53.1.8 Bulk Upload Using Batch Service

In addition to the manual upload to KERIS functionality, Aleph provides an option for bulk uploading to KERIS.

Bulk upload to KERIS functionality is divided into two stages:

- Initial filtering of BIB documents.
- Secondary filtering of BIB documents and uploads to KERIS via batch.

The purpose of the initial filtering is to create an input file of all updated and new records that should be uploaded to KERIS.

The initial filtering of BIB documents can be performed using the "Retrieve Catalog Records (RET-01)" batch process that enables retrieving documents according to various parameters, for example:

- Records that have items that were updated between dates ranges(From-To)
- System record number ranges(From-To)
- The last updated Cataloger
- The last updated Date (From-To)

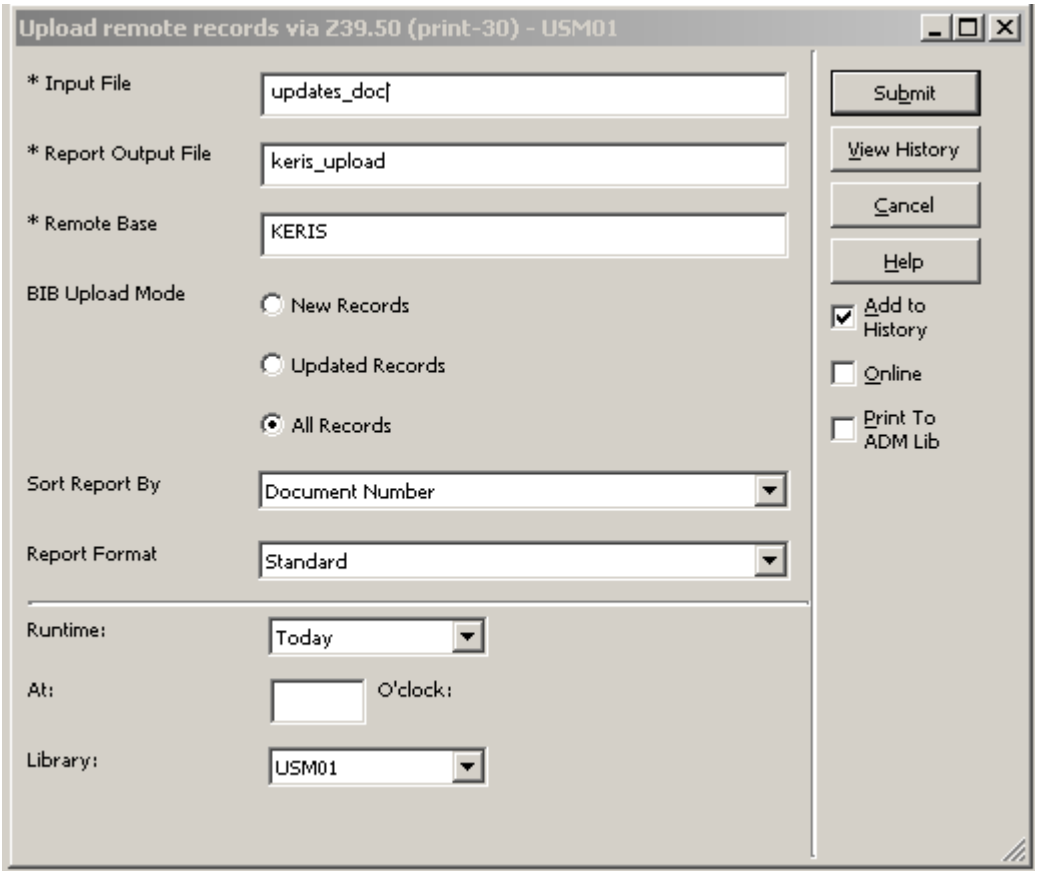
The output BIB documents numbers is written to an output file that is saved in the $alephe_scratch directory and is used as a basis for the secondary filtering.

The Secondary filtering of BIB documents and uploading to KERIS is done using the "Upload remote records via Z39.50 (print-30)" batch process.

The batch interface is accessed via Cataloging module-Services Menu-Retrieve Catalog Records- Upload remote records via Z39.50 (print-30).

The batch can be submitted by authorized staff users only.

Required staff user privilege: Record Services/Upload Remote Records Via Z39.50



The Batch parameters:

Input File: file name in $alephe_scratch directory. This file needs to contain BIB documents numbers that should be uploaded to KERIS. Usually this file is the output file of "Retrieve Catalog Records (RET-01)" batch process.

Report Output File: The name of the file in which you want to save the output report.

Remote Base: Z39.50 base code to which the process sends the BIB document information.

Upload BIB Records: Secondary filter which allows the users to choose to upload:

- **New Records**: Records from the input file that were not uploaded to KERIS before.
- **Updated Records**: Records from the input file that were uploaded to KERIS before and have been updated.
- **All Records**: All the records that are ready for upload (new and updated) from the input file.

Using the CAT fields on the BIB document, it is possible to identify the last update date of the BIB document.

When the BIB document is uploaded to KERIS, subfield $$a in the KER field is updated with the last upload date.

Using these dates, it is possible to identify the BIB documents that are ready for KERIS upload and the BIB documents that are new to KERIS.

Sort By: The parameter by which the output report is sorted.

Report Format: The format of the output report IB Records (template file name: remote-base-upload.xsl).

1.3.1 The Batch Execution

Before sending the BIB documents to KERIS, the documents are enriched using the fix and expand standard functionality. In addition, the BIB documents LDR field are populated in position 10 according to the library documents format:

- Value “c” for KORMARC documents.
- Value “u” for MARC21 documents.

After the BIB document is sent, subfield $$a in the KER field is updated with the upload date, using a fix routine.

53.1.9 The Batch Output Reports

After the batch is submitted, a detailed output report is produced, describing the outcome of each uploaded BIB document. The Upload status can be one of the following:

- Success
- Library problem
- Document problem
- Base problem
- Not uploaded

The report template filename: `remote-base-upload.xsl`.

The template translation filename: `remote-base-upload.trn`.

In addition, print-30 batch service generates statistic information that can be retrieved using the TCO Batch Summary Report (sys-90). The summary includes the following statistics information:

- Number of documents processed
- Number of documents successfully uploaded to KERIS
- Number of documents that failed to upload to KERIS

54 OUF Loader

The OUF Loader (p_ouf_load) batch service loads into Aleph the bibliographic records BIB, AUT, and HOL fetched from an OCLC/PICA system.

The OUF Loader process receives input in Aleph sequential format. BIB, AUT, and HOL records have to be processed in separate files. Retrieving the records from the

OCLC source system via OUF (OCLC Online Update Fetch) and converting the records into Aleph sequential format is done by external programs.

Possible actions (update modes) are add, update, delete, and linked. Both add and update do the same thing - they add as new record if it does not exist in database and updates it if it is found in database. The difference is that for add, a warning log is created if the record is already in the database, and for update a warning log is created if this record is not found. Linked value means the same as add or update but no warning log is created whether the record is new or not and there are more restricted conditions on the record timestamp.

Records identification is based on a OCLC ppn number, put in Marc 001 field and indexed as 001 IND index (Z11). Proper ppn and working unique 001 index are necessary to update the record and for the link to work properly. Before a record is inserted a fix routine can be applied. Before a record is updated a merge routine can be applied. In addition it is possible to apply fix routines before and after the merge. The OUF Loader is intended to be part of a regular batch process for updating the Aleph database with data from the OCLC union catalogue. Therefore, the OUF Loader cannot be run from the Services menu of the Cataloguing module, but only from a command line or other scripts.

54.1 Instructions for Running the OUF Loader

54.1.1 Running from the UNIX Prompt

Enter the following command line:

csh –f $aleph_proc/p_ouf_load <parameter_list>

where <parameter_list> ::= <library>,<library_type>,<input_file>,<reject_file>,
<log_file_ok>,<log_file_wrn>,<log_file_err>,
<update_mode>,
<fix_a>,<fix_u1>,<merge>,<fix_u2>,
<update_db>,
<bib_iln>,<aut_type>,<hol_related_bib>

54.1.2 Parameters Description

<library>
The Aleph library code (example: USM01). Mandatory.

<library_type>
Type of the library. Possible values: “BIB”, “AUT”, “HOL”. Mandatory.

<input_file>
Name of input file with doc records in Aleph sequential format. Mandatory.

<reject_file>

Name of output file for rejected records. Optional.

<log_file_ok>
Name of output file for logging information of records processed without errors or warnings. Mandatory.

<log_file_wrn>
Name of output file for logging information of records processed with warnings. Mandatory.

<log_file_err>
Name of output file for logging information of records processed with errors. Mandatory.

<update_mode>
Update action. Possible values: “A” (add), “U” (update), “D” (delete), “L” (linked). Mandatory.

<fix_a>
Name of fix routine for adding records. The fix routine has to be defined in tab_fix in $data_tab of <library>. Optional.

<fix_u1>
Name of fix routine for updating records. The fix routine has to be defined in tab_fix in $data_tab of <library>. This routine is called before a merge is done. Optional.

<merge>
Name of merge routine. The merge routine has to be defined in tab_merge in $data_tab of <library>. Optional.

<fix_u2>
Name of fix routine for updating records. The fix routine has to be defined in tab_fix in $data_tab of <library>. This routine is called after a merge is done. Optional.

<update_db>
A Y/N-flag indicating if the update in the database should be done. Possible values: “Y”, “N”. Mandatory.

<bib_iln>
Field code of the Internal-Library-Number (ILN). Relevant only when library_type = “BIB”. Optional.

<aut_type>
Type of the authority data. Relevant only when library_type = “AUT”. This parameter is used only for logging information. Optional.

<hol_related_bib>
Library code of the related BIB library when library_type = “HOL”. Mandatory if when library_type = “HOL”.

Examples:

csh –f $aleph_proc/p_ouf_load USM01,BIB,bibin.mrc,bibout.reject,biblog.ok, biblog.wrn,biblog.err,A,OUFA,OUFU1,OUFM,OUFU2,Y,iln,,,

csh –f $aleph_proc/p_ouf_load USM60,HOL,holin.mrc,,hollog.ok, hollog.wrn,hollog.err,U,,,,,Y,,,USM01,

55 Preventing the Automatic Creation of PAR Reciprocal Links Between Records

Two BIB records can be linked based on the LKR field cataloged in one record.

After you set tab_z103 and catalog LKR$$aPAR (parallel link types) in one BIB record, the system automatically creates a reciprocal PAR link in the corresponding linked record.

There is an option to set Aleph to prevent the automatic creation of reciprocal links for PAR type linkage. To configure this, set the `update_z103_lkr_extended` routine in the tab_z103 library table with the parameter `1way-par`.

For example: ./usm01/tab/tab_z103

```
!                1                              2
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!->
update_z103_lkr_extended        1way-par
```

The above setup prevents the creation of the reciprocal link so that the Z103 link is not created for the corresponding record. The link will be one-way – from the current record to the corresponding record – but not vice-versa.

In order to make the link reciprocal, you must manually catalog a LKR in the linked record.

Note that modifying tab_z103 requires re-starting UTIL/E/1 (update doc index daemon); otherwise, the changes in tab_z103 do not take effect.

56 Generating a Locally Assigned Call Number In Bibliographic and Item Records

The fix routine, fix_doc_090_call_no, assigns a local call number to the 090 field of the BIB record using a running sequence counter.

56.1 Creating the Call Number in the BIB Record

The sequence is automatically assigned, depending on the prefix that the user inserts.

To activate the routine, set up bib_lib/data_tab/tab_fix as follows:

```
! 1                  2                                  3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CALL  fix_doc_call_no_090            MODE=PREVIEW
INS2  fix_doc_call_no_090            MODE=UPDATE
```

In the above example, the CALL instance is activated manually. To activate it, you must first define it in bib_lib/pc_tab/catalog/fix_doc.lng

Column 3 of tab_fix contains the parameters of the fix. The MODE parameter is mandatory. It has two options:

- PREVIW – After the sequence number a text is added in parenthesis. The sequence number is not yet stored in the database (Z311). This is to allow for a check of the new prefix before the database is updated.
- UPDATE – The sequence is stored in the database. If the sequence number has the added text, it is removed.

By default, the added text is "(expected)". To change the text, use the parameter "TXT=" with the defined string as the text to be added. Note that if you change the text, you must add the TXT parameter to both modes – UPDATE and PREVIEW.

In order to activate this mechanism, the Z311 Oracle table must be created in the BIB library using util/a/17/1.

The following is an example workflow:

1. The user inserts the prefix TRN-T in subfield 090$$a.
2. The user activates fix_doc_090_call_no.
3. The system automatically assigns 090$$a with the next sequence – for example, TRN-T14(expected).
4. When the BIB record is saved, the appending text "(expected)" is removed and the 090$$a subfield is set with TRN-T14.

If the user enters a prefix ending with numbers (for example, TRN-T66), the system relates to 66 as the sequence number and does not assign a new sequence number.

Note that when activating this fix with a loader, the MODE parameter must be set to UPDATE only. In this case, the TXT parameter is not relevant.

It is possible to copy the call number from the 090$$a subfield of the BIB record to the call number type field of the Item record. For more information, refer to the **Copying the Call Number from the BIB Record to the Item Record** section of the *Aleph 21 System Librarian's Guide - Items*.

56.2 Check Routine for Call Number Prefix

The check_doc_call_no_090 check routine reviews the prefix entered in subfield 090$$a and provides a record check warning when a new prefix is used. The message "New Prefix Alert" is displayed each time a new counter is used. Note that if you defined the TXT parameter in the fix routine, the same text must also be defined in the check.

The check routine is set in ./bib_library/tab/check_doc.

```
!        1                         2                          3
!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!>
CATALOG-INSERT       check_doc_call_no_090

Alert text is set in   ./aleph/error_eng/check_doc
! 1  2              3
!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
1100 L ($1$$a): Call number prefix $2 is a new entry.
1101 L ($1$$a): Call number prefix $2 is a new entry, with manual input of
the sequence=$3 !
1102 L ($1$$a): Call number prefix $2 is a old entry, with manual input of
the sequence=$3 !
1105 L ($1$$a): Error, Call number prefix $2 is an old entry, but maximum
for sequence reached; no automatic sequence handling !
1106 L ($1$$a): Error, Call number prefix $2 to long (l'max=25); no
automatic sequence handling !
1107 L ($1$$a): Error, Manual input of sequence $2 > 99999999; no automatic
sequence handling !
1110 L ($1$$a): Error, unknown check status of the Call number $2, create an
issue !
```

57 Automatic Creation of 6XX Fields

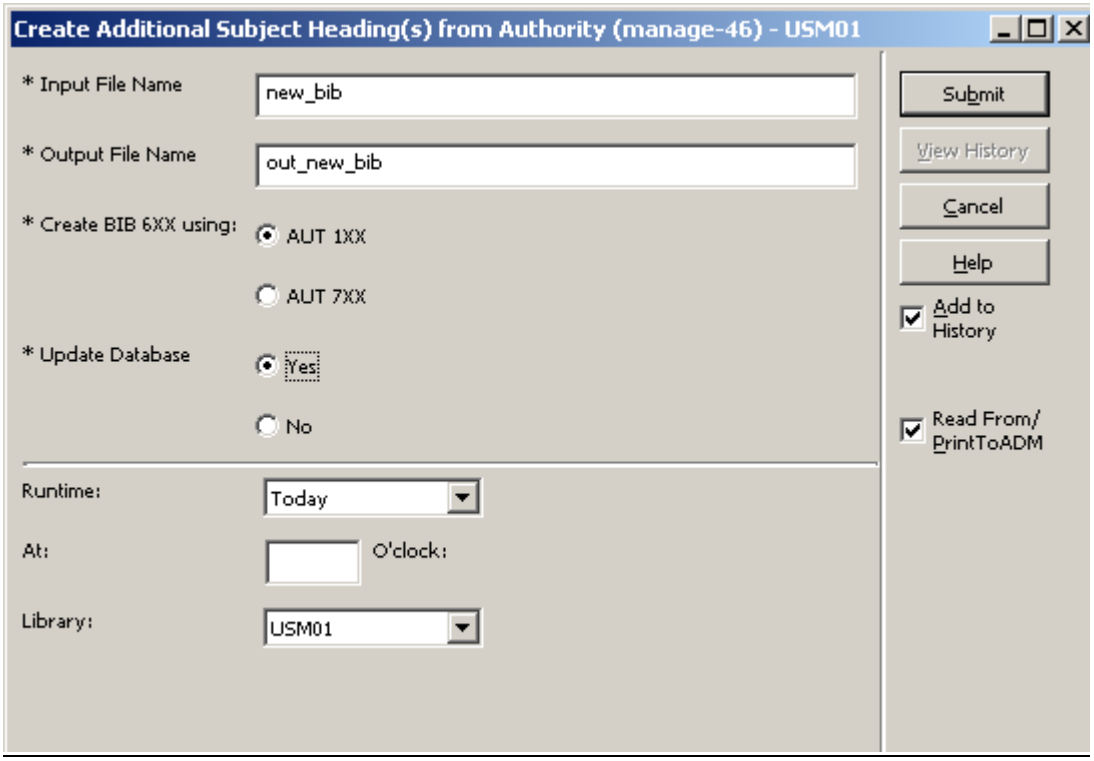
This chapter describes the activity of the Create Additional Subject Heading(s) from Authority (manage-46) cataloging batch service.
The service creates a translated subject heading within the bibliographic record based on authority matching indexes. This can be used, for example, for updating bibliographic records with French RVM subject headings based on matching terms in the authority heading.

The Manage-46 service creates the 6XX bibliographic field by finding a match with the authority records and applying the 1XX/7XX authority field.

The service works in conjunction with the bibliographic library table tab_bib_aut_match that points to the relevant AUT database and index codes to which the match is performed.

57.1 The Batch Service: Create Additional Subject Heading(s) from Authority (manage-46).

The procedure that creates 6XX is submitted by the Create Additional Subject Heading(s) from Authority (manage-46) batch service. The service can be accessed via **Cataloging>Bibliographic Library Services> Catalog Maintenance Procedure**.



Batch parameters:

1. **Input File** – Mandatory. The name of the file that contains the bibliographic document numbers that are handled by the batch procedure. The input file must exist in the $alephe_scratch directory. An input file can be generated by submitting various cataloging batch services, such as Retrieve Catalog Records (ret-01), Cross Files (ret-10), etc.

The following are examples of an input file (this is the format of the file produced by ret-<nn> services):

- 000000130USM01
- 000000131USM01
- 000000132USM01
- 000000133USM01
- 000000134USM01

2. **Output File** – Mandatory. The name of the file in which you want to save the output file. The output file contains the bibliographic document numbers for which no translation occurred. The file can be found later in the $alephe_scratch directory and be used for further processing (for example, print, load in cataloguing client, Search tab, etc).
3. **Create 6XX using: AUT 1XX** or **AUT 7XX** – Mandatory. This selection determines whether the process will create the new subject heading 6XX using the AUT 1XX tag (can be used for English to French translation) or using the AUT 7XX tag (can be used for French to English translation).
    - If 1XX is selected, the system applies the tab_bib_aut_match table in order to find the BIB 6XXx0/2/7 tags that should be translated and the Authority library plus headings index code that is used for finding a matching authority record.
    - If 7XX is selected the system applies the tab_bib_aut_match table in order to find the BIB 6XXx6 tags that should be translated and the Authority library code. The system always uses the GEN heading index of the AUT library for finding a matching authority record.

4. **Update Database** – Mandatory. You can decide whether updated records with new 6XX fields are updated immediately by selecting **Yes**) or whether you want to produce the report information in the standard log file only by selecting **No**). If you select **No**, then you can check the first run. If the run is ok, you can repeat the same run with Update Database= Yes.

57.2 Defining the AUT Index code for Detecting the AUT Headings - tab_bib_aut_match

The Aleph configuration table ./<bib_library>/tab/tab_bib_aut_match defines the Bibliographic tag that should be translated and sets the AUT index code to which the BIB tags are compared in order to find the authority record that can be used for translation.

The tab_bib_aut_match table is used by the Create Additional Subject Heading(s) from Authority (manage-46)service. The following is a partial example of ./usm01/tab/tab_bib_aut_match:

```
! TABLE_KEY 1,2
! COL   1.   5; ALPHA_NUM, UPPER; #;
!            BIB Tag & indicator;
! COL   2.  10; TEXT; ;
!            For second position 7 only: Subfield 2 content;
! COL   3.   5; ALPHA_NUM, UPPER; ;
!            AUT library code;
! COL   4.   5; ALPHA_NUM, UPPER; ;
!           AUT Index Heading code.Irrelevant for 6XX 2nd indicator 6;
! COL   5.   2; ALPHA_NUM, UPPER; ;
!            Filing procedure;
!            Filing procedure for the "text comparison" action.
!            Used for creating a temporary Z01-NORMALIZED-TEXT from
!            Z01-DISPLAY-TEXT without deletion of subfield codes.
!            Default is 99 of the AUT library (make sure it doesn't
!           include filing_del_subfield nor filing_del_subfield_code
!            routines);
! 1     2        3     4   5
!!!!!-!!!!!!!!!!-!!!!!-!!!!!-!!
650#0          USM10 LCS
650#2          USM10 MLC
650#7 aat       USM10 AAT
6###6          USM10
```

Explanation of tab_bib_aut_match activity

1. The content of the Bibliographic record's tag+indicator (set in col. 1) is compared with the Authority heading set in the Authority index code (col.4).

2. Col. 2 is relevant only when second indicator is 7 (e.g. col.1 is set with 650#7). If col.2 is populated, the system ensures that subfield 2 of the tag+indicator (set in col. 1) contains exactly the text entered in col.2. If the

bibliographic records subfield 6XXx7 is not identical to col. 2, no translation occurs.

3. When activating manage-46 using “Create 6XX using AUT 1XX” (English to French), the system looks for the Authority index (col.4) in the Authority library (col. 3).

4. When activating manage-46 using “Create 6XX using AUT 7XX” (French to English), the system looks for the Authority library (col.3) in which GEN index is used. Therefore, col. 4 (Authority index) is irrelevant (GEN index is always used).

5. When activating manage-46 using Create 6XX using AUT 7XX (French to English)**,** the system looks for the Authority library (col.3) and uses GEN Authority index (hard-code). Therefore, col. 4 (Authority index) is irrelevant.

6. The Authority index code (col.4) must be defined and properly set in the Authority database (tab11_acc, tab00.lng, etc).

7. For matching the BIB 6XX to the Authority index, the system activates the filing routine set in col. 5 of tab_bib_aut_match. The filing routine should be set in tab_filing of the AUT library. This routine should not include del_subfield/del_subfield_code so that text is normalized without deletion of subfield codes. The default filing routine is 99. This way, the system compares the text of the bibliographic record (tag+subfield+indicator) and the authority indexed heading in order to determine if a match is detected.

8. The library should set the order of the lines in tab_bib_aut_match according to the translation priority preferences so that, for example, if the bibliographic record does not contain 650x0, the system continues to look for 650x2. If it does not exist, the system continues to look for 650x7 with subfield 2=aat.

9. The library may set multiple lines per bibliographic tag+indicator. The system should prefer 6XXx0, 6XXx2 or 6XXx7 (in that order) and stop the process when a translation for one type of code has been found. However, the process attempts to find translations for all of the occurrences of this preferred tag.

Note: Since the AUT indexes for creating the French 6XX is built through regular indexing setup and processes, with a separate index for 750x0, 750x2 and 750x7 $$2=aat. The filing procedure for these indexes should NOT include del_subfield nor del_subfield_code filing routines and therefore the subfield code will be retained in the AUT Z01 lists that are built for this purpose.

57.3 Manage-46 Service Functionality - Workflow and Example

1. The library prepares in advance an input file using standard Aleph retrieval options, for example, records added from date-to-date or records from system number-to-system number, etc. Such input files can be automatically generated by Aleph standard cataloging batch services. The manage-46 batch service processes each record contained in the input file records.

2. The user activates the service for AUT 1XX or AUT 7XX translation (user selection within service interface).

    - If AUT 1XX is selected, the system applies tab_bib_aut_match to find the Bibliographic fields that should be translated and the authority and index codes to which to make the text comparison. The bibliographic text that should be translated is normalized using the filing routine set in tab_bib_aut_match (filing routine that do not delete subfield codes).

    - If AUT 7XX is selected, the system applies tab_bib_aut_match to find the Authority library (e.g. USM10) in which GEN index code is used for text comparisons. The bibliographic record tag 6XXx6 that should be translated is normalized using the filing routine set in tab_bib_aut_match (a filing routine that does not delete subfield codes).

3. The Bibliographic normalized text is compared with the AUT index text (as defined in tab_bib_aut_match or GEN index). The comparison is made according to the match algorithm (see details below).

4. If a single authority record is found, the BIB is enriched with a new 6XX.
    - If AUT 1XX is selected - a new 6XXx6 is created
    - If AUT 7XX is selected - a new 6XXx0 is created

    The newly created field is written directly after the source field.
5. The procedure is repeated for each of the 6XX fields. If no translation whatsoever occurs, the BIB record number is reported in the batch output file. The output file is in standard Aleph retrieval format and can be used for further processing. The following is an example of an output file:

000000131USM01
000000132USM01

57.4 Match Algorithm and Translate

Following is a description of the match and translate algorithm for both translations:

- Create 6XX using 1XX (English to French)
- Create 6XX using 7XX (French to English)

57.4.1 Match and Translate for Create 6XX Using 1XX

When a user activates the service with "Create 6XX - using 1XX" (English to French):

1. The batch process of a record first checks if there is at least one 6XXx6 field in the BIB record (Laval's French subject heading):

    - If yes, (i.e. there is a 6XXx6), the record is not processed. This record is not listed in the output file.

- If no, (i.e. there is no 6XXx6), the system activates the "match algorithm" in order to create a new 6XXx6 field.

2. The fields listed in columns 1 and 2 of tab_bib_aut_match are the candidate fields for translation. The library should set the lines in tab_bib_aut_match in the following order so that the translation occurs for:
    - 6XXx0, or
    - 6XXx2 if there is no 6XXx0, or
    - 6XXx7 containing $$2 aat, if there is no 6XXx0 or 6XXx2
3. Text normalization (filing routine) is activated for the fields.
4. Search full text string for match: BIB 6XX is compared with the AUT index.

    - If the procedure runs smoothly (the entire text is translated and there are no multiple possible translations), a new 6XXx6 field is created using the 1XX of the matching AUT record. The record is not listed in the output file. The newly created 6XXx6 field is written directly after the source field.

    - If a translation is not found for the first subfield, the procedure stops for this field. The system checks other fields. If no other fields are translated (a new 6XXx6 is not created), the record is listed in the output file.

    - If multiple translation records are found, the procedure stops for this field. If no other fields are translated (new 6XXx6 is not created); the record is listed in the output file.

When checking for a match:

- If a match is not found, the system drops the last subfield and repeats the search. The dropped subfield and its position are kept in a buffer for a separate search.
- If a match is not found, the system drops the last two subfields and repeats the search. The subfield buffer is cleared and dropped subfields and their position are kept in a buffer for a separate search.
- If a match is found, and the subfield buffer is not empty, the system deals with the subfield buffer in same manner.
- If there is a non-translated subfield, the system cuts off the field from that subfield to the end. This is still considered a successful match.
- If the first subfield is not translated, or if there are multiple options for any of the strings, this is considered a non-successful match.
- If the process does not create at least one successful match for a single record, the document number is written on the output file.

In this process, subfield codes should match and the geographic subdivision is treated the same as other subdivisions. The system keeps the order of subfields as in the original text.

Example cases:

Case 1: - Full "match" is found and BIB 650x6 is created with AUT 1XX content.

BIB record 000000123 contains an English Subject heading:
650_0 $a Gardens
$x Styles
$x History

$y 20th century
$v Pictorial works

The Authority database contains a matching French heading (7XX_0) derived from 3 different records:

AUT record 000000001
150__$a Jardins
$x Styles
750_0 $a Gardens
$x Styles

AUT record 000000002
180_0$x Histoire
$y 20e siècle
780_0$x History
$y 20th century

AUT record 000000003
185_0$v Ouvrages illustrés
785_0$v Pictorial works

manage-46 activation causes an update of the BIB record with a new field. The source field is retained as is:

650_0 $a Gardens
$x Styles
$x History
$y 20th century
$v Pictorial works
650_6 $a Jardins
$x Styles
$x Histoire
$y 20e siècle
$v Ouvrages illustrés

Case 2 – Partial "match" is found.

BIB record 000000123 contains an English Subject heading:
650_0 $a Gardens
$x Styles
$x History
$y 20th century
$v Pictorial works

The Authority database contains:

AUT record 000000001
150__ $a Jardins

$x Styles
750_0 $a Gardens
$x Styles

AUT record 000000003
185_0$v Ouvrages illustrés
785_0$v Pictorial works

AUT record 000000004
185 0 $x Histoire
785 0 $x History

The Authority database does NOT contain a record for "$x History $y 20th century" nor "$y 20th century"

manage-46 activation causes an update of the BIB record with a new field. Source field retains as is:

650_0 $a Gardens
$x Styles
$x History
$y 20th century
$v Pictorial works
650_6 $a Jardins
$x Styles
$x Histoire

57.4.2 Match and Translate for Create 6XX Using 7XX

When user activates the service with "Create 6XX - using 7XX" (French to English):

1. The batch process of a record first checks if BIB record contains 6XXx0 or 6XXx2 or 6XXx7 + $$2=aat
    - If yes, record is not processed. This record is not listed in the output file.
    - If no, the system activates the "match algorithm" in order to create new 6XXx0 field.
2. The BIB record 6XXx6 field is used to match with the GEN Authority index of the Authority library that is set in column 3 of tab_bib_aut_match.
3. Text normalization (filing routine) is activated for the BIB 6XXx6 field (set in col. 5 of tab_bib_aut_match (make sure it does not include del_subfield/del_subfield_code routines).
4. The service searches full text string for a match. BIB 6XX is compared with the GEN index.
5. If the procedure runs smoothly (the entire text or part of it is translated and there are no multiple possible translations), a new 6XXx0 field is created. The content of the field is based on the AUT 7XXx0 (LC subject heading) of the matching Authority record. The record is not listed in the output file. The newly created 6XXx0 field is written directly after the source field.

6. If multiple translation records are found, the procedure stops for this field. If no other fields are translated (new 6XXx0 is not created), the record is listed in the output file.

When checking for a match:

- The entire string is searched for and if it is not found, the last subfield code is cut off, cursively. The search stops when a match is found.
- No buffer for remaining non-matched is required (i.e. no need to keep non-translated subfields for a separated translation).
- Non-found subfields are dropped.

In this process, subfield codes should match and geographic subdivisions are treated the same as other subdivisions, i.e. if they are not in RVM (AUT), subfields are cut-off. The system should keep the order of subfields as in the original text.
Example case:

"Match" is found for a single subfield and BIB 650_0 is created with AUT 7XX content.

BIB record 000000123 contains French Subject headings:

650_6 $a Jardins
$x Styles
$x Historie
$y 20e siècle
$z Ouvrages pictoriales

The authority database contains a matching English heading (1XX_0) derived from 1 record:

AUT record 000000001
150__ $a Jardins
750_0 $a Gardens

manage-46 activation causes an update of the BIB record with a new 650_0 field. The source field remains as is.

650_6 $a Jardins
$x Styles
$x Historie
$y 20e siècle
$z Ouvrages pictoriales
650_0 $a Gardens

58 Automatic Translation of Bibliographic Note Fields

Aleph can automatically translate bibliographic note fields from one language to another. To enable this feature, configure the following files:

- Fix record routine: fix_doc_notes (set in ./xxx01/tab/tab_fix)
- Aleph table: tab_fix_notes - list of translations, per Bibliographic tag and subfield (./xxx01/tab/ tab_fix_notes).

58.1 Fix Routine for Translation - fix_doc_notes

To configure the automatic translation of Bibliographic fields, set fix_doc_notes routine in tab_fix.

The following is an example of ./usm01/tab/tab_fix

```
! 1                 2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INS2  fix_doc_notes
```

As with all other fix routines, you can set the fix routine (in column 1 of tab_fix) to be invoked in various routines, for example, online record handling, batch loaders, and batch updates) so that its activation is part of the bibliographic record management.

This fix routine replaces the text within a bibliographic field with a translated text defined in the tab_fix_notes Aleph configuration table.

58.2 Setting Up a List of Translations - tab_fix_notes

The Aleph configuration table, ./xxx01/tab/tab_fix_notes, contains a list of translations per tag and subfield. The library may edit this table to include translations of relevant tags and subfields.

The following is a partial example of ./usm01/tab/tab_fix_notes. (The actual lengths of columns 3 and 4 are longer than that what they appear in the following table structure.)

```
! TABLE_KEY 1,2,3
! COL   1.   5; ALPHA_NUM, UPPER; #;
!               Tag & indicator;
! COL   2.   1; ALPHA_NUM, LOWER;  ;
!               Subfield;
! COL   3.  150; TEXT; ;
!               Source text;
! COL   4.  150; TEXT; ;
!               Translated Text;
! 1   2                   3                        4
!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>-!!!!!!!!!!!!!!!!!!!!!!!!->
```

```
245## h [electronic resource]                [ressource électronique]


310## a Annual                               Annuel
310## a Biannual                             Semestriel
310## a Bi-annual                            Semestriel
310## a Biennial                             1 no par 2 ans
310## a Bimonthly                            Bimestrie
500## a Adaptation of:                       Adaptation de :
500## a Added t.p. title                          Titre de la p. de t.
addit. :
500## a At head of title                     En tête du titre
500## a Caption title                        Titre de départ
500## a Cataloguer's title                  Titre du catalogueur
500## a Colophon title                             Titre de l'achevé
d'imprimer
500## a Cover title                         Titre de la couv.
```

58.3 Automatic Translations – Functionality and Examples

The following is a description of the automatic translate feature that is implemented when fix_doc_notes is activated for a bibliographic record:

58.3.1 Compare Action

The system compares the source table tab_fix_notes values with the bibliographic record. The fix routine (fix_doc_notes) reads the source list (tab_fix_notes) and compares it with the bibliographic record relevant tag + subfield left-anchored text.

Notes:

- The source file has to list the lines for a specific tag one after the other. For example, if there are 10 translation lines for tag 500$$a, group all lines (similar to the example above of tab_fix_notes).
- In the source file, list longer phrases before shorter ones.
- The comparison is not case sensitive.

58.3.2 Replace Action

1. The matched text is replaced with the translated text as defined in the source file.

Notes:

- The phrase in English must exist in its entirety in the bibliographic record in order for the translation text to replaces it.
- Text that is not left aligned is not translated.

2. Parts of a field that do not have a matching string in the source file are copied as is into the translated field.
3. The source field is overwritten.

For example:

The following is an example of the tab_fix_notes source:

```
! 1   2                    3                          4
!!!!!-!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>!!!!!!!!!!!!!!!!!!!!!!!->
500## a Title from container            Titre du conteneur
500## a Title romanized:                Titre converti en
écriture latine:
500## a Title                           Titre
```

Case 1:

BIB record 000000123 contains the following text:

```
500## $$aTitle from container
```

tab_fix_notes contains the following text:

```
500## a Title from container         Titre du conteneur
```

This causes an automatic translation of the note to:

```
500## $$aTitre du conteneur
```

Case 2:
BIB record 000000456 contains the following text:

```
500## $$aTitle from back cover
```

There is no translation for `Title from back cover` so the following line in the source field:

```
500## a Title                           Titre
```

changes the note field to:

```
500## $$aTitre from back cover
```

Case 3:
Bibliographic record 000000789 contains the following text:

```
500## $$aAdditional Title from container
```

There is no translation for the left aligned text `Additional`; therefore, the line does not change:

```
500## $$aAdditional Title from container
```

59 Link to RDA Toolkit

GUI-Cataloging provides the option to access the RDA (Resource, Description and Access) toolkit.

There is an option in GUI-cataloging-Edit Action menu: Show RDA Toolkit.

This option enables the linking of the MARC21 BIB and AUT cataloging tags to the RDA toolkit.

RDA users are registered in the RDA Web site. When applying to RDA services, Aleph uses the username and password set in the RDA site.

Set this information the PC file: CATALOG.INI

```
[RdaToolkit]
RdaEnabled=Y
RdaUser=user_name
RdaPassword=Password
RdaURL=http://access.rdatoolkit.org/?encode=marc&rec=
```

If localization is needed, insert the following line in Catalog\Tab\Lng\MENU.DAT:

```
SHOW_RDA                                    Show RDA Toolkit
```

60 Update HOL Record Based on Item Arrival

Holding records can be automatically updated with summary holdings information (86X fields) based on the arrival registration of the linked items.

For this functionality, there is no need to define an expand/fix routine. The relevant set up is done in the ADM Library's tab_z105 table:

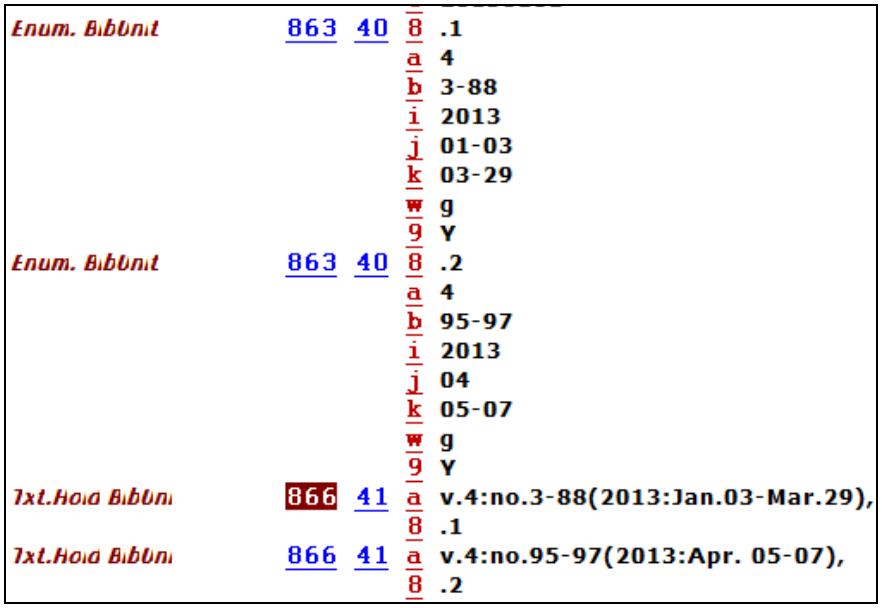
Example of ./usm50/tab/tab_z105:

```
!      1         2   3     4     5     6     7     8     9    10    11
12
!!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!!!!!-!!!!!
UPDATE-ENUM     n USM60
```

This routine is relevant only if the 85x/85xX Publication Pattern fields reside in the HOL document record and the items are linked to an HOL Record.
Make sure util e-11 is activated in z105_library (for example, USR00).

Message type (column 2) n: update HOL record with 86X info (Summary Holdings) using expand_doc_hol_86x expand routine (for more information about expand_doc_hol_86x, see the *Aleph 22 System Librarian Guide-Indexing*).

The above setup updates 863 and 866 fields, for example in the HOL record, classifying the items that are registered as arrived.



There is an option to set the above UPDATE-ENUM section with message type (column 2): o

```
!      1         2   3     4     5     6     7     8     9     10    11
12
!!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!!!!!-!!!!!
UPDATE-ENUM     o USM60
```

Message type o: update HOL record with 86X info (Summary Holdings) using expand_doc_hol_86x_iso expand routine. (For more information about expand_doc_hol_86x_iso, see the *Aleph 22 System Librarian Guide-Indexing*).

System Librarian's Guide - Circulation

Version 23

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1. PATRONS IN A MULTI-ADM ENVIRONMENT 10

2. ITEM MATERIAL TYPES 10

3. SUBLIBRARIES 11

3.1. Loaning and Returning Items per Sublibrary 12

4. ITEM STATUSES 13

5. PATRON STATUSES 16

6. DUE DATES, FINES AND LIMITS 18

7. OVERDUE NOTICE SCHEDULE 29

7.1. Overdue Notice Letters and Summary 30

8. PATRON BLOCKS 31

9. CASH TRANSACTIONS 33

9.1. Setup of Cash Transactions (tab18.lng) 33

9.2. Registration of Cash Transactions in Offline Circulation 35

10. CIRCULATION TRANSACTION CHECKS (TAB_CHECK_CIRC) . 36

11. HOLD AND PHOTOCOPY REQUESTS 42

11.1. Hold Request Checks 42

11.2. Priority 49

11.3. Pickup/Delivery Locations 50

11.4. Request Service Hours Checks (tab_service_hour) 50

11.5. Hold Shelf 51

11.6. Printing Call Slips 51

11.7. Item Holds List 51

11.8. Loans List 51

11.9. Hold Request Log 52
11.10. photo_request_style 52
11.11. KEEP-FILLED-PHOTO 52
11.12. PHOTO-USE-EXACT 53
11.13. CREATE-Z38H 53
11.14. Shipping 53
11.15. Rush Cataloging Requests 53
11.16. Requesting Unbarcoded Items 54
11.17. Requests Refusal 55
11.18. Request Fulfillment by Preferred Items 56
11.18.1. Defining the Preference Order of Materials 56
11.18.2. Defining the Relevant Pickup Locations for which the Preferred Items Mechanism is Applied 57
11.18.3. Defining Check Routines to Check if the Preferred Item is Available 58
11.18.4. UE_06 Processing of the Requests 58
11.19. Staff Request 58
11.19.1. Defining the Subscribed Library's Account 59
11.19.2. Adding "requester name" Fields to Hold and Photocopy Forms in the OPAC 59
11.19.3. Adding "requester name" Columns to Request Lists 59
11.19.4. Adding Sort Options Based on the Requester Name 60
12. SETUP FOR RECALLS 61
12.1. tab16 61
12.2. tab18.lng 61
12.3. tab100 62
13. PRINT TEMPLATES 63
13.1. Available Templates 63
14. DROP-DOWN LISTS 68
15. CATALOG ITEM 69
16. ITEM RESHELVING TIME 70
17. ITEM LIST 71

18. ADVANCE BOOKING SLOTS 72

19. VALID DATES OF PATRON ADDRESSES 74

20. ITEM HISTORY AND SUMMARY WINDOW 76

21. LOAN AND RETURN WINDOWS 76

22. CASH 76

22.1. Cash Transaction Information Window 76

22.2. Cash Receipt Numbering 77

22.3. E-mails Free of Charge 77

22.4. Cash Reports 78

23. SQL QUERIES 78

24. AUDIO WARNINGS 79

25. ITEM PROCESS STATUS 81

26. OVERRIDING A BLOCK FOR LOANS AND RENEWALS 81

27. CLIENT SETUP (CIRC.INI) 82

28. CIRCULATION VALUES OF PC_SERVER_DEFAULTS 95

29. CIRCULATION VALUES OF ALEPH_START 98

30. CHANGE ITEM INFORMATION DISPLAY 99

31. TABLES 100

31.1. Library Tables 100

31.2. alephe Tables 111

32. TAB100-RELATED ENTRIES IN CIRCULATION 111

33. CIRCULATION VALUES OF WWW_SERVER.CONF 126

34. RETURN RECEIPT 127

35. COLUMN HEADINGS (PC_TAB_COL.LNG) 128

36. OVERNIGHT LOAN SETUP 130

37. PRINTING IN CIRCULATION 131

38. PATRON PICTURE 131

39. EXPAND DISPLAY 132

39.1. Display of Workstation Identity 132

40. CIRCULATION VALUES OF PROF_LIBRARY.CONF 132

41. DUPLICATE PATRON IDS 132

42. REMOTE REGISTRATION 133

42.1. Remote Registration Settings 134

43. DIFFERENT HOLD REQUEST FORMS (TAB_HOLD_REQUEST_FORM) 136

43.1. tab_hold_request_form 136

43.2. Creating a Different Set of Hold Request Forms 137

44. DIFFERENT PHOTOCOPY REQUEST FORMS (TAB_PHOTO_REQUEST_FORM) 138

44.1. 43.1 tab_photo_request_form 138

44.2. Creating a Different Set of photocopy Request Forms 139

45. PATRON RETRIEVAL BY CONCATENATION OF ID AND VERIFICATION 139

46. DEFINING THE ZIP CODE FORMAT FOR A PATRON'S ADDRESS IN PRINTOUTS 140

47. LOG FILE EXPANSION 142

48. BOOKING REQUESTS 142

48.1. Patron Booking Permissions 142

48.2. Staff Booking Permissions 143

48.3. Item Booking Ability 143
48.4. Maximum Number of Bookings and Maximum Booking Period 144
48.5. Like Item Policy 144
48.6. Booking Request Checks 145
48.7. Booking Request Default Times 147
48.8. Delivery and Pickup Location Policies 148
48.8.1. tab37_booking_pickup 148
48.8.2. tab37_booking_delivery 149
48.8.3. tab_delivery_locations.lng 149
48.8.4. tab27 149
48.9. Cash Transactions 149
48.10. Policy of a Booking Request Fulfillment 150
49. DISPLAYING ITEM INFORMATION 150
50. PATRON DIRECT QUEUE 151
50.1. Patron Consortium Privileges 151
50.1.1. Creating the Consortium Privileges 151
50.2. Setting the Library's Policies for PDQ 152
50.2.1. Setting Library Global Policies 152
50.2.2. Setting Patron-specific Policies 153
50.3. Creating a Title Request 153
50.3.1. The Pool of Items 153
50.3.2. The Pickup List 153
50.3.3. The Roster 155
50.3.4. Creating the Request 156
50.4. Fulfilling the Request 156
50.5. Loan and Return Anywhere 156
50.6. Staff Permissions 157
51. MANAGING LOST ITEMS 157
52. SMS MESSAGING 158
52.1. Sending SMS from ALEPH 159
52.2. Sending SMS by Using Third Party SMS Servers 160
53. PATRON NAME IN JAPANESE – KANJI AND KATAKANA 160

53.1. Storing Patron Name in Kanji and Katakana 160

53.2. Sorting Patron Name in Kanji 162

53.3. Searching for Patron Name in Kanji and Katakana 163

53.3.1. Search for Patron Kanji Name 163

53.4. Patron Keywords Search – Kanji and Katakana 163

53.5. Z111_index table 164

54. ALEPH – GATEWAY COMPATIBILITY 165

54.1. Events that Trigger Messages for the Gateway 166

54.2. Configuration and Activation 166

55. ALEPH AND SSO INTEGRATION 167

55.1. Events that Trigger Messages for the SSO 168

55.2. Activation and Setup 169

56. LOG FOR PATRON RECORD CHANGES 169

56.1. tab_patron_log.<lng > 170

56.2. Patron Log Record Changes Report 171

57. TWO PATRON NAME FIELDS 172

57.1. PATRON-NAME-FORMAT 172

57.2. Migrating the Current Single Name Field (Z303-NAME) into Two Name Fields 173

57.2.1. Input Parameters 173

57.3. Implementation Notes 174

58. ALEPH SUPPORT FOR RFID 176

58.1. Overview 176

58.2. System Architecture 177

58.3. Functionality 179

58.3.1. Identifying Items 179

58.4. Check-Out and Check-In 180

58.5. Self-Check 181

58.6. Registering Item Information 181

58.7. Configuring RFID- Nedap, 3M, and Bibliochip 181
58.7.1. Configuring RFID Reader Location and Vendor 181

58.8. Activating Reader and Success Messages 182

58.9. Configuring the RFID Web Server 182

58.10. Configuring writing of action data 183
58.10.1. tab_rfid 183
58.10.2. tab_rfid_translate 184

1. Patrons in a Multi-ADM Environment

In a multi-ADM environment, each ADM library can define whether its patron records (Z303, Z304, Z308, Z353) will participate in the shared or the non-shared section of the database (USR00).

The USER-SHARING variable in `tab100` of each library's `tab` directory defines whether it participates in the shared environment. If the ADM library has defined USER-SHARING=Y, then the patrons created by this library will appear in the Patron List when a staff user is connected to ANY library that is defined as Y in the USER-SHARING variable.

Patrons belonging to an ADM library which has defined the variable USER-SHARING=N in `tab100` will not be included in this Patron List. If the ADM library defined USER-SHARING=N, then the patrons created by this library will appear in the Patron List only when a staff user is connected to this specific library.

When new patrons are created, the following rules always apply:

- Whether or not logins and passwords are automatically created based on system counters depends on the variables DEFAULT-BOR-ID, DEFAULT-BOR-ID-1, DEFAULT-BOR-VERIFY-1 and DEFAULT-BOR-VERIFY-2 as defined in the ADM library tab100 table.
- If the system is set to automatically create login and passwords, it will first try to use the ADM library counters last-bor-id, last-bor-id-1, last-bor-verify-1 and last-bor-verify-2. If the relevant counters are not found in the ADM library, the usr_library counters will be used.

2. Item Material Types

Define the types of materials (books, periodicals, etc.) that are available in your library system.

In order to define material types, complete the following steps:

1. In terminal mode, select an administrative library. After you have completed all the steps below, choose another administrative library and complete the steps again. Follow this procedure until all administrative libraries have been covered.

2. Edit `tab25.lng`. This table enables you to list the material types available in your library. Remember that the option ISSUE has a special functionality related to the Serials module.

3. Test the system to make sure that the English language version works properly.

4. Translate `tab25.lng`, which you edited in English in Step 2 above, in order to list material types in the language appropriate for your patrons. Translate the

table.

Translate only the description and not the code for the material type.

3. Sublibraries

Define valid sublibraries using the `tab_sub_library.lng` table. This table also sets up information that is used by three other tables (tab15, tab16 and tab17) and should therefore be filled in together with the other tables.

Following is a sample from the table:

```
! 1   2   3   4  5                                6     7     8     9
10
!----!-!-----!-!-------------------------------!-----!-----!-----!----
-!-----!
UARCV 4 USM50 L Archives Library               15A   UARCV UARCV
UARCV
UELEC 1 USM50 L Electronic Resources           15A   UELEC UELEC
UELEC
URLEC 4 USM50 L UELEC Reading Room             15A   UELEC UELEC
UELEC
UGDOC 1 USM50 L Government Documents           15A   UGDOC 17A
UGDOC
URDOC 4 USM50 L UGDOC Reading Room             15A   UGDOC 17A
UGDOC
UEDUC 1 USM50 L Education Library              15A   16A   17A
UEDUC ALEPH
UHLTH 1 USM50 L Health Library                 15A   16A   17A
UHLTH ALEPH
ULAW  1 USM50 L Law Library                    15A   16A   17A   ULAW
ALEPH
```

Key to Table:

- **Column 1**: Enter here the sublibrary code being defined. To define all sublibraries of a particular ADM library, enter first the code of the ADM library. When you want to define a sublibrary for a global patron (that is, the patron is not assigned to a specific sublibrary), enter the code *ALEPH*.

- **Column 2**: Enter the code for the sublibrary type:

    - 1 - full sublibrary (patron and items)
    - 2 - patron sublibrary only (not used for items)
    - 3 - special (for photocopy and hold requests delivery site, for example, HOME or BOX)
    - 4 - reading room sublibrary (for requests that are delivered to the Reading Room)

- 5 - ordering unit (for Acquisitions)

- **Column 3**: Enter the code of the ADM library to which the sublibrary in column 1 belongs. Leave column 3 blank if column 1 contains the code for the ADM library or the ALEPH global library.

- **Column 4** is the Alpha code for the character set of the menu choice. The standard is L, which stands for Latin.

- **Column 5** is the name of the sublibrary as it will appear in the Web and GUI interfaces. In order to define library names for the WEB Opac use table tab_www_item_desc.eng.

- **Column 6**: The Group ID identifies the section in tab15 that is relevant for this sublibrary. Type here the Group ID that should be entered in column 1 of `tab15` (the table for item statuses). This allows for the definition of common statuses in `tab15` that are used by a number of sublibraries. It would mean that only the item statuses that belong to the group ID are valid for use in that sublibrary.

- **Column 7**: The Group ID identifies the section in `tab16` that is relevant for this sublibrary. Type here the Group ID that should be entered in column 1 of `tab16` (the table for due dates, fines and limits). It means that the only definitions valid for this sublibrary are those that relate to this group ID.

- **Column 8**: The Group ID identifies a group of sublibraries that share a set of open/close hours. Type here the Group ID that should be entered in column 1 of `tab17` (the table for library hours). It means that this sublibrary relates to the opening hours set in this group ID.

- **Columns 9-11**: Items belonging to the sublibrary defined in column 1 may be borrowed by patrons that have borrowing privileges in the sublibraries shown in columns 9-11. The order in which the system checks the patron records is determined by the order in which the sublibraries appear in columns 9-11.

3.1. Loaning and Returning Items per Sublibrary.

In ALEPH, the loan and return of items from a specific sublibrary is not controlled via the staff user names and passwords (in the Staff Privileges option) but rather via the IP identification of the PC or a special ALEPH Station ID.

To identify the Station ID for a PC, in Windows Explorer (checked for Windows NT and Windows 2000), right-click My Computer, then click **Properties**>**Network Identification**>**Properties**. The station ID appears in the Computer Name field. To identify and set the special ALEPH Station ID, right-click the button at the right of the status bar, then click **Set Workstation Identifier**.

This mechanism for controlling loan and return for specific sublibraries is contained in the `$data_tab/tab_attr_sub_library` table. This table defines, for a particular workstation, or group of workstations, which sublibraries come under its jurisdiction. It lists the sublibrary relationships based on station ID / IP address with regard to loan/return, cash and reading room management. For example:

```
!       1             2   3     4     5     6     7
!!!!!!!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!>
10.1.555.125          1 WID   LAW   MUS   HYL   USM50
                      1 HIL   GDOC5 MED
                      3 WID   LAW   MUS   HYL   USM50
                      3 HIL   GDOC5 MED
                      4
                      5 USM50
                      6
                      7
                      8
```

Key to the tab_attr_sub_library table

- **Column 1 -** Station identification (Station ID or IP address). Column 1 can have asterisks, such as:

  10.1.555.* for sub library a

  and

  10.1.555.* for sub library c

- **Column 2** – Attribute type. Two attributes are relevant for loan and return:

  1 - item can be loaned and returned

  2 - item can be returned

- **Columns 3, 4, and so on** - Sublibrary names**.**

4. Item Statuses

You need to define the statuses that may be assigned to items (copies) and various conditions related to each status, such as whether an item:

- can be loaned
- can be renewed
- can be available for hold and photocopy requests
- will be displayed in the Web OPAC

In order to define item statuses, complete the following steps:

1. In terminal mode, select an administrative library. After you have completed all the steps below, you should choose another administrative library and

complete the steps again. Follow this procedure until all administrative libraries have been covered.

2. Edit `tab15` in English to define the statuses that may be assigned to items (copies) and various conditions related to each status. You must first edit the English version of this table and only later translate the table (see below).

3. Test the system to make sure that the English-language version works properly.

4. Translate `tab15` in order to list item statuses in the language appropriate for your patrons.

Following is a sample from the `tab15` table:

```
! 1   2  3  4        5                          6 7 8 9 1 1 1 1 1 1  1 1
!                                                  0 1 2 3 4 5  6 7
!!!!!-!!-!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-!-!-!-!-!-!!-!-!
HIL   ## BD L At bindery                        Y N Y N Y Y N N N 00 Y A
HIL   ## BP L Bindery prep                      Y Y Y N Y Y N N N 00 Y A
HIL   ## CA L Cancelled                         Y N N N N Y N N N 00 Y A
HIL   ## CL L Claimed                           Y N N N N Y N N N 00 Y A
HIL   ## CT L Cataloging                        Y Y Y N Y Y N N N 00 Y A
HIL   ## MI L Missing                           Y N Y N Y Y N N N 00 Y A
HIL   ## MS L Lost                              Y N N N Y Y N N N 00 Y A
HIL   ## NA L Not arrived                       Y N N N N Y N N N 00 Y A
HIL   ## NP L Not published                     Y N N N N Y N N N 00 Y A
HIL   ## OI L Order initiated                   Y N N N N Y N N N 00 Y A
HIL   ## OR L On order                         Y N Y N Y Y N N N 00 Y A
HIL   ## RE L Ordered--received                Y N Y N Y Y N N N 00 Y A
HIL   92 HD L Depository                        Y Y N N Y Y N Y Y 00 Y A
HIL   28 HD L 28-day loan (depository)          Y Y C N Y Y N Y Y 00 Y A
HIL   28 ## L 28-day loan                       Y Y Y N Y Y N Y Y 00 Y A
HIL   04 ## L Restricted loan                   Y Y N N Y Y N N N 00 Y A
HIL   02 ## L In-library use                    Y Y C N Y Y N N N 00 Y A
HIL   01 HD L Regular loan (depository)         Y Y C Y Y Y N Y Y 00 Y A
HIL   01 ## L Regular loan                      Y Y Y Y Y Y N Y Y 00 Y A
```

Key to Table:

- **Column 1**: Group ID for group of definitions. In the above example, there is one group of definitions. You can create as many groups as you want. By entering the Group ID in column 6 of the `tab_sub_library` table, a number of different sublibraries can use the same group of definitions. Be sure that you fill in the tab15 table together with the `tab_sub_library` table.

- **Column 2**: Item status code. This is the code by which the system identifies the status. The code must be unique within one group, but does not have to be unique across groups. This means that the same status code may have different meanings in different groups. The status code must be two digits. For example, for status number 1, enter *01*. Note that you may NOT use status code *00*.

- **Column 3** is the two-character code for the item's process status.

- **Column 4** is the code for the character set of the column heading. The standard is L, which stands for Latin.

- **Column 5**: Item status name. Enter here the name of the status, such as *overnight* or *one week*. The name may be up to 15 characters long.

  Note: In the GUI-Item Form-Item Status drop-down list: If in `tab15` there is more than one occurrence of the same item status code (of the same Group ID), the drop-down menu of the Item Status field will display the Item Status Name of the line that has ## in the Process Status column rather than the Item Status Name of the first occurrence of that item status code.

- **Column 6**: Can be loaned. Determine whether or not items with this status may be loaned. Enter *Y* for Yes or *N* for No.

- **Column 7**: Can be renewed. Determine whether or not items with this status can be renewed. Enter *Y* for Yes or *N* for No.

- **Column 8**: Hold requests. If the item is on an open shelf (Y), the system will check to see if the item is available when a hold request is placed. If the item is available but the patron does not have permission to place a request on an available item, a message will be displayed. If the item is out on loan, the system will place a hold request for the item.

  If the item is in a closed stack (C), hold requests are always allowed. If the item cannot be requested (for example, lost, permanently shelved outside the library, privately held, etc.), enter *N*. Availability will not be checked and no hold request will be placed for the item.

  Options "T" and "R" may be used for items that can be requested but have not yet been made available for circulation. In both cases, requesting the item will be allowed even though the item is not registered as checked out. The p_cir_12 (print request slips) and ue_06 (online printing of request slips) jobs consider the item as not available.

  The "R" option causes the hold request submission to trigger the creation of a cataloging trigger, which is used to indicate to the Cataloging department that the cataloging of the item should be prioritized. In addition, the hold request submission is blocked if the requested item has available like items.

  The "U' option is also for items that have not been registered in the system. If this option is configured, when a privileged patron requests the item, a new temporary item is created. The request is placed on the new temporary item.

  For more information see **Requesting Unbarcoded Items** on page **54**.

- **Column 9**: Available for photocopy requests. If you want a patron to be able to put a photocopy request on the item, and the item is accessible to the patron enter *Y*. If the item cannot be requested, or is unavailable for photocopy requests (because, for example, the item is lost, permanently shelved outside the library, privately held, etc.), enter *N*.

- **Column 10**: Display item in Web OPAC. Determine whether or not the item will appear in the Web OPAC when the patron searches for items. Enter *Y* or *N*. If set to *N*, advance booking slots for this item are not created. (Note: the item is always displayed in the GUI.)

- **Column 11**: If this item is requested in the Web OPAC, the patron will be given this specific item and not another copy. Enter *Y* or *N*.

- **Column 12**: Limit the number of hold requests that may be made for an item. If Y is entered, the number of hold requests is limited to the number of copies.

- **Column 13**: Recall. Enter Y if a recall is allowed. Otherwise enter N.

- **Column 14**: Rush Recall. Enter Y if a rush recall is allowed. Otherwise enter N.

- **Column 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 counting from the time he returned it. Use "00" or " " (blanks) for unrestricted reloaning. If the item has been loaned to another patron in the meantime, there is no restriction in any case.

- **Column 16:** Controls Booking permission. Enter Y when Booking is possible, N when Booking is not possible, O if Booking is possible only during open hours, and C if Booking is possible only during closed hours.

- **Column 17:** Controls Booking hours. Enter A if Booking is possible at all times (both open and closed hours), O if Booking is possible only during open hours, and C if Booking is possible only during closed hours.

5. Patron Statuses

Define the statuses that may be assigned to patrons and various conditions related to each status, such as whether the patron may:

- Borrow items
- Renew items
- Place hold and photocopy requests
- Place multiple hold requests

- Request items that are on the shelf.
- Request to receive an item in the reading room
- Book an item that is not available for all.

In addition, you can define the patron's:

- Cash balance limit
- Expiration date of library privileges
- Hold request priority

You can also define whether, for a particular patron status:

- The system will check for restrictions before loaning an item to the patron.
- The librarian may override restrictions (such as fines due) on the patron's ability to borrow an item.
- The system will ignore late returns and not charge the patron a fee.
- The patron will be charged for photocopy requests.
- The patron's details will be displayed in the OPAC where an item that is checked out to the patron is displayed.
- The system will ignore library open hours restrictions when booking an item.
- A default ALEPH Z305 record will be created when the patron is registered at the sublibrary.

In order to define patron statuses, complete the following steps:

1. Edit `tab31`. This enables you to define the statuses that may be assigned to patrons and various conditions related to each status. This is not a language-sensitive table, which means that there is no need to translate it later. You can assign specific patron statuses to a specific sublibrary. When an attempt is made to use these statuses in other sublibraries, a message will appear.

   Whenever the patron is assigned the first local record, the system automatically creates an ADM level record for the patron. Automatically creating an ALEPH record depends on the setup of tab31, in column 22. You can create an overall definition of privileges that will be globally used whenever an ADM level or an ALEPH record is created for any patron status. Do this by defining a line with the ADM code or with ALEPH in the sublibrary column.

   Note that if an ALEPH record is to be created, the patron status with which this record will be created depends on the patron status that is being registered and the mapping that is defined in tab_map_privileges.

2. Edit `pc_tab_exp_field.eng` (in English). This table enables you to define the menu options that appear whenever the librarian needs to choose a patron status. Therefore, the list of patron status names (as opposed to patron status codes) in pc_tab_exp_field.eng must correlate with the patron status codes defined in tab31 (step 2 above). In the `pc_tab_exp_field.eng` table, the code that identifies the correct menu is *BOR-STATUS*. If you want the drop-down

list to be sublibrary-sensitive, place the BOR-STATUS lines only in `pc_tab_exp_field_extended.eng`. The file `pc_tab_exp_field_extended.eng` can be edited.

You must first edit the English version of this table and only later translate the table (see below).

3. Test the system to make sure that the English-language version works properly.

4. Translate `pc_tab_exp_field.eng`, which you edited in English in Step 3 above, in order to provide menu options in the language appropriate for your patrons.

6. Due Dates, Fines and Limits

Due dates are computed daily for combinations of item/patron status, and take into account days on which the library is closed.

The results of the computation may be viewed in the GUI Circulation module by choosing the *Dates* option on the Main Menu and then *Setup for Future Loans* from the drop-down menu. Specific due dates may be changed by the librarian by using this Setup function. The changed date will stay in effect only until the next day, after which the due dates are re-computed.

There are two prerequisites that must be met before you can define due dates:

Prerequisite 1:
Define Item Statuses
Before defining due dates, you must already have defined item statuses. If you have already completed Prerequisite 1, skip to Prerequisite 2.

Prerequisite 2:
Define Patron Statuses
Patron statuses indicate the privileges allowed for each type of patron.

Steps for Defining Due Dates, Fines and Limits
Once the prerequisites have been met, define due dates by completing the following steps:

Step 1: Select a Library
In terminal mode, select an administrative library. After you have completed all the steps below, you should choose another administrative library and complete all the steps again. Follow this procedure until all administrative libraries have been covered.

Step 2: Define Rules for Due Dates

Use tab16 in the ALEPHADM GUI module to edit tab16 to define rules for computing due dates, fines and borrowing limits. Following is a sample from the table (it is broken into two sections because the table is very wide).

```
 !                                                          1 1          1
! 1   2  3  4  5    6     7  8  9    10   11   12  13  4 5    16    7  18
!!!!!-!!-!!-!!-!-!!!!!!!!-!!-!-!!!!-!!!!-!!!!!-!!!-!!!-!-!-!!!!!!!!-!-!!!!-
HIL   28 ## 12 + 00000028 03 A 2359 0000 00050 999 999 D + 00000014 A 2359

      19  20 21  22       23        24        25       26 2728
       !!!-!!!-!!-!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!-!-!!
007 014 05      28.0000000                        999
```

Key to Table:

Note: You may enter up to 200 active lines per group ID in this table.

- **Column 1**: Group ID for group of definitions. In the above example, there is one group of definitions. You may create as many groups as you want. By entering the Group ID in column 7 of the `tab_sub_library` table, you can have a number of different sublibraries use the same group of definitions. Be sure that you fill in the tab16 table together with the `tab_sub_library` table.

  Item status "99" is used for defining the maximum overall number of loans/holds/bookings for a particular patron status in a sublibrary or for all patron statuses if you use ## in the patron status column. It overrides any other maximum number defined for the patron for a particular item status.

  Item status "98" is used for defining the maximum overall number of loans/holds/bookings for a particular patron status in a group of sublibraries or for all patron statuses if you use ## in the patron status column. The total number of loans/holds/bookings in all the sublibraries that belong to the same group as the sublibrary from which the item is being loaned/requested/booked must not exceed the limit set in the “98” line.

  Item statuses "99" and “98” must be repeated for every group ID or for every patron status within the group ID if there are differences. They must be the last line in the group (for example, GEN).

- **Column 2**: Item status code. Enter here one of the item status codes defined in one of the groups of `tab15`. Enter ## for ALL item status codes.

- **Column 3**: Item process status code. Enter here one of the item process status codes defined in `tab15`. Enter ## for ALL.

- **Column 4**: Patron status code. Enter here one of the patron status codes defined in `tab31`. Enter ## for ALL.

- **Column 5**: Method of computing due date. If you want to specify an actual due date, enter *A*. To define the due date by adding a specific number of days to the date on which the item is loaned, enter +.

- **Column 6**: Due date. If you entered *A* in column 5, then enter a specific due date here in the format yyyymmdd. If you entered + in column 5, then enter the number of days to be added to the loan date. You must enter 8 digits. For example, to add 2 days to the loan date, enter *00000002*. If you want the item to be returned on the same day it is borrowed, enter *00000000* (8 zeros). Short loan items also need to be defined with *00000000* (8 zeros). The maximum you can add is 999 days. For any number higher than that, digits representing thousands and higher will be disregarded; that is, if you enter 3650 days, the value taken will be 650.

- **Column 7**: Grace days for late return. Type the number of days you will allow the item to be late before the patron is fined. Note that you may use a combination of grace hours and grace days. To enter only grace hours, enter *00* in the days field.

- **Column 8**: Method of computing due hour. To specify an actual due hour, enter *A*. If you want to define the due hour by adding a specific number of hours to the hour on which the item is loaned, enter +.

- **Column 9**: Due hour. Enter an actual hour (according to a 24-hour clock) or to add a specific number of hours to the current hour, according to the definition in Column 8. To add a specific number of hours to the current hour, you must enter 4 digits. For example, if you want to add 5 hours, enter *0500*.

- **Column 10**: Grace hours and minutes for late return. Type the number of hours you will allow the item to be late before the patron is fined. Note that you may use a combination of grace hours/minutes and grace days. If you want to enter only grace days, enter *0000* in the hours field.

- **Column 11**: Fine rate. Fines are given per day overdue. The total fee is the fine rate multiplied by the number of days overdue.

  The fine rate has 5 digits and should be entered as though a decimal point exists before the last two digits. For example, *00025* means the days should be multiplied by *0.25* (for example, 25 cents).

  Note: Column 11 works in conjunction with column 14 of this table. Column 14 sets whether the fine is calculated per day or per hour, and whether the time that the library is closed is taken into account when the fine is calculated. If you want an hourly fine, the sum registered here must be the amount per hour x 24 (no. of hours in a day). For example, if you want to charge 0.50 per hour, you must set the amount in this column to 01200.

If you want a fine by minutes, the sum registered here must be the amount per minute x 1440 (no. of minutes in a day). For example, if you want to charge 0.10 per minute, you must set the amount in this column to 14400.

- **Column 12**: Maximum number of items. Enter the maximum number of items that may be borrowed by this combination of item status and patron status. Note that you must enter two digits, for example, *05* for 5 items.

- **Column 13**: Maximum number of hold requests. Enter the maximum number of hold requests that may be made by this combination of item status and patron status. Note that you must enter two digits, for example, *05* for 5 hold requests.

- **Column 14**: Fine method. This column works in conjunction with Column 11. Use one of the following values:

    - **0 - No fine charged**

    - **A - Open minutes** - The fine is calculated per minute overdue, and does not include the minutes that the library was closed.

    - **1 - Open hours** - The fine is calculated per hour overdue, and does not include the hours that the library was closed. If the patron returns an item one minute late, he will be charged for being one entire hour late.

    - **2 - Open days** - The fine is calculated per day overdue, and does not include the days that the library was closed. If the patron returns an item one minute late, he will be charged for being one entire day late.

    - **B - Total minutes** - includes the minutes that the library was closed, for example, if a patron returns a book 30 minutes after the due hour, he is fined for 30 minutes, even if the library has been closed for part of the period in which the loan became overdue.

    - **R - First day fixed, then open days**. This method levies a fixed fine of 2.00 for the first day and then continues to calculate like fine method 2 for all further days.

    - **3 - Total hours** - includes the hours that the library was closed. If the patron returns an item one minute late, he will be charged for being one entire hour late. If the patron returns an item that was due Friday on the

subsequent Monday, and the library was closed Saturday and Sunday, the patron is nevertheless charged for Sat.-Sun.-Mon. If at 9 a.m. a patron returns a book that was due the previous evening at 5 p.m., he is charged for all the hours in that period, even if the library was closed.

- **4 - Total days -** includes the days that the library was closed. If the patron returns an item one minute late, he will be charged for being one entire day late. If the patron returns an item that was due Friday on the subsequent Monday, and the library was closed Saturday and Sunday, the patron is nevertheless charged for Sat.-Sun.-Mon.

- **5 - Overlapping blocking date** - This option blocks a patron from borrowing any items, without charging him a fee for overdue items, blocking him for the longest overdue period when a group of books is returned at the same time. For example, if the patron returned three books late and one book was 3 days late, one book was 7 days late, and one book was 14 days late, you can have the system block the patron from borrowing additional items for a period of **14** days. This is "overlapping" blocking. The longest overdue period is used to determine how long the patron is blocked. The block starts when the items are returned. The number of days that should be blocked because of each late return can be multiplied by a value set in the tab100 BLOCK-RATION variable. For example, if a 14 day block is calculated, setting the tab100 variable BLOCK-RATION to 2 will cause the block to be 28 days long.

- **6 - Cumulative blocking date** - This option blocks a patron from borrowing any items, without charging him a fee for overdue items, for a time which matches the total amount of time of all overdue books. For example, if the patron returned three books late and one book was 3 days late, one book was 7 days late, and one book was 14 days late, the system blocks the patron from borrowing additional items for a period of **24** days. This is "cumulative" blocking. The overdue periods are added together. The block starts when the items are returned. The number of days that should be blocked because of each late return can be multiplied by a value set in the tab100 BLOCK-RATION variable. For example, if a 14 day block is calculated, setting the tab100 variable BLOCK-RATION to 2 will cause the block to be 28 days long.

- **7 - Include recall** - This option calculates overdue fines taking into consideration the fact that the item has been recalled. Two debit transactions are created:

    1. A late return fee - calculated according to method 4 (total days).

    2. A multiplication of the fine with a recall/rush recall factor that is defined in tab18.lng (cash transaction codes 50 and 51).

- **C** - This option calculates overdue fines taking into account the fact that the item has been recalled. Two debit transactions are created:

    1. A late return fine calculated according to method 1 (the library's open hours).

    2. A multiplication of the fine with a recall/rush recall factor that is defined in tab18.lng (cash transaction codes 50 and 51).

- **D** - Similar to method C except that the basic fine is calculated in relation to method 2 (the library's open days).

- **E** - Similar to method C except that the basic fine is calculated in relation to method 3 (total hours - open and closed).

- **F** - Total days by intervals - This method uses the intervals set up in the new `tab_late_return` table to calculate fines. The sums set up in this table are treated as an actual amount and not as a ratio per day.

- **8** - This option combines fine 5 and a monetary fine according to the value set in col. 11. The patron is blocked for a certain period and has to pay a fine before he can effect a transaction again. The fine is calculated according to total days.

- **9** - This option combines fine 6 and a monetary fine according to the value set in col. 11. The patron is blocked for a certain period and has to pay a fine before he can effect a transaction again. The fine is calculated according to total days.

- **Columns 15-18**: These four columns allow you to define alternate due dates/hours for items that have hold requests. Now, if a patron checks out an item that has more hold requests on it than the number defined in column 28 of this row, the due date/hour that will be assigned will be the alternate one, not the original one. If no alternate due date/hour is defined in this table, the original one will be used. The system may be configured to display a message when using the alternate due date by setting column 4 of message 1020 in $aleph_root/error_lng/pc_messages to Y.

- **Column 15**: Method of computing due date (for items with hold requests). To specify an actual due date, enter *A*. To define the due date by adding a specific number of days to the date on which the item is loaned, enter +.

- **Column 16**: Due date (for items with hold requests). If you entered *A* in column 15, then enter a specific due date here in the format yyyy/mm/dd . If you entered + in column 15, then enter the number of days to be added to the loan date. You must enter 8 digits. For example, if you want to add 2 days to the loan date, enter *00000002*. If you want the item to be returned on the same day it is borrowed, enter *00000000* (8 zeroes).

- **Column 17**: Method of computing due hour (for items with hold requests). To specify an actual due hour, enter *A*. If you want to define the due hour by adding a specific number of hours to the hour on which the item is loaned, enter +.

- **Column 18**: Due hour (for items with hold requests). You may choose to enter an actual hour (according to a 24-hour clock) or to add a specific number of hours to the current hour, according to the definitions in Column 17. To add a specific number of hours to the current hour, enter 4 digits. For example, to add 5 hours, enter *0500*.

- **Column 19**: Recall period. This is the period in days that the library gives the patron for the return of a recalled item (assuming the minimum guaranteed loan period (column 20) has already passed or it is a rush recall). This period should be calculated according to the estimated time it would take the letter to reach the patron and the time it would take the patron to return the item to the library.

- **Column 20**: Minimum guaranteed loan period. This is the period in days that the original patron is allowed to have the item on loan, even if the item has been requested by another patron and recalled. This period is ignored when the recall is a rush recall.

- **Column 21**: Maximum number of renewals. Enter a value from *00* to *09*. Renewals can be limited up to 8; enter *09* to define that renewals

are unlimited. To limit renewals by period rather than by number, enter 09 in this column, and use column 22.

- **Column 22**: Total loan period, including all renewals and the original loan period. Enter *D* for days, *W* for weeks or *M* for months, then enter the number of permitted days, weeks or months. You must enter 3 digits for the number. For example, for 3 days, enter *D003*.

- **Column 23**: Maximum fine. Enter up to two decimal places. The field should be left-aligned.

- **Column 24**: Minimum fine. Enter up to two decimal places. Fines below this amount will be treated as zero amount fines. The field should be left-aligned.

- **Column 25**: Fixed amount to be added to any fine. Enter up to two decimal places. The field should be left-aligned.

- **Column 27**: Adjust Due Date and Hour. Enter one digit. Use one of the following values when calculating the due date:

    - **0** - Adjust a due hour which is after closing time to the closing time (if the due date is an open day), or to the closing time of the *next* open day (if the due date is a closed day).

    - **1** - Adjust a due hour which is after closing time to the closing time (if the due date is an open day), or to the closing time of the *previous* open day (if the due date is a closed day).

    - **2** - Do *not* adjust due hour if the due date is an open day; if the due date is a closed day, find the *next* open day. This is the default value in the event that `tab16` cannot be read or if this column has an invalid value (that is, other than 0, 1, 2 or 3).

    - 3 - Adjust a due hour which is after closing time to the *next* open day's opening hour.

  The column is sensitive to sublibrary and item status.

- **Column 28:** Item requested threshold. If an item has this number of active hold requests, it will be considered requested for the purpose of using columns 15-18 for setting its due date. The system may be configured to display a message when using the alternate due date by setting column 4 of message 1020 in $aleph_root/error_lng/pc_messages to Y.

Step 3: Define Library Hours

Edit `tab17`. This table enables the system to set new due dates for items that would otherwise be due on days or at hours when the library is closed.

Note that there is an easy-to-use interface in the Circulation GUI for the editing of this table - the Library Open Hours Management Interface which is accessible from the Circulation GUI's Loan Tab. Below is an explanation of how to define the library open hours by editing the tab17 table directly.

In the following example, specific days on which the library is closed appear before the general definition of the library's opening hours:

```
! 1       2    3  4   5   6                  7
!!!!!-!!!!!!!!-!!-!-!!!!-!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
17A   20060501    C           Comment - May First
17A   ####06##    O 1000 1300 Comment - summer vacation
17A   ####07##    O 1000 1300
17A   ####08##    O 1000 1300
17A   ####09##    O 1000 1300
```

Key to Table:

- **Column 1**: Group ID for group of definitions. In the above example, there is one group of definitions. You may create as many groups as you want. By entering the Group ID in column 8 of the `tab_sub_library` table, you can have a number of different sublibraries use the same group of definitions. Be sure that you fill in the tab17 table together with the `tab_sub_library` table.

- **Column 2**: The dates are written in the format yyyymmdd. For example 19990714. You may use the # character as a wildcard for dates. For example, ####0714 indicates July 14th of any year. ######## indicates any date.

- **Column 3**: Day of the week. Sun=00, Sat=06.

- **Column 4**: Enter C for Closed; O for Open.

- **Column 5**: Fill in this column if you entered "O" in column 4. Column 5 is the hour that the library opens. The system uses a 24-hour clock, so if the library opens at 9 AM, enter 0900.

- **Column 6**: Fill in this column if you entered "O" in column 4. Column 6 is the Hour that the library closes. The system uses a 24-hour clock, so if the library closes at 5 PM, enter 1700.

  Note that a library open hours may be configured so that closing time is after midnight. In that case, the closing hour will be configured as the real hour +24. For example, if the library opens on Friday morning at 9 A.M. and closes at 2.A.M the following day, the tab17 line will be configured as:

  ```
  17A   ######## 05 O 0900 2600
  ```

Step 4: Deciding Whether or Not the Patron's Expiration Date Influences a Due Date

In `tab100`, you can set a variable to determine if a patron's expiration date shortens a loan period if the due date is later than the patron's expiration date.

The line is:

```
BOR-EXPIRY-DUE-DATE
```

and the options are Y/N/H.

Step 5: Deciding if the Library's Closing Hour will Adjust the Due Hour
The due hour can be changed according to the library's closing hour. This is determined by `tab16`, column 27 'Adjust Due Date and Hour', which has four possible parameters.

For example, the library's working hours are 8:00 - 17:00 and the due hour is 23:59.

- If the variable is set to 0, the due hour will be set to 17:00, or to 17:00 the next open day (if the due date is a closed day).
- If the variable is set to 1, the due hour will be set to 17:00, or to 17:00 the previous open day (if the due date is a closed day).
- If the variable is set to 2, the due hour remains 23:59.
- If the variable is set to 3, the due hour will be set to 8:00 of the next open day.

Step 6: Rounding and Factoring Fines
The system provides the option to round up/down the fine amount. In addition, you can 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 (for example, 10,000 lira).

Following are the relevant parameters in the `aleph_start` table:

```
setenv fine_factor_fix     000
setenv fine_rounding         Y
```

Step 7: Determine if "No Fine" Transactions will be included in the List of Cash Transactions

You can determine whether or not the system will display transactions for which no fine (zero amount due) was charged. To do this, edit tab18.lng.
enter Z in col. 5, So that "no fine" transactions will not be included in the List of Cash Transactions.

Step 8: Define Lost Material Charges
To define lost material charges, edit `tab34`. This table defines the charges to be added to the patron's cash record when an item is lost. (An item is defined as lost by the librarian using the GUI Circulation module, or automatically by the system (`tab32`) after the maximum number of overdue notices has been sent.)

Following is an example from the `tab34` table. (Note that the widths of columns 4, 6 and 9 have been shortened in this example.)

LIT	## ##	#	#	#####	#####	15.00	0.00	75.00
MED	## ##	#	#	#####	#####	22.00	30.00	120.00
MUS	## ##	#	QA45.62	#####	#####	0.00	75.00	75.00

WID	01	03	4	PE.25#	GEN	ISSBD	10.00	20.00	30.00

Key to Table:

- **Column 1**: Sublibrary. Use ##### for "all sublibraries."
- **Column 2**: Item status (from tab15). Use ## for all.
- **Column 3**: Patron status (from tab31). Use ## for all.
- **Column 4**: Identifier of the call number type. Use ## for all
- **Column 5**: Call Number. Enter a specific call number. Specifying a range is done by using a # mark to separate the 'From Call Number' from the 'To Call Number'. For example "ABC#DEF" means 'from call number ABC to call number DEF.

  For all call numbers, use only one # in the first position of the column.
- **Column 6**: Collection. Use ##### for all.
- **Column 7**: Item Material Type. Use ##### for all.
- **Column 8**: The charge for sending a notice. You may use up to two decimal places. You may enter, for example, *10*, *10.0*, or *10.00*.
- **Column 9**: Handling charges. You may use up to two decimal places. You may enter, for example, *10*, *10.0*, or *10.00*.
- **Column 10**: Fixed price for replacing the item. You may use up to two decimal places. You may enter, for example, *10*, *10.0*, or *10.00*.

  Column 10 can be set to use the value set in Z30-PRICE of the lost item. This is done by prefixing the fixed value with "I/". If Z30-PRICE is a decimal number other than zero, it will be used as the replacement cost. If not, the value in Column 10 will be used.
- **Column 11**: Not yet implemented (MARC price).
- **Column 12**: Not yet implemented (patron factor).

Note: When an item is registered in the GUI as "Lost", the librarian has the option to manually change the replacement price charged for the lost item.

Step 9:Define the Effective Due Date in Printouts and GUI HTML Information
In XML-XSL printouts and in GUI HTML information (based on templates defined in `./xxx50/pc_display_eng`), z36-due-date can be translated into the "effective due

date" by defining the following setting in a .trn file (`./xxx01/form_eng/global.trn` or a specific translation file):

```
!                                                  M
!                                                  /
!Tag                                    Type       S Parameters
!!!!!!!!!!!!!!!-!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
z36-due-date    DUE-DATE   S z36-recall-date z36-recall-due-date
```

The translation process uses the get_effective_due_date routine: the `tab100` variable RECALL-METHOD and the Z36 fields Z36-DUE-DATE, Z36-RECALL-DATE and Z36-RECALL-DUE-DATE are all taken into account in order to generate the effective due date.

7. Overdue Notice Schedule

In order to set the overdue notice schedule, edit `tab32`. Following is an example from the table:

```
! 1   2  3  4   5   6  7  8
!!!!!-!!-!!-!!-!!!-!!!-!-!!!!
##### 04 ## 00 003 001 O 0080
##### 04 ## 01 007 007 O 0081
##### 04 ## 02 014 014 O 0082
##### 04 ## 03 021 021 L
##### 04 ## 09 040 028 L
```

Key to Table:

- **Column 1**: Enter a sublibrary code or use ##### to indicate all sublibraries.

- **Column 2**: Enter an item status from tab15 (enter "##" for "all" item statuses).

- **Column 3**: Enter a patron status from tab31 (enter "##" for "all" patron statuses).

- **Column 4**: Current letter number in the loan transaction record z36.
  00 - 1st notice
  01 - 2nd notice
  02 - 3rd notice
  03 - 4th notice
  09 - for items which are "claimed returned". This code is used when b-cir-50 (Print Overdue and Lost Billing Notices) is run for "Bills for lost material that was claimed to have been returned"

- **Column 5**: Days overdue (for Notice ID=00) or Days since last letter (for other IDs).

- **Column 6**: Days overdue for requested items (for Notice ID=00) or Days since last letter (for other IDs).

- **Column 7**: Enter O if the item is considered Overdue; enter L for Lost.

- **Column 8**: Transaction number as listed in tab18 (Cash Management). The patron will be charged the fee defined for this transaction number in tab16.

Reading the Table
The table should be read in the following manner:

```
! 1   2  3  4   5   6  7  8
!!!!!-!!-!!-!!-!!!-!!!-!-!!!!
##### 04 ## 00 003 001 O 0080
##### 04 ## 01 007 007 O 0081
##### 04 ## 02 014 014 O 0082
##### 04 ## 03 021 021 L
##### 04 ## 09 040 028 L
```

Important note:
The overdue days calculation (column 5 or 6) does not include the due date or the last letter date, it starts with the day after.

For Notice ID=00: The first notice is sent when the item is three days overdue (or one day overdue if the item has a hold request for it). Three days overdue means that the first notice is sent on the forth day after the due date.

For Notice ID=01: The second notice is sent seven days after the first notice. This means that the system waits until seven days have passed since the first notice, then it sends the second notice on the eighth day.

For Notice ID=02: The third notice is sent 14 days after the second notice. This means that the system waits until 14 days have passed since the second notice is sent, then it sends the third notice on the 15th day.

For Notice ID=03: The forth notice is sent 21 days after the third notice, and then the item is declared lost.
This means that the system waits until 21 days have passed since the third notice, then it send the forth notice on the 22nd day.

For Notice ID-09: An item that was set as "claimed returned" is declared lost 40 days after it was set as "claimed returned".
This means that the system waits until 40 days have passed since the item was set as “claimed returned”, then it sends a bill for lost material on the 41st day.

7.1. Overdue Notice Letters and Summary

You can determine whether the notices sent to the patron will always use the same template (overdue-letter-0 or overdue-summary-0) or use different templates for each notice, as follows:

- 1st notice: overdue-letter-0-nn or overdue-summary-0-nn
- 2nd notice: overdue-letter-1-nn or overdue-summary-1-nn
- 3rd notice: overdue-letter-2-nn or overdue-summary-2-nn
- 4th notice: overdue-letter-3-nn or overdue-summary-3-nn
- 5th notice: overdue-letter-4-nn or overdue-summary-4-nn

The notices usually become progressively more demanding.

To define which templates will be used, edit the OVERDUE-LETTER-STYLE variable in tab100. If you want the same template to be used for each notice, enter N. If you want different templates to be used, enter Y.

If the loan has been made by a proxy, then a second notice letter is printed for the proxy in addition to the initial notice letter that is printed for the sponsor.

Overdue Notice (single item per notice) - filenames: overdue-letter-0, overdue-letter-0-nn, overdue-letter-1-nn, overdue-letter-2-nn, overdue-letter-3-nn, overdue-letter-4-nn. "0" is first notice, "1" is second notice, and so on.

Overdue Summary - filenames: overdue-summary-0, overdue-summary-0-nn, overdue-summary-1-nn, overdue-summary-2-nn, overdue-summary-3-nn, overdue-summary-4-nn

8. Patron Blocks

Define blocks and their effect on patron privileges. To do so, edit the `tab_delinq.lng` table. Following is a sample from the table:

```
!  2   3          4          5           6
!           abcdefghi
!!-!-!!!!!-!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>

00 1                 L Patron has no global blocks
01 1                 L Patron consistently returns books late
02 1                 L Disruptive behavior in library
03 1                 L Personal item left in library - contact Circ
desk
04 1                 L Too many Claimed Returns
53 1                 L Patron to contact academic adviser
!
00 2 #####           L Patron has no local blocks
01 2 ##### N    NN   L Patron is blocked from placing a hold request.
02 2 ##### NN   NN   L Patron is not allowed to place hold or
photocopy re
quest.
03 2 ##### N N   N   L Patron's override permission is blocked.
04 2 ##### N  N  N   L Patron cannot place more than one request on
this i
tem.
```

```
05 2 ##### NNNNNNNNN L All patron's privileges are blocked.
50 2 ##### NNNNN NN  L New patron self registration - Web OPAC.
```

Key to Table:

- **Column 1**: Block Code. Enter a code from *01* to *99*. You may have up to 99 block codes for global patrons and another 99 block codes for local patrons.

- **Column 2** indicates whether the block code is for a Global Patron (1) (Z303 record) or a Local Patron (2) (Z305 record).

- **Column 3**: Sublibrary. Enter a Sublibrary only if the block code is for "Patrons". You may use the # character as a wildcard.

- **Column 4**: You may limit the patron's privileges if he has been blocked. In order to block a specific privilege, enter *N* in the appropriate column. Otherwise leave it blank. The codes for the privileges are as follows:

    - a - Loan
    - b - Photocopy requests
    - c - Override
    - d - Multiple hold requests
    - e - Check transactions
    - f - Hold request
    - g - Renewal
    - h - Reading room permission
    - i – Booking Request

    Each privilege is explained in detail in the online help for the Local Patron Record in the GUI.

- **Column 5** is the code for the character set. It should be set to L, unless you are using CJK.

- **Column 6** is the description of the block as it will appear in the GUI.

9. Cash Transactions

9.1. Setup of Cash Transactions (tab18.lng)

You can define various transactions for which a patron is charged a fee. To do this, edit tab18.lng.

Two types of transactions can be defined in tab18.lng:

- System-generated Transactions: ID numbers 0000 - 1028
- Librarian-generated Transactions: transactions which are only generated by the librarian, using the New Cash Transaction form (accessible from the Patron Information window / Cash Transactions tab / New button).
  You can define your own librarian-generated transactions by adding a new line with an unused ID number.

Note that charges for the following letters are dependent on the setting of column 6: ILL material arrival, Hold request filled, Lost material - Notice, Overdue letter and Overdue summary letter. If column 6 has the value **N**, no charges will be levied. This can be used by the library to prevent charging for postal charges when the letter is sent by e-mail.

Following is a sample from the table:

```
1     2   3  4  5     6          7      8  9
!!!!-!!!!!-!!-!!-!-!!!!!!!!!!-!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
0000 ##### ## ## Y            10.00        General
0001 ##### ## ## Y 1.00                     Photocopy request
0002 ##### ## ## Y 10.00     12.00        Hold request
0003 ##### ## ## Y                        Late return
0040 ##### ## ## Z                        Lost material - Handling
```

Key to Table:

- **Column 1**: ID number of the transaction.
- **Column 2**: Sublibrary. For "all sublibraries," use ##### . (Be sure to define a catch-all line that has ##### in column 2 together with ## in column 3.)
- **Column 3**: Item status. For "all items", use ##.
- **Column 4**: Patron status. For "all patrons", use ## . (Be sure to define a catch-all line that has ##### in column 2 together with ## in column 3.)
- **Column 5**: Switch to activate/deactivate cash control. If you enter *Y*, patrons will be charged for the transaction according to the fee entered in column 5, if the setup of column 6 does not prevent this. If you enter

N, patrons will not be charged for the transaction. Very similar to *Y*, the *Z* parameter also charges for cash transactions, but only when the cash rate does not equal zero. Note that the Z parameter affects only new transactions.

Note: If column 5 is set to *Y* for Late Return fines, the rate will be determined according to the fine rate entered in column 11 of tab16, and column 5 will be ignored. See the Late Return line in the example above. The same applies to lost material charges that are calculated according to tab34.

- **Column 6**: Fine. Enter the amount in your currency that you want to charge the patron.

  Note: If column 4 is set to *Y* for Late Return fines, the rate will be determined according to the fine rate entered in column 11 of `tab16`, and column 5 will be ignored (see the Late Return line in the example above). The same happens with lost material charges that are calculated according to `tab34`. If tab`34` is not present or if there is no value in tab34, the sum is taken from `tab16.lng`.

  In some lines, the value in column 5 functions as a ratio multiplicator. The actual amount is set in other tables. This applies to the following lines:

  - 0003,0014 - late return fee, dependent on tab16
  - 0012,0040,0041,0042 - lost material, dependent on tab34
  - 0015 - ILL material arrival, dependent on tab_ill_charge
  - 0050,0051 - ratio, additional daily amount or a combination of both charged for late return of recalled / rush-recalled loan. The tab100 variable OVERDUE-RECALL-RATIO defines whether the number is a ratio, a set amount or a combination of both.
  - 0052,0053 - ratio, additional daily amount or a combination of both for a fine limit for late return of recalled/rush-recalled loan. The tab100 variable OVERDUE-RECALL-RATIO defines whether the number is a ratio, a set amount or a combination of both.
  - 0054,0055 - ratio, additional daily amount or a combination of both for a lost letter for recalled/rush-recalled loan. The tab100 variable OVERDUE-RECALL-RATIO defines whether the number is a ratio, a set amount or a combination of both.

- **Column 7** 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 6, 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.

- **Column 8** is a switch for charging/not charging if the patron's valid address contains an e-mail address. This is used for preventing the levying of postal charges if the letter is sent by e-mail.

- **Column 9** is the name of the cash transaction as it will appear in the GUI.

Y/N Ratio
For some lines the value 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. This is so for:

- 0003, 0014 - late return fee, dependent on tab16
- 0040, 0041, 0042, 0054, 0055 - lost material, dependent on tab34
- 0015 - ILL material arrival, dependent on `tab_ill_charge`
- 0050, 0051 - ratio for late return of recalled loan

Alphanumeric Paper Formats
There are some lines, A4-6, A3-6, B2-6, LE-6 where the ID number of the transaction (Column 1) is not purely numeric. These are used to register photocopy requests for specific paper formats.

You can add other paper formats according to your needs and add an appropriate line. For example, when LE-6 was added for 81/2"x11" Letter paper formats to be processed, a new line was added to tab 18:

```
LE-6 #####  ##  Y 0.20       L Photocopy processing (Letter)
```

The ID 0006 will be used as a default if the special entries are missing.

9.2. Registration of Cash Transactions in Offline Circulation

Whether or not offline circulation registers cash transactions depends on the variable OFFLINE-CASH-TRANS in tab100. Set to Y for the system to register cash

transactions in offline circulation; set to N for the system not to register cash transactions in offline circulation.

10. Circulation Transaction Checks (tab_check_circ)

`tab_check_circ` sets the checks that will be performed when an item is loaned or renewed.

```
1          2
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
LOAN-GBL   check_circ_1_a
LOAN-GBL   check_circ_1_d
LOAN-GBL   check_circ_9_a                tab_block_circ_alternative
RENEW-GBL  check_circ_1_a
RENEW-GBL  check_circ_1_b
RENEW-GBL  check_circ_1_c
RENEW-GBL  check_circ_1_e
RENEW-GBL  check_circ_9_a
```

Column 1 defines when a check is performed:

1. **B-CIR-77** - Update patron records (p_cir_77)
2. **LOAN-GBL** - checks patron prior to loan and and when "Check" is performed on the Global Patron Information window
3. **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 sublibrary of the item to find the z305 patron record. Used with check_circ_2_d.
4. **LOAN** - item is being loaned to a patron
5. **RENEW** - item is being renewed to a patron applies to Renew and Renew all functionality
6. **RENEW-WWW** - patron renewing loan in www
7. **RENEW-GBL** - checks on patron after the **Renew All** (for renewing all loans) or the **Renew Marked** (for renewing more than one selected loan at a time) buttons from the Circulation GUI's Loan List have been clicked.
8. **B-CIR-10** - item renewed to a patron (p_cir_10)
9. **B-CIR-08** - item renewed to a patron (p_cir_08)
10. **ROUT-LOAN** - item is being loaned to a routing member
11. **EXTEND** - due date/hour is being changed from the List of Loans or during a loaning activity

Column 2 defines which check procedure will be performed: The available checks are:

Check	Procedure	Action	$aleph_root/error_lng/check_circ
1_a	patron blocks Z303	Checks patron blocks on Z303 record	0101, 0102
1_b	patron blocks Z305	Checks patron blocks on Z305 record	0111, 0112, 0113, 0115
1_c_a	patron expiration date + renew warning	Checks z305_expiry_date Renew warning if the expiration date has already passed	0121
1_c_b	patron expiration date + renew warning	Checks z305_expiry_date Renew warning for approaching expiration date according to parameters set in col.3 tab30	0122
1_d	overdue check	Works according to the variable OVERDUE-LETTER-NO which determines when an item is "overdue" (i.e., the due date has passed and N (0/1/2/3/4) notices have been sent)	0131
1_e	block date check	checks z305_end_block_date	0141
1_f	proxy's expiration date check	checks z305_expiry_date	0261
2_a	patron loan permission	checks z305_loan_permission	0151
2_b	patron renew permission	checks z305_renew_permission	0152
2_c	patron reading room permission	checks z305_rr_permission, taking the z305 that matches the reading room sublibrary at which the loan is being performed sublibrary (i.e. when item is loaned at a reading room station, type 4 in tab_attr_sub_library)	0153
2_d	loan and reading room permission	check patron loan and reading room permission using the z305 that matches the original sublibrary of the item (i.e. the sublibrary of the item before it was assigned the RR sublibrary)	0154

Check	Procedure	Action	$aleph_root/error_lng/check_circ
2_e	Handling magnetic media	Checks if a patron tries to loan/renew/return a magnetic media (tab25.lng, col. 4 = Y).	155
3	patron renew limit	check the patron's renew limits according to parameters set in tab16 (col.21)	0161, 0162
4_a	patron loan limits	check the patron's loan limits according to parameters set in tab16 (col.12)	0171, 0172
4_b	patron loan limits (99 line)	check the patron's overall loan limit in a sublibrary according to parameters set in tab16 99 line	0181
4_c	patron loan limits (99 line)	check the patron's overall total loan limits for ADM library according to parameters set in tab16 99 line	0182
4_d	patron loan limits (98 line)	check the patron's overall total loan limits for the sublibrary's group according to parameters set in the tab16 98 line. The entire group (all libraries that share a column 7 of tab_sub_library value) may not exceed the number of loans set up in the 98 line	0183
5	cash limit	checks if the patron has exceeded the amount allowed in z305_cash_limit (including open and transferred cash transactions) This check retrieves list of charges to include/exclude. For example: CHARGES=-0001,0006,0008,0009 This means that all charges except for 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted. If the CHARGES parameter is not retrieved, all cash charges are taken into account.	0191

Check	Procedure	Action	$aleph_root/error_lng/check_circ
5_b	cash limit	checks if the patron has exceeded the amount allowed in z305_cash_limit (including open cash transactions only) This check retrieves a list of charges to include/exclude. For example: CHARGES=-0001,0006,0008,0009 This means that all charges except for 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted. If the CHARGES parameter is not retrieved, all cash charges are taken into account.	0191
6	item requests	checks whether the item has hold requests placed on it	0201
6_a	item requests	checks whether the item has hold requests placed on it	0201
6_b	Item hold queue	Checks whether the loaning patron is the first in the hold queue.	202
7_a	item loan permission	checks whether the item can be loaned, column 6 in tab15.lng	0211, 0212
7_b	item renew permission	checks whether a loan on this item can be renewed, column 7 in tab15.lng	0211, 0213
7_c	reloaning limit	checks whether the same patron did not return the exact item recently, as per column 15 in tab15.lng	0214
8_a	item - lost/claimed return	checks if the item was declared as lost or claimed to have been returned	0221, 0222
8_b	renew - overdue letter	checks if an overdue letter has been printed for a loaned item	0231

Check	Procedure	Action	$aleph_root/error_lng/check_circ
9_a	overdues and fines - ADM library	checks maximum overdues, recalled overdues and fines, for ADM library based on tab_block_circ or any other table defined in column 3 of tab_check_circ This check retrieves two parameters: • block circ table name. If not specified, the default table, tab_block_circ, is consulted. • list of charges to include/exclude For example: CHARGES=-0001,0006,0008,0009 This means that all charges except for 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted. If the CHARGES parameter is not retrieved, all cash charges are taken into account. The two parameters should be separated by a semicolon (;). For example: tab_block_test;CHARGES=+0001,0006,0008,0009	0241, 0242 0243
9_b	overdues and fines - sublibrary	checks maximum overdues and recalled overdues in a sublibrary based on tab_block_circ or any other table defined in column 3 of tab_check_circ	0251, 0252

Check	Procedure	Action	$aleph_root/error_lng/check_circ
10_a	overdues and fines - ADM library	checks maximum overdues, recalled overdues, and fines, for regular, lost, and claimed returned loans, for ADM library based on tab_block_circ or any other table defined in column 3 of tab_check_circ This check retrieves two parameters: • block circ table name. If not specified, the default table, tab_block_circ, is consulted. • list of charges to include/exclude. For example: CHARGES=-0001,0006,0008,0009 This means that all charges except for 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted. If the CHARGES parameter is not retrieved, all cash charges are taken into account. The two parameters should be separated by a semicolon (;). For example: tab_block_test;CHARGES=+0001,0006,0008,0009	0241, 0242 0243
10_b	overdues and fines - sublibrary	checks maximum overdues and recalled overdues, for regular, lost and claimed returned loans in a sublibrary based on tab_block_circ or any other table defined in column 3 of tab_check_circ	0251, 0252
12_a	ILL patron identity	checks if there is a match between the original ILL patron and the patron who is trying to loan the ILL item	0801
13_a	Renew ILL item	Checks if the renewed items are ILL items (items that have been created as a result of receiving borrowing request from supplier). If so, the renew action is enabled via Circ-Loans List only for authorized ILL staff user. If this check is not defined, then ILL items will be renewed like any other item.	803
14_a	Check loan of like item.	Checks whether the patron that is making the loan already has a like item on loan.	291

Check	Procedure	Action	$aleph_root/error_lng/check_circ
14_b	Check loan of similar item.	Checks if the patron making the loan already has a similar item on loan. Similar items are items that has the same enumeration/chronology and are of the same material type. Items in all sublibraries, collections, and statuses are considered similar. This check is the same as 14_a, only the definitions for Like Copy in tab100 (SUB-LIBRARY-DIVISION, HOLD-REQUEST-ITM-STATUS, HOLDREQ-PROCESS-STATUS, and HOLD-REQUEST-COLLECTION) are not taken into account.	292

The text of messages associated with the checks is defined in `$aleph_root/error_lng/check_circ`. The minimum circulation user level to override them is defined in the `check_circ_override` table in the ADM library.

Alternative blocks are useful when you need to define a separate set of blocks for different activities. They can be set for any of the activities defined in Column 1. Alternative blocks are defined in a user-defined alternative table, or in the default alternative table `tab_block_circ`.
Column 3 defines the parameters for the check routine

- The alternative table that is consulted for checks 9_a, 9_b, 10_a, and 10_b. This alternative table must be present in the data tab of the ADM library. If not specified, the default table `tab_block_circ` is used.
- The list of cash charges to include/exclude for checks 5, 5_b9_a, 9_b ,and 10_a. If the CHARGES parameter is not retrieved, all cash charges are taken into account.

11. Hold and Photocopy Requests

In order to set the management policy for hold and photocopy requests, you must make decisions regarding such factors as which checks the system will perform before/after a request is made; which pickup/delivery location will be used for each combination of sublibrary, item status and patron status; the amount of time an item may be kept on the hold shelf; the approach that will be used for printing call slips; and so on. Each factor is discussed below.

11.1. Hold Request Checks

It is up to you to determine which checks the system will perform before/after a hold request is placed. For example, the system may check whether or not the item can be requested at all or may check whether or not the item is on loan, in which case it will display information on its status (lost/recalled/on loan until). To set up the checks, edit `tab_hold_request` in the ADM library's `tab` directory.

```
! 1                     2
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
PRE         check_hold_request_a
PRE         check_hold_request_b
PRE         check_hold_request_c

POST        check_hold_request_d
POST        check_hold_request_e
POST        check_hold_request_f

CIRC        check_hold_request_f
CIRC        check_hold_request_g
CIRC        check_hold_request_h

INFO        check_hold_request_k
```

This table defines checks for hold requests. 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.

The following separate instances have been defined when checks are made:

- AB-POST is used when submitting an advance booking request from the OPAC or from the GUI
- AB-PRE is used when the list of items is displayed in the Web OPAC. Passing this check is required to show the 'request' link next to advance booking items
- BK-CIRC is used in the Circulation client
- BK-POST is used when the booking request fill-in form is displayed
- BK-PRE is used for the 'Booking' link when the list of items is displayed in the Web OPAC
- CIRC is used in the Circulation client
- CIRC-INFO is used in the Circulation client. The errors reported by this instance can always be overridden
- FWD-TITLE is used when the active stop in a title request is changed. Queue stops whose items do not pass FWD-TITLE checks will be considered not available.
- HL-GRP is used for title level hold requests ('BIB Request'). Only items that pass the check are regarded as available.
- HL-GRP-PRE is used for title level hold requests ('BIB Request'). This check is performed on each ADM that has items that are related to the selected BIB record. If the ADM does not pass the check, its items will not be included in a group.
- INFO is used when the hold request fill-in form is displayed

- POST is used when a request is sent to the server
- PRE is used when the list of items is displayed in the Web OPAC
- REALLOC-AB is used when an advance booking request is reallocated to another like item
- REALLOC-BK is used when a booking request is reallocated to another like item
- TITLE-REQ is used for title level
- XML-CIRC is used when a hold request is created from an X-Service
- UE06-PREF and UE06-LIKE are used by the ue_06 request handling daemon. They are relevant only when the preferred items mechanism is applied and are used in order to define whether a preferred item is available or not.

  UE06-LIKE is used when a like item of the original requested item is checked. UE06-PREF is used when an item that is not a like item is checked.

  Check routines that check hold request limits (for example, check_hold_request_h) should be defined only in UE06-PREF and not in UE06-LIKE.

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.

When setting which checks to use, take into account the system resources required for checks that are made when the items list is displayed (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. If a check is listed for PRE, it need not be listed for INFO or POST, since the INFO/POST scenario will not happen. Checks that relate to the patron usually appear in POST.

Check	**Action**	**$aleph_root/error_lng/ check_hold_request**	**Suggested use**
a	if tab15 col.8=N, item cannot be requested	1100	PRE,POST, CIRC
b	if sublibrary and item status are not listed in tab37, item cannot be requested.	1105	PRE,POST, CIRC
c	if tab15 col.12=Y, the item can have only one request	1120	PRE,POST, CIRC
d	z305 hold-permission	1110	PRE,POST, CIRC
d1	checks if the item is unavailable (requested or loaned to any patron)	1185	PRE, POST, CIRC
e	if combination of sublibrary, item status and patron status are not listed in tab37, item cannot be requested	1115	POST, CIRC
f	z305 multi-hold allowed	1125	POST, CIRC

Check	Action	$aleph_root/error_lng/check_hold_request	Suggested use
g	z305 hold-on-shelf (item availability)	1130	POST, CIRC
h	tab16 col.13, maximum number of hold requests	1135	POST, CIRC
h_1	tab16 col.13, maximum number of hold requests This routine is similar to check_hold_request_h, but also includes in the check existing loans that were originally hold requests.	1135	POST, CIRC
h_2	checks if the maximum number of hold requests has been reached. The maximum number is received as a parameter.	1350	POST, CIRC
i	if the item is on loan, displays due date, due date that will be active after recall action is performed, and information on whether lost or recalled	6000-6002,6020	INFO
j	if item has requests, displays on hold, patron's place in queue	6005-6006	INFO
k	tab15 col.8 value (Y,N,C)	6010-6012	INFO
k1	Z30-description value	6016	INFO
l	patron-id has not been entered	1140	PRE
m	patron has item on loan and is attempting to request a like copy	1150	POST, CIRC
n	patron tries to reserve an item that has been declared as "Lost" or "Claimed Returned."	1095	POST, CIRC
o	checks if there is a "like" copy available (same year,volume, part, issue AND depending on tab100 setup, same sublibrary, collection, status, and process status).	1150	INFO
p	checks if all "like" 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)	1155	POST, CIRC
q	tab16 col.13 check for maximum number of hold requests for the sublibrary ("99" line)	1160	POST, CIRC
q_1	tab16 col. 13 check for maximum number of hold requests for the sublibrary (line 99) This routine is similar to check_hold_request_q, but also includes in the check existing loans that were originally hold requests.	1160	POST, CIRC
r	patron expiration date has been reached	1165	POST, CIRC

Check	Action	$aleph_root/error_lng/check_hold_request	Suggested use
s	patron has exactly the same item on loan	1170	POST, CIRC
t	check for maximum number of hold requests for the ADM library (ADM line(s) in tab16).	1175	POST, CIRC
t_1	Check for maximum number of hold requests for the ADM library (ADM line(s) in tab 16) This routine is similar to check_hold_request_t, but also includes in the check existing loans that were originally hold requests.	1175	POST, CIRC
u	reading room permission	1176	POST, INFO, CIRC
v	If an item belongs to a remote storage facility and is available, the expected delivery time is calculated (using tab17 and tab24). The message displays in INFO in place of the *Your place in queue* message.	6003	INFO
w	Items that are currently loaned to a specific patron status or that have a specific item status cannot be requested. The patron and item status that are checked in combination are defined in col. 3. For example: `BOR=02,08;ITEM=03`	6004	PRE,POST,CIRC
x	Check for maximum number of hold requests for the sublibrary's group ("98" line)	1161	POST, CIRC
x_1	Check for maximum number of hold requests for the sublibrary's group (line 98) This routine is similar to check_hold_request_x, but also includes in the check existing loans that were originally hold requests.	1161	POST, CIRC
y	Patron with a specific status cannot request specific items (by item status and process status). Set up using parameters in column 3, for example: BOR=03,06;ITEM=03;PROC=MI,NA	6030	POST, CIRC
z	The hold request being submitted will not be first in the queue. This is mainly intended for the requests that require recalling, so that an indication is made that although a recall will be made, the requesting patron will not be the first to receive the item.	1177	CIRC-INFO
sh	Applies to tab_service_hour. Checks the request submission time (day+hour) against the pickup location service hours. The request is allowed only within the service hour's timeframe.	1235	POST, CIRC, XML-CIRC

Check	Action	$aleph_root/error_lng/check_hold_request	Suggested use
1_a	Checks patron global blocks. Error message include block codes and description from tab_delinq.lng table.	6041	PRE, POST, CIRC
1_b	Checks patron local blocks. Error message include block codes and description from tab/tab_delinq.lng table	6051	PRE, POST, CIRC
1_d	Checks if patron has overdue loans. A loan is considered overdue if the due date has passed and "n" or more notices have been sent, as defined in the library's tab/tab100 OVERDUE-LETTER-NO.	6070	PRE, POST, CIRC
1_e	checks patron block date	6060	PRE, POST, CIRC
4_a	checks if patron has exceeded the loan limit for the item status (tab16 column 12)	6111	PRE, POST, CIRC
4_b	checks if patron has exceeded the loan limit for the sub library (tab16 column 12)	6120	PRE, POST, CIRC
4_c	checks if patron has exceeded the loan limit for the ADM library (tab16 column 12)	6130	PRE, POST, CIRC
4_d	checks if patron has exceeded the loan limit for the sub library group (tab16 column 12)	6140	PRE, POST, CIRC
5	checks is the patron has exceeded the cash limit allowed in z305_cash_limit - uses O(pen) and T(ransferred) transactions. This check retrieves list of charges to include/exclude. For example: CHARGES=-0001,0006,0008,0009 This means that all charges except of 0001,0006,0008,0009 will be counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted.	6100	PRE, POST, CIRC

Check	Action	$aleph_root/error_lng/check_hold_request	Suggested use
5_b	Checks if the patron has exceeded the cash limit allowed. But only for O(pen) transactions. This check retrieves a list of charges to include/exclude. For example: CHARGES=-0001,0006,0008,0009 This means that all charges except of 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted.	6100	PRE, POST, CIRC
9_a	Checks the limits defined in tab/tab_block_circ (Maximum number of overdue loans, Maximum number of recalled overdue loans, Maximum cash owing). This check retrieves two parameters: • block circ table name. If not specified, the default table tab_block_circ is consulted. • list of charges to include/exclude For example: CHARGES=-0001,0006,0008,0009 This means that all charges except for 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted. If the CHARGES parameter is not retrieved, all cash charges are taken into account. The two parameters should be separated by a semicolon (;). For example: tab_block_test;CHARGES=+0001,0006,0008,0009	6081	PRE, POST, CIRC
9_b	Checks the limits defined in tab/tab_block_circ as 9_a but at item sub library level. This check retrieves the block circ table name. If not specified, the default table tab_block_circ is consulted.	6091	PRE, POST, CIRC

Check	Action	$aleph_root/error_lng/check_hold_request	Suggested use
10_a	As 9_a but counts all types of loans, i.e. active loans and claimed return. This check retrieves two parameters: • block circ table name. If not specified, the default table tab_block_circ is consulted. • list of charges to include/exclude For example: CHARGES=-0001,0006,0008,0009 Means that all charges except for 0001,0006,0008,0009 are counted. CHARGES=+0001,0006,0008,0009 This means that only 0001,0006,0008,0009 charges are counted. If the CHARGES parameter is not retrieved, all cash charges are taken into account. The two parameters should be separated by a semicolon (;). For example: tab_block_test;CHARGES=+0001,0006,0008,0009	6081	PRE, POST, CIRC
10_b	As 9_b but counts all types of loans, i.e. active loans and claimed return. This check retrieves the block circ table name. If not specified, the default table tab_block_circ is consulted.	6091	PRE, POST, CIRC

Note that only like copies that are available according to column 8 of `tab15` are considered available like copies.

TAB_PHOTO_REQUEST

`tab_photo_request` 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 photocopy requests for the Web OPAC determine whether the "Photo" link appears on the item line.

11.2. Priority

The position of the Hold Request in the queue is determined by assigning a Priority. The highest level priority is "00", the lowest is "99". You can set the default Priority for the GUI Circulation module and Web OPAC by editing `tab31`, column 19. The priority is dependent on the sublibrary of the requested item and on the patron's status.

11.3. Pickup/Delivery Locations

For each combination of sublibrary, item status and patron status, you may define the location from which the patron may pick up an item or photocopy being held for him, and the location to which the item or photocopy will be delivered. To do so, edit `tab37` (for hold requests) and `tab38` (for photocopy requests). When a pickup location is not defined, it is not possible to place the request.

When creating a Hold or Photocopy Request (WWW, Circ, OPAC), the default pickup sublibrary is the patron's home library. If the `Z303-HOME-LIBRARY` is not set up as a valid pickup library for the item in `tab37` (for holds) or `tab38` (for photocopies), the item's `Z30-SUB-LIBRARY` or the first sublibrary in `tab37` is used as the default instead. This depends on the setup of the `PICKUP-SORT-BY-ITEM` `tab100` variable.

11.4. Request Service Hours Checks (tab_service_hour)

The hold and photocopy request submission time (day+hour) can be checked against the pickup location service hours. The pickup location service hours are set in the ./xxx50/tab/tab_service_hour table. For table structure and more information, see tab_service_hour on page 109.

The service hour check is performed after the request form is filled and submitted by the user. If a request is not submitted during the pickup location service hours, the request is not accepted and the following message is displayed: “Pickup location is currently out of service. Request cannot be submitted” (message 1235).

The system checks the service hour (tab_service_hour) only when check_hold_request_sh and check_photo_request_sh check routines are set in tab_hold_request and tab_photo_request respectively.

To configure the pickup location service hour check for hold requests, set the following in ./xxx50/tab/tab_hold_request:

```
! 1                      2
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
POST       check_hold_request_sh
CIRC       check_hold_request_sh
XML-CIRC   check_hold_request_sh
```

Note: For hold requests, the only relevant instances for the service hour check are POST, CIRC, and XML-CIRC.

To configure the pickup location service hour check for photocopy requests, set the following in ./xxx50/tab/tab_photo _request:

```
! 1                      2
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
POST       check_photo_request_sh
CIRC       check_photo_request_sh
```

Note: For photocopy requests, the only relevant instances for the service hour check are POST and CIRC.

11.5. Hold Shelf

You may define different limits on the amount of time that an item is kept on the hold shelf, 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.

To define the different limits, edit `tab44`.

Note that the amount of time that the item is kept on the hold shelf is configured in tab44 depending on the pickup location. Also note that the amount of time that is defined in tab44 is net time, not counting the pickup location's closing days.

11.6. Printing Call Slips

The System Librarian is responsible for determining the approach that will be taken for printing call slips.

11.7. Item Holds List

The sort order on the Item Holds List is defined in `pc_server_defaults` with the following parameters:

```
setenv circ_user_z37_sort_routine      01
setenv circ_user_z37_sort_order        A
```

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
04 - sort using group ID and sequence, priority, request interest date and request date
A for ascending; D for descending.
06 – sort using sublib, pickup location, open date and requester name.
A for ascending; D for descending.

11.8. Loans List

The sort order on the Loans List is defined in `pc_server_defaults` with the following parameters:

00 - "standard" sort using due-date

01 - sort using sublib, item-status, collection, due-date

A for ascending; D for descending.

If the standard sort is sufficient, then these lines can be commented out.

11.9. Hold Request Log

The system librarian can decide which of the following hold request related events are recorded in the circulation log:

- Hold request created
- Hold request created (OPAC)
- Hold request deleted
- Hold request deleted (OPAC)
- Hold request placed on shelf
- Hold request filled letter printed
- Hold request not filled letter printed
- Hold request picked up

In order to create the circulation log, configure these events in <ADM library>/tab/tab_circ_log.lng (events 70-77).

11.10. photo_request_style

The System Librarian can decide whether or not to delete the request after the slip is printed. This is done by defining the `tab100` variable:

```
PHOTO-REQUEST-STYLE
```

If the variable is left blank, then when the Print Photocopy Request Slips (cir-22) service is run the system will print Photocopy Request slips and the requests will not be deleted. When the librarian enters the Circulation module to print a letter to the patron telling him to come and pick up the requested material, another slip will be printed for placing in the material, and only then the request will be deleted.

If the slip is printed by ue_06 or by cir_22, the request is not deleted but its status is changed to H.

If the variable is set to H, then when the Print Photocopy Request Slips (cir-22) service is run the system will print "Photocopy request filled" letters to the patrons in addition to the slips. The requests will be deleted and will no longer be available in the Circulation module.

11.11.KEEP-FILLED-PHOTO

The System Librarian can decide whether or not to keep photocopy requests after they have been filled. This is done by defining the `tab100` variable:

```
KEEP-FILLED-PHOTO
```

If the variable is left blank or set to N, then filled photocopy requests are not kept. If the variable is set to Y, then after performing the "Photocopy request filled" action, requests are kept in the photocopy requests list, with “Filled” status.

11.12. PHOTO-USE-EXACT

The System Librarian can decide whether or not to use the exact item information after the photocopy request slip is printed. This is done by defining the `tab100` variable:

PHOTO- USE-EXACT

If the variable is set to “Y”, then after the photocopy request slip is printed (in ue_06 or in other place) the request moves to the item used to fulfill it. Relevant item information of the item is printed on the slip.
If the variable is left blank or set to N, information of the first like item is used.

11.13. CREATE-Z38H

The System Librarian can decide whether or not to keep a photocopy requests history. This is done by defining the `tab100` variable:

```
CREATE-Z38H
```

If the variable is left blank or set to N, a photocopy requests history is not kept.
If the variable is set to Y, then “Deleted” and “Supplied” requests are kept in the photocopy requests history table (z38h).

11.14. Shipping

If the library uses the photocopy shipping functionality, the System Librarian should define the photo-shipment-id counter in the ADM library (by util g/2).

11.15. Rush Cataloging Requests

Placing a Rush Cataloging request is possible only if the item is defined as a Rush Cataloging item, and the patron is authorized to place a Rush Cataloging request.

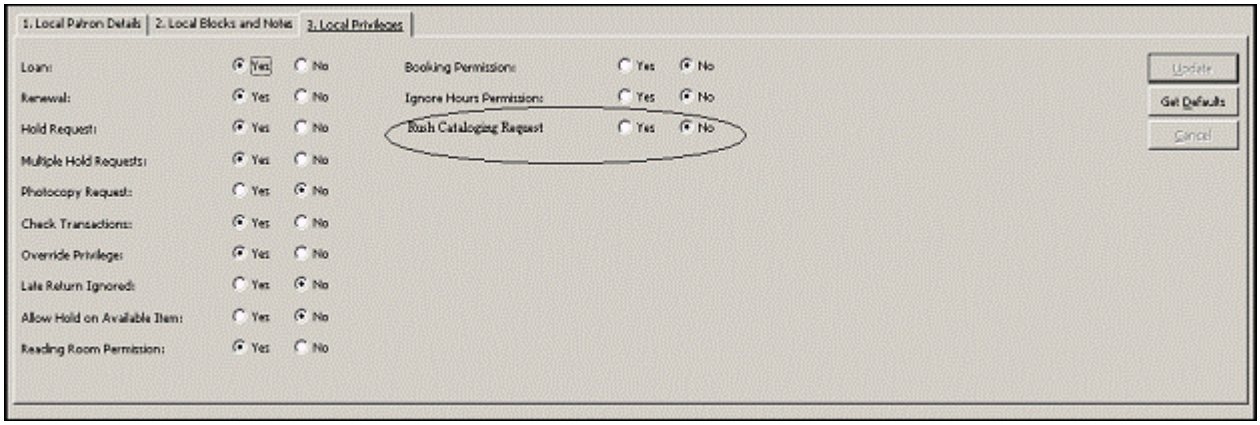
Defining the Item

An item is a Rush Cataloging item if it is set to “R” in column 8 of the relevant tab15.lng.

```
! 1  2 3 4       5                  6 7 8 9  1 1 1 1 1 1 1  1
!                                          0 1 2 3 4 5 6 7
!!!!!-!!-!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-!-!-!-!-!-!-!!-!-!
WID   ## BD L At bindery           Y N R Y Y N N N N 00 N A
```

Defining The Patron

The patron is allowed to place a Rush Cataloging request if the Local Patron Privileges have been set to ‘Y’:



Column 23 of tab31 sets the default for this privilege:

```
!                          1 1  1  1                          1   1 1 1   2  2  2   2
! 1      2   3  4 5 6   7  8  9 0 1  2  3     14          15        6   7 8 9   0  1  2   3
! ! ! ! !-!!-  !- !- !-!-  !-  !- !-!- !- !- !-  !! ! ! ! !! !-! ! !! ! !! ! ! !-!-  !- !-! !- !- !- !-   !
WID    02 Y Y Y N Y Y Y N F + M 00000006 9999999999 N Y Y 21 Y N Y N
```

The Rush Cataloging Request Department

The Rush Cataloging department is set by the tab_sublibrary_grouping table. This table, defined in the data_tab directory of the ADM library, sets a group code for each sublibrary. The trigger department is the group code that is defined in column 2 of this table for the requested item’s sublibrary (in column 1). Note that the sublibrary code is used if no group code is defined for the sublibrary.

11.16. Requesting Unbarcoded Items

It is possible to allow patrons to perform circulation activities on items that were not yet cataloged (“unbarcoded items”):

The bibliographic records for these items have at least one item record attached. This item record has an item status which denotes it as being a “multi” part item that is, this item record denotes that there are multiple items available for this bib record.
Note that this item does not represent a real item, but is used to indicate the existence of unbarcoded items related to this bibliographic record.

When requesting such item, a new temporary item record is created in the system.

Defining the ‘multi part’ item

An item is a multi part item if it is set to “U” in column 8 of the relevant tab15.lng.

```
! 1   2  3  4        5                    6 7 8  9   1  1  1  1  1  1  1   1
!                                                  0 1 2 3 4 5 6 7
```

!!!!!-!!-!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-!-!-!-!-!-!-!!-!-!
WID ## MP L Multi part Y N U Y Y N N N N 00 N A

The temporary item record

In general, the temporary item record will be created with the same information as the original 'multi-part' item (shelfmark, sublibrary, collection, process status, remote storage ID etc.). Specific fields are built as following:

- The item status of the temporary item record will always be "Temporary Unbarcoded" - item status 88 will be used.
- The barcode will be assigned according to a new counter: "last-barcode-tmp-num".
- The z30_description is built from the following request fields and in the following order:
  - Author (z37_author)
  - Title (z37_title)
  - Pages (z37_pages)
  - Note 1 (z37_note_1)
  - Note 2 (z37_note_2)

Information from the above fields is appended with commas between the values. Since the z30_description field is limited to 200 bytes, if the appended information is more than 200 bytes, it will be truncated.

The temporary item record is used to fill the request, and is loaned to the patron etc, i.e. would work as all other z30 item records.

After returned to the library, the temporary item record is deleted but is kept in the system. It is possible to use this temporary item record, give it a proper barcode and item status, and use it as a permanent item.

11.17. Requests Refusal

The library can decide to refuse a hold or a photocopy request. This is done via the Requests menu in the Circulation module – print letter refusal. This option enables the printing of a refusal letter, including the refusal reason. The request is moved to the History log with Refused status. A Circulation Logger record (Z309) is created that includes the refusal reason.

To enable the refusal flow, the following steps should be applied by the System Librarian:

1. Add the request refusal option to the requests menu by adding the following lines to GUI config under Circ/Tab/Lng/menu.dat:

REFUSE_HOLD_BARCODE Print Letter - Hold Request &Refused
REFUSE_PHOTO Print Letter - Photocopy Request &Refused
REFUSE_BOOKING Print Letter - Booking Request &Refused

2. Define the following refusal forms in ./<Bib Lib>/form_lng:

- hold-refused-letter-s.xsl
- photocopy-refused-letter-s.xsl
- booking-refused-letter-s.xsl

3. Add the following lines to./<Adm Lib>/tab/tab_circ_log.lng, r to allow log event creation on refusal:

```
64 Y Y LRefusal Hold Request note
65 Y Y LRefusal Photocopy Request note
66 Y Y LRefusal Booking Request note
```

11.18. Request Fulfillment by Preferred Items

The ue_06 request handling daemon prints slips for hold and photocopy requests. These slips include information about the item that is used to fulfill the request. Sometimes a library wants to define the preference order in which materials are used. For example, a library that keeps copies and originals wants to use the copies first, and use the original only when no other material is available. This is in order to avoid deterioration of the original materials.
In order to enable the usage of preferred items, apply the following:

1. Define the preference order of the items.
2. Define the relevant pickup locations for which the preferred item mechanisms are applied.
3. Define the check routines for checking whether the preferred item is available.

11.18.1. Defining the Preference Order of Materials

tab_req_preferred in the ADM library's tab directory is used to define which items are preferred and their priority when ue_06 is handling hold and photocopy requests.

Note:

Only preferred items should be listed in this table. There is no need to list all possible items here. If the tab_req_preferred table does not exist or is empty, the preferred item mechanism is not applied

Table Structure

- **Column 1: request type**
  Size: 1
  Possible values:
    - H–Hold Request
    - P-Photocopy request
    - # - Hold request and Photocopy request
- **Column 2: sublibrary**
  Size: 5
  Possible values:
    - Sublibrary code
    - # - any sublibrary

- **Column 3: item status**
  Size: 2
  Possible values:
    - item status code
    - # - any item status
- **Column 4: item process status**
  Size: 2
  Possible values:
    - Item process status code
    - # - any item process status
- **Column 5: item collection**
  Size: 5
  Possible values:
    - Item collection code
    - # - any item collection

Table Example

```
!  2    3  4   5
!-!!!!!-!!-!!-!!!!!
# LAW   01 ## GEN
# LAW   01 ## NBS
# LAW   ## ## #####
# HYL   01 ## GEN
# HYL   01 ## NBS
# HYL   01 ## #####
```

In the above example, the items of the LAW sublibrary with status 01 from the GEN collection are the most preferred. If these items are not available, then items of the LAW sublibrary with status 01 from the NBS collection should be used. If they are also not available, then the rest of the LAW items should be considered.

Only if all items similar to LAW items are not available should HYL items be considered.

Note that in the above example, the definitions are the same for both hold requests and photocopy requests. It is possible to define a different setup for hold and photocopy requests by setting column 1 accordingly.

11.18.2. Defining the Relevant Pickup Locations for which the Preferred Items Mechanism is Applied

The preferred items mechanism is applied by the requests handling daemon (UE_06) only for requests that have a pickup location that is listed in tab_req_preferred_loc in the ADM library's tab directory.

Note: If the table does not exist or is empty, the preferred item mechanism is applied for any pickup location.

Table Structure

Column 1: pickup location

Size: 5

Table Example

```
!  1
!!!!!
HOME
LAW
```

In the above example, the preferred items mechanism are applied to requests that are requested for pickup at the HOME or LAW pickup locations.

11.18.3. Defining Check Routines to Check if the Preferred Item is Available

The ue_06 requests daemon attempts to use an available preferred item.
The preferred item is considered available if it passes the check routines defined in tab_hold_request (sections UE06-LIKE, UE06-PREF) for hold requests or tab_photocopy_request (section UE06-AVAIL for photocopy requests.

11.18.4. UE_06 Processing of the Requests

Ue_06 attempts to fulfill hold and photocopy requests created in WEB OPAC, which are requested for one of the pickup location listed in `tab_req_preferred_loc` by a preferred item:
Items of the requested title that are preferred according to tab_req_preferred are checked:

- Do they match the item enumeration and chronology values of the item on which the request was made?
- Are they available?

The order in which the items are checked are according to the order of the tab_req_preferred lines. The most preferred items are checked first.
The first available preferred item that is found is used – i.e. this is the item that is printed on the slip.
In case of a hold request – the request moves to this item.
In case of a photocopy request – PHOTO-USE-EXACT should be used in order for the request to move to this item.
If no preferred item is available, the request is filled by the regular Aleph request fulfillment mechanisms.

Note:
The preferred items mechanism should be used only when PHOTO-REQUEST-STYLE is 3.

11.19. Staff Request

The Central Library is a library that offers services to subscribed libraries. Subscribed libraries may request material from the central library on behalf of their patrons.

The subscribed libraries are identified in the central library as "pseudo patrons", but their patrons are not identified in the central library.

When a patron of the subscribed library wants to request material from the central library, the staff of the subscribed library creates the request for the patron.

The staff of the subscribed library connects to the OPAC of the central library using the subscribed library's account.

When connected to the subscribed library's search base, the request forms for hold and photocopy are altered to include a field for the patron's name. The staff of the subscribed library fills in the request details, including the information of the requesting patron in the requester name field.

When the form is submitted, standard photocopy checks are done based on the subscribed library's logon information. The request is linked to the subscribed library's user, with the requester name saved in the Z37-REQUESTER-NAME / Z38-REQUESTER-NAME fields.

If cash charges are created, they are charged to the subscribed library's user, and the Z31-REQUESTER-NAME field is populated with the name of the requester.

To enable the staff request flow, the following steps should be applied by the System Librarian:

1. Define the subscribed library's account.
2. Add "requester name" fields to hold and photocopy forms in the OPAC.
3. Add "requester name" columns to request lists.
4. Add sort options based on the requester name.

11.19.1. Defining the Subscribed Library's Account

Define a new "pseudo" patron for the subscribed library with permissions according to the central library's policy.

The status of this patron indicates that it is not a regular patron but a subscribed library.

11.19.2. Adding "requester name" Fields to Hold and Photocopy Forms in the OPAC

Refer to the **Requester Name Field on Request Creation Forms** chapter of the Aleph 21 System Librarian's Guide - OPAC for details.

11.19.3. Adding "requester name" Columns to Request Lists

1. Display requester name information in the Patron Tab → hold requests list, by adding the following line to ./<BIB library>/tab/pc_tab_col.lng:

```
PC_CIR_B_HOLD_LIST     L  Requester  Name          21  000  04  C04
Requester name
```

2. Display requester name information in the Item Tab → history →hold requests

list, by addig the following line to ./<BIB library>/tab/pc_tab_col.lng:

```
PC_COM_HOLD_H_LIST     L  Requester  Name           11  000  04  C04
Requester Name
```

3. Display requester name information in the Patron Tab → photocopy requests list, by adding the following line to ./<BIB library>/tab/pc_tab_col.lng:

```
PC_CIR_B_PHOTO_LIST    L  Requester  Name           08  000  04  C04
Requester name
```

4. Display requester name information in the Patron Tab → historical photocopy requests list, by adding the following line to ./<BIB library>/tab/pc_tab_col.lng:

```
PC_CIR_B_PHOTOH_LIST  L  Requester  Name           06  000  04  C04
Requester Name
```

5. Display requester name information in the Patron Tab → cash list, by adding the following line to ./<BIB library>/tab/pc_tab_col.lng:

```
C_CASH_SUMMARY         L  Requester  Name           22  000  04  C04
Requester Name
```

Refer to the **Requester Name Field In My Library Card** chapter of the System Librarian's Guide -OPAC: for more details about adding the requester name field's information in the OPAC.

11.19.4. Adding Sort Options Based on the Requester Name

Sort the Patron Tab → hold requests list according to sublib, pickup location, open date, and requester name, by setting 06 in the circ_user_z37_sort_routine environment variable (defined in `pc_server_defaults)`.

The Patron Tab → photocopy requests list can be sorted according to the following sort options:

- Pickup Location / requester name / photocopy number
- Sub library / requester name / photocopy number
- Pickup Location / Sub library / requester name / photocopy number
- Pickup Location / Status / requester name / photocopy number
- Sub library / Status / requester name / photocopy number
- Pickup Location / Sub library / Status / requester name / photocopy number
- Pickup Location / Status / Sub library / requester name / photocopy number

These options are defined in ./<ADM library>/tab/tab_z38_sort, and matching entries in
./<ADM library>/tab/pc_tab_exp_field.lng under PHOTO-SORT-TYPE..

Refer to the **tab_z38_sort** section of the Aleph 21 Configuration Guide for more details.

- Sort the Patron Tab → Cash List according to transfer number, date, and requester name, with the 09 sort routine in ./<ADM library>/tab/tab_z31_sort, and define matching entries in

./<ADM library>/tab/pc_tab_exp_field.lng under CASH-SORT-TYPE.

- Sort the Patron Tab → Cash List according to requester name, by using the10 sort routine in ./<ADM library>/tab/tab_z31_sort, and define matching entries in ./<ADM library>/tab/pc_tab_exp_field.lng under CASH-SORT-TYPE.

Refer to the **tab_z31_sort** section of the *Aleph 21 Configuration Guide* for more details.

Refer to the **Sort Based On Requester Name Information** section of the System Librarian's Guide - OPAC for more details about sorting using the requester name field's information in the OPAC.

12. Setup for Recalls

For the System Librarian: To enable the Recall functionality, set up the following tables:

12.1. tab16

Define the recall period, the minimum guaranteed loan period and the fine method.

- Fill in column 19, No. of Days for Recall Notice, to define the number of days from the day the recall notice is sent to the new due date (this is referred to as the recall period). This sets the new due date.

- Fill in column 20, Minimum Guaranteed Loan Period, to define the minimum guaranteed loan period. This is the number of days that the original patron can have the item on loan, even if it has been requested by another patron and recalled.

In order for the system to generate a special extra fine for overdue recalled items, define the related fine method in tab16.

- In column 14, Fine Method, fill in the fine method according to library policy. Use only those methods that are marked as "include recall". One of the options that you can choose from is fine method number 7 which will take the recall into account when the item is returned late.

12.2. tab18.lng

Define fine rates for late return of recalled items. This is required when you use fine method number 7 that takes the recall into account when the item is returned late. To do so, edit tab18.lng. Examples of the relevant lines of this table are shown below:

```
0050 ####  ## Y 2.00        Recall late return fine
0051 ####  ## Y 4.00        Rush Recall late return fine
0052 ####  ## Y 1.50        Recall max fine ratio
0053 ####  ## Y 1.50        Rush Recall max fine ratio
```

The amounts specified in column 5 can be treated as an amount or as a ratio (depending on the setup in tab100). When a ratio is used the number 1.5 in column 5 will mean that in addition to the fee on the late return, the patron will be charged an

extra 1.5 times the late fee because the item was recalled. In other words, instead of paying a fee of X, the patron will pay a fee of X + 1.5X = 2.5X.

12.3. tab100

Define the recall related variables in tab100. These variables determine the way Recall works in your system.
OVERDUE-RECALL-RATIO. This variable defines whether lines 0050-0055 in tab18.lng (fines relating to recalled items) are treated as a ratio or as an actual amount.

UPDATE-RECALL-DATE. This variable determines whether, 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 Recall Items on Loan (cir-13) batch service.

- Y - loaning an item can trigger a 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 the Recall Items on Loan (cir-13) batch service.

RECALL-METHOD. This variable determines whether the Z36-ORIGINAL-DUE-DATE is updated and what due date will be displayed as the loan's due date in the GUI and in the Web OPAC.

- 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 later.

ONLINE-RECALL. This variable sets the library policy regarding online recall.

- Y - any request that is created in the GUI and is required for recall will trigger a recall when the request is submitted.
- N- recall will be triggered only when the 'Recall Items on Loan (cir-13)' service will be run.

RESET-LETTER-NUMBER. This variable sets whether an online recall of an item will reset the loan's Z36-LETTER-NUMBER field.

- Y - reset the Z36-LETTER-NUMBER field when an online recalled is triggered by the GUI.
- N - do not reset the Z36-LETTER-NUMBER field when an online recalled is triggered by the GUI.

13. Print Templates

13.1. Available Templates

The names of the Circulation template files are as follows (where nn is a number identifying the version):

GUI Templates

Template	**Purpose**	**print.ini function name**
Advance Booking Cancellation Letter item-sched-delete-letter-s-nn item-sched-delete-letter-nn		
Patron Summary bor-list-nn	May be sent from the Patron Information screen by choosing LIST after clicking the Letter button.	BorLetterLIST
Cash Transaction Details cash-line-details-nn	Show full details of single cash transaction record	SingleCashTrans
Hold Request Filled hold-request-letter-mm-nn	hold-request-letter-01-nn - hold shelf hold-request-letter-02-nn - home delivery hold-request-letter-03-nn - mailbox delivery hold-request-letter-04-nn - reading room hold-request-slip-nn (with "p" letter)	ReturnHoldnn (01-04) HoldLetternn (01-04) HoldLetterDeleten (01-04)
Hold Request Not Filled hold-request-wait-nn		HoldWait
Hold Request Slip hold-request-slip-nn	The "Requested by:" section refers to a hold placed by a proxy for his patron and is not displayed at all if the hold has not been placed by a proxy.	ReturnHoldnn HoldLetternn
Item Note item-slip-nn		ItemSlip

Template	Purpose	print.ini function name
Letters to Patrons borrower-letter-a-nn borrower-letter-b-nn borrower-letter-c-nn borrower-letter-d-nn		BorLetter
Library Card bor-card-label-nn	may be sent from the Patron Information screen by choosing CARD after clicking the Letter button	BorLetterCARD
Loan receipt loan-receipt-nn		LoanReceipt
Lost Material Bill lost-material-bill		LostMaterialBill
Photocopy Request Filled - Pickup photo-request-letter-p-nn		PhotoLetter ReturnPhoto
Photocopy Request Not Filled photo-request-wait-nn		PhotoWait
Photocopy Request Letter photo-request-letter-d-nn	Letter to patron informing him that his photocopy request has been filled and the item is attached (because the patron's delivery type is "deliver").	PhotoLetter ReturnPhoto
Photocopy Request Slip photo-request-slip-nn		PhotoSlip
Reading Room Report reading-room-report-nn	Online Reading Room Report	RRReport.prn
Reading Room Shelf Report rr-shelf-report-nn	Online Reading Room Shelf Report	RRShelf.prn
Renew All bor-list-renew-all	List of the patron's loans that were renewed and the reasons for not renewing others	BorListRenewAll BorListRenewAllRemote
Return Receipt		ReturnReceipt

Template	Purpose	print.ini function name
return-receipt-nn		
Single Hold Request has been Deleted hold-delete-letter-s-nn		HoldDelete (01-04)
Station Cash Report cash-site-payment-nn		StationReport
Transfer Slip transfer-slip-nn		TransferSlip
Patron Cash Summary Statement cash-summary-nn		TransLetter
Patron Circulation Summary bor-list-nn		
Cash Receipt cash-pay-receipt-pay		Payletter
Cash Receipt cash-pay-receipt-refund		Payletter
Cash Receipt cash-pay-receipt-waive-debit		Payletter
Advance Booking Schedule item-schedule		
Payments Received Report payment-received-report-nn		
Automatic Return Letter auto-return-letter-nn	Return letter that is created if the 'Create Return Receipt on Automatic Return' Loan option is on, the tab100 variable RETURN-DURING-LOAN is "1" or "2", and the item being checked out is registered as on loan to another patron.	AutoReturnLetter

Services Templates

Template	Purpose

Template	Purpose
Advance Booking Schedule Report reserve-item-schedule	Generated by cir-09
Cash Patron Letter cash-user-letter-nn	Generated by cash-05
Cash Transaction Report cash-transaction-report-nn	Generated by cash-02
Claimed Returned Items Report claimed-returned-items-report-nn	Generated by cir-19
Courtesy letters loan-notice-letter-nn	Used when the "automatic renew" option in the "Print Courtesy Notices" service (cir-10) is not checked.
Debt Notice Letter debt-notice-letter	Generated by cash-06
Expired Hold Requests Report / Deleted Expired Hold Requests Report lapsed-holds-nn	Generated by cir-17
Hold Delete Letter hold-delete-letter-nn	Generated by cir-06
Hold Request Wait Letter hold-request-wait-00	Generated by cir-12
Hold Request Slip hold-request-slip-00 hold-request-slip-01	Generated by cir-12
Hold Requests Report hold-requests-report	Generated by cir-07
Hold Shelf Report hold-shelf-report-nn	Generated by cir-06
Loan Notice Renewal Letter loan-notice-renew-letter-nn	Used when the "automatic renew" option on the "Print Courtesy Notices" service (cir-10) is checked, and the patron has items that have been renewed.
Loan Renewal Report loan-renewal-report	Generated by cir-08
Loan Report loan-report	Generated by cir-04
Lost Items That Were Returned returned-lost-items-summary-00	Generated by cir-16
Lost Material Bill lost-material-bill-nn	Generated by cir-50

Template	Purpose
lost-material-bill-m-nn	
Lost Material Summary Bill lost-material-summary-nn	Generated by cir-51
Outstanding Loan Requests outstanding-loan-requests	Generated by cir-11
Overdue Items Single Letter overdue-sum-single-0 overdue-sum-single-0-nn overdue-sum-single-1-nn overdue-sum-single-2-nn overdue-sum-single-3-nn overdue-sum-single-4-nn	Generated by cir-52
Overdue notice(single item per notice) overdue-letter-0 overdue-letter-0-nn overdue-letter-1-nn overdue-letter-2-nn overdue-letter-3-nn overdue-letter-4-nn	"0" is first notice, "1" is second notice, and so on. Generated by cir-50
Overdue Summary overdue-summary-0 overdue-summary-0-nn overdue-summary-1-nn overdue-summary-2-nn overdue-summary-3-nn overdue-summary-4-nn	Generated by cir-51
Patron List Report borrower-list-report-nn	Generated by cir-05
Patron Loader - Deleted Patrons patron-delete-approved	Generated by cir-23
Patron Loader - Rejected Patrons patron-delete-rejected	Generated by cir-23
Photocopy Requests photo-request-letter-d-00 photo-request-letter-p-00 photo-request-slip-00 photo-request-slip-01	Generated by cir-22
Photocopy Request Wait Letter photo-request-wait-00	Generated by cir-22
Recall Loan loan-recall-1-nn loan-recall-2-nn loan-recall-3-nn	loan-recall-1-nn - used when the due date of the recalled item has been changed. loan-recall-2-nn - used when the recall does not trigger a change in due date.

Template	Purpose
	loan-recall-3-nn - used when a request was cancelled and therefore the recall is also cancelled. Generated by cir-13
Report of Items in High Demand items-in-high-demand	Generated by cir-14
Report of Items in Reading Room items-in-reading-room	Generated by cir-15

14. Drop-down Lists

Many of the choices available on the drop-down lists of the Circulation GUI have been set by the developers and you may not add or delete choices. However, you may add or delete choices for some of the lists. The major Circulation drop-down lists that can be edited are:

Patron Status
You can add new patron statuses by editing the `pc_tab_exp_field_extended.lng` file for a sublibrary-sensitive list. For a standardized list that is not dependent on a sublibrary filter, add patron statuses by editing the `pc_tab_exp_field` file. In both files, the list is identified by the ID "BOR-STATUS." Make sure that the code for the patron status you want to add is present in `tab31`.

Patron Type
You can add new patron types by editing the `pc_tab_exp_field_extended.lng` file for a sublibrary-sensitive list. For a standardized list that is not dependent on a sublibrary filter, add patron types by editing the `pc_tab_exp_field` file. In both files, the list is identified by the ID "BOR-TYPE."

Block Code
You may add new block codes by editing `tab_delinq.lng`.

Letters to Patrons
If you have defined new templates for letters to patrons (that are sent from the Patron Information screen), be sure to add new list choices by editing the `pc_tab_exp_field` file. In that file, the menu is identified by the ID "BOR-PRINT."

Address Type
You may add new Address Types by editing the `pc_tab_exp_field` file. In that file, the list is identified by the ID "USER_ADDRESS_TYPE." If this type represents a periodical address, be sure to edit `tab_bor_address` accordingly.

Language
This is the language of correspondence with the patron. You may add new language choices by editing the `pc_tab_exp_field` file. In that file, the list is identified by the

ID "PATRON-LANGUAGE." Be sure that the language you add here has also been defined in `./alephe_tab/allowed_languages`.

ILL Library
You may add new ILL Libraries (as long as they also appear in the `tab_sub_library` file by editing the `pc_tab_exp_field` file. In that file, the list is identified by the ID "ILL-LIB."

Notes 1, 2, 3
You may add new options to the list for Notes 1, 2 and 3 of the Global Patron Information screen by editing the `pc_tab_exp_field` file. In that file, the lists are identified by the IDs "FIELD1," "FIELD2" and "FIELD3".

See also the `pc_tab_exp_field` and the `pc_tab_exp_field_extended.lng` tables for additional lists.

15. Catalog Item

The Catalog Item function (available on the Item menu) enables you to briefly catalog an item with minimum information. The System Librarian is responsible for setting the fields that are displayed on the Catalog Record form and which of these fields are mandatory.

To do so, go to the Bibliographic library's tab directory and edit the `pc_tab_circ_fast_cat.lng` file (where "lng" is replaced by the code for the language). Following is an example of that file:

```
!   2   3           4                       5
!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!
Y 100   a Author
N 245   a Title (mandatory)
Y 260   a Place
Y 260   c Date
G 035   a Sys Control No (Group)
G 020   a ISBN (Group)
G 022   a ISSN (Group)
```

- **Column 1**: Field control.
  Optional values are:
  Y = the field is optional
  N = the field is mandatory
  G = At least one of the fields, marked with G, is mandatory when fast cataloging is performed.
  H = the field is hidden. The field content is taken from this table, and not input in the form.
- **Column 2**: Field code.
- **Column 3**: Subfield code.
- **Column 4**: Field name or text of hidden field.

- **Column 5**: BIB line: This is only relevant for an ADM table and will cause any lines in the corresponding BIB table to be ignored.

For entering fields for an ADM record, edit a table with the same name located under the ADM library tab directory. This is optional. The table in the ADM library can be left empty, but must be present.

Note that there is an option to determine the bibliographic fields in the `pc_tab_circ_fast_cat.lng` table located in the administrative library by using column 5 of the table to specify that the tag is a Bibliographic tag.

If the `pc_tab_circ_fast_cat.lng` table located in the administrative library includes at least one line with a tag that is specified as a BIB tag, the `pc_tab_circ_fast_cat.lng` table in the Bibliographic library will be ignored.

This option can be used by a Multi-ADM environment as well by a single-ADM environment.

You can set up your system to automatically create a hold request, so that the item can be transferred to the Cataloging department to complete the cataloging process.

Steps

Configure the FAST-CAT-HOLD-ID variable in tab100. This variable determines the ID for which a hold request will be created when an item is created using the Fast Cataloging option. If this entry does not exist, then if the library uses user-sharing, the ID will be "CATALOGER". If the library does not use user-sharing, the ID will be a concatenation of "CAT-" and the library name.

Create a Global Patron with the required ID. If the FAST-CAT-HOLD-ID variable in `tab 100` is set to a patron ID, make sure such a patron exists. If FAST-CAT-HOLD-ID is not set to a patron ID, make sure that either a Global Patron is defined with a CATALOGER ID, or that the ADM library's code is prefixed by "CAT-" if the library does not use user sharing.

You can set a default "from date" for the automatically generated hold request of the "CATALOGER" patron when an item is processed through Catalog Item. The default is set in `alephe_root/pc_server_defaults`:

```
setenv circ_cataloger_request_date      001
```

16. Item Reshelving Time

The item reshelving time is the amount of time it takes to reshelve an item after it is returned. When an item is in the process of being reshelved, this condition will be displayed in the Item List of the Circulation module.
To define the item reshelving time, edit `tab14`. Following is an example from that table:

```
! 1     2   3  4   5
!!!!!-!!!!!-!!-!!-!!!!
UEDUC GEN   ## ## 0115
```

```
UEDUC ##### ## 04 0100
ULINC ##### ## ## 0115
UMUSI ##### ## ## 0115
UHLTH GEN   ## ## 0115
##### ##### ## ## 0015
```

- **Column 1**: is the sublibrary. This information is mandatory. For "all sublibraries," enter ##### .
- **Column 2**: is the collection. For "all collections," enter ##### .
- **Column 3**: is the item status. For "all statuses," enter ## .
- **Column 4**: is the item process status. For "all statuses," enter ##.
- **Column 5**: is the item reshelving time, that is, the number of hours and minutes it takes to reshelve an item after it is returned. (Note that the maximum number of minutes is 59.)

17. Item List

You can determine whether the Item List in the Circulation GUI client will display all copies of a bibliographic record, or only those copies on loan. To do so, open the GUISYS.INI file (found in the client's CIRC/TAB directory). Go to the [ItemList] section. Following is an example of what you may find there:

```
[ItemList]
FilterType=0
```

If you want the Item List to display all copies of a bibliographic record, type **0** to the right side of the equal (=) sign. If you want the Item List to display only those copies on loan, type **1**, instead.

This is a default setting which can be changed on the item list by checking (or removing check mark) in the "Loan Filter" check box.

There is a drop-down list for choosing the sort order of the list of items. When the "Loan Filter" is cleared, the choices are the same as the choices available in OPAC and ITEMS; when the Loan Filter is selected, you can choose the sort type from a list that is defined in the ADM library's `/tab/pc_tab_exp_field.lng` table where column 1 is LOAN-SORT-TYPE. The codes in column 4 of this table are the sort options that are defined in the `tab_z36_sort` table. Note that the sort type drop-down list in the patron's loan list is also defined in the `pc_tab_exp_field.lng` configuration table where column 1 is LOAN-SORT-TYPE.

18. Advance Booking Slots

In order for a patron using the Web OPAC to be able to advance booking items, you must set up the reserve advance booking schedule. This schedule shows the time intervals or "time slots" for which an advance booking item may be booked in advance.

To set up the schedule, edit `tab43`, an expanded table for defining advanced booking parameters. It defines the time slots available for placing an advance booking request. The table allows for the definition of the start day and hour, as well as the end day and hour. It can also be used to define "closed" times, that are independent of tab17.

Note that an item can be reserved for a patron for up to three days; therefore, the time slots should be defined to a maximum of three days.

An example from the table is shown below:

```
!-!!!!!-!!-!!!!!!!!-!!-!!!!-!!-!!!!
P WID## 44 20051101 00 1300 01 0900
                    01 0900 01 1100
                    01 1100 01 1300
                    01 1300 01 1500
                    01 1500 01 1700
                    01 1700 01 1900
                    01 1900 02 0900
                    02 0900 02 1100
                    02 1100 02 1300
                    02 1300 02 1500
                    02 1500 02 1700
                    02 1700 02 1900
                    02 1900 03 0900
                    03 0900 03 1100
                    03 1100 03 1300
                    03 1300 03 1500
                    03 1700 03 1900
                    03 1900 04 0900
                    04 0900 04 1100
                    04 1100 04 1300
                    04 1300 04 1500
                    04 1500 04 1700
```

You can define an entire schedule and in addition define days where the slots are different from the schedule as a whole. To do so, use the E line type to define the exact day.

The definition can be of a slot that is different from the one already set for the scheduled 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 day. The 00 in an E type of line indicates that the definition is for the day listed in column 4.

A 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.

The slot 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 day listed in column 4 and ends at 09:00 one day after that.

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 closed days in a row, set a line for each day.

You can use ## instead of the year, month or day in an E type of line. This way it is possible to create a more general definition for days on which no slots will be created or for different settings of time slots.

For example:

```
E ##### 60 200310##
```

This definition indicates that no slots will be created for October 2003.

```
E ##### 60 2003##01
```

This definition indicates that no slots will be open for the 1st of every month in 2003.

To define a time slot that continues to the following week, the following days which represent the end of the time slot (column 7) can be inserted: 07, 08, or 09.

For example:

06 0900 09 1100

In this example, the slot starts on Saturday at 0900, and wraps around the weekend, ending on the following Monday, at 1100.

Note: In order to see periodic slots for days before the E lines, the P lines must come before the E lines.

Note: There is a limit of 100 time slot lines (columns 5 to 8) for each group of columns 1 to 4 (Line Type, Sublibrary, Item Status, and End of Period Date).

Key to Table:

- **Column 1** is the line type. The available values are P, to indicate that the slots are for a period, and E, to indicate that the slots are for an exact date.

- **Column 2** is the sublibrary for which the schedule is being defined. Use ##### to define all sublibraries.

- **Column 3** is the item status (from tab15) which will be used for reserve items.

- **Column 4** is the date up until which the schedule is valid. Enter the date in the format yyyymmdd.

- **Column 5** sets the beginning day (00=Sunday) of the time slot during which patrons may borrow reserve items.

- **Column 6** sets the beginning hour and minutes of the time slot. The system uses a 24-hour clock, so for 6 p.m., enter **1800** . (Note that a colon (:) is **not** used to separate the hour from the minutes.)

- **Column 7** sets the end day of the time slot. Enter the number of days since Sunday. For example, enter 08 to specify the Monday that is 8 days after the first Sunday. (Note that in this case, column 5 can be 05 or 06).

- **Column 8** sets the end hour and minutes of the time slot. The system uses a 24-hour clock, so for 11 p.m., enter **2300**. (Note that a colon (:) is **not** used to separate the hour from the minutes.)

Note: Be sure to separate one time slot from another by a semi-colon (;).

Note

Apart from being confined to predefined time slots, an advance booking request is a regular booking request and it is managed by the system in the same manner as regular booking requests. Please refer to section 48 Booking Requests on page 142 for a description of the booking request setup.

19. Valid Dates of Patron Addresses

There are two ways to define the valid dates of the patron's address. One way is to edit the from-to dates on the patron address record (Z304), and use the accompanying setup of type "02" for the mailing address and type "01" for the permanent address.
Alternatively, the library can opt to set the validity period on a system-wide basis. In this case, the convention of "01" for permanent address and "02" for mailing address is irrelevant. This system-wide option is set in the administration library, in tab_bor_address. If there is a period defined in the table, then the system wide address conventions are in effect.

Note that the two methods cannot be used together.
The tab_bor_address definitions include the From date, the To date and the Address type:

```
19991001 19991223 21
19991224 20000109 22
20000110 20000615 23
20000616 20000923 24
20000924 20001223 21
```

Each line defines which address-type identification is in effect for a time period. The periods defined must be continuous, and must not overlap.

- If the address period is defined in tab_bor_address:
    - The "valid from-to" dates fields from the Z304 record are not displayed on the WEB update form (www_f_lng/bor-update).
    - When the patron updates his address record using the Web form (www_f_lng/bor-update), the current record will be updated. If there is no current record, the record that suits the immediately preceding line in tab_bor_address will be updated (only one line up).
    - In the same manner, when the system searches for the appropriate patron address record for a printout, if there is no record with the address type expected according to the time period in tab_bor_address, the system takes the record that has the type registered in the previous line in tab_bor_address, and duplicates it, assigning the expected address type.
    - The system's default address type will be the one that is is valid according to the date ranges that are listed in tab_bor_address. This means that when a patron is created, the default address type that will be created will be the one that is currently active according to this table.
- If the address period is NOT defined in tab_bor_address:
    - Updating the address in the Web form (www_f_lng/bor-update) will change the last record.
    - 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 with the highest sequence number. Remember, when sending notices to the patron, the system first searches for an address of type "02" where the date of the notice falls within the valid from-to period. If the system does

not find a suitable 02 address, the system then searches for a suitable 01 address.

- The system's default address type and validity range will be based on the ADM library's tab100 variable named DEFAULT-ADDRESS. This variable sets both the default address type and the number of days, starting from the patron creation date, on which the address will be considered valid. The default will be stored in the Patron Address's Date From and Date To fields.

20. Item History and Summary Window

There are two tabs on the Item History window which is accessible from the Item List. One tab is for previous loans and the other is for previous hold requests.

You can determine whether or not the system will store information about **previous** loans and hold requests by editing the `tab100` table. Following are the relevant lines in `tab100`:

```
CREATE-Z36H=Y
CREATE-Z37H=Y
```

In the above lines, Z36H refers to loans and Z37H refers to hold requests. If you want the system to store the information, enter Y (as shown above). If you do not want the system to store the information, enter N, instead.

The columns for displaying the information about **previous** loans and hold requests in the Item History and Summary window are defined in `pc_tab_col`. The codes in that table are: PC_COM_LOAN_H_LIST (for Previous Loans) and PC_COM_HOLD_H_LIST (for Previous Hold Requests).

21. Loan and Return Windows

The content of the Patron Details window is determined by the System Librarian in the user-details display template. See the Setup for Expanded Information Display section of the General - System Librarian chapter for more details.

22. Cash

22.1. Cash Transaction Information Window

On the Cash Transactions tab of the Patron Information window, there is a Status column with text such as Paid, Not Paid, Cancelled, Credited, Not Paid By/Credited to Patron. This text is defined in `$aleph_root/error_lng/cash_status_heading`.

In tab_attr_sub_library, type "5" is used to set under which sublibrary a general cash transaction (that is, one which does not belong to a specific sublibrary) will be registered. If this section is left empty, the transaction will be automatically registered under the active ADM library.

In tab_attr_sub_library type "6" is used to identify a workstation/library for CASH, as follows:

If an IP station has a sublibrary registered as type 6, when the Cash tab of the Patron Information window is viewed, only the transactions belonging to that sublibrary are displayed. In addition, the Sum to Pay is updated according to that sublibrary and contains only the sum owed to that specific sublibrary. You can define multiple sublibraries for one IP station.

A staff user can sort the cash list by selecting a choice from the drop-down list that is defined in the ADM library's /tab/pc_tab_exp_field.lng table where column 1 is CASH-SORT-TYPE. The codes in column 4 of this table are the sort options that are defined in the tab_z31_sort table.

22.2. Cash Receipt Numbering

Cash receipts can have separate counters by IP address/station ID.

Attribute type 8 in column 2 of tab_attr_sub_library enables use of different cash receipt counters by IP address/station ID. If type 8 is defined in column 2 of tab_attr_sub_library, the cash receipt number is based on the UTIL/G/2 counter last-cash-rec-<suffix>. The name of the IP address/station ID registered as type 8 has to match the suffix of last-cash-rec-<suffix>.

If type 8 in column 2 of tab_attr_sub_library is not defined, the cash receipt number is based on the UTIL/G/2 counter last-cash-receipt-no and not on the counter last-cash-rec-<suffix>. In this case cash receipt numbering is unified for all stations.

22.3. E-mails Free of Charge

You can set up the system to refrain from charging postage for e-mail. This function is only available for the following cash transactions: 0015, 0042, 0080, 0081, 0082, 0083, 0090, 1024, 1026, 1027, 1028. Use column 6 in tab18 in order to define whether a patron with an e-mail address should be charged for a specific library service or not.

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. The patron's currently valid address has to include an e-mail address. Please note that the system only checks whether an e-mail address exists in the Z304 address record and not if it is valid.

22.4. Cash Reports

Cash reports can be created either online from the **Reports** menu, or by running the 'Payments Received Report (cash-03) service. In both cases, the output report includes a line for summary of each group of transactions (depending on how the user decided to group them), a line for the balance of each group, and a line for the full sum owed.

The texts that are used by the reports for these lines are taken from the $aleph_root/error_lng/cash_report_heading.

Cash Transaction Report (cash-02)
This service produces a report of cash transactions for library patrons. Note that 15 is a special report format. Use it to include patron information in each line and not only in the first line of each new patron.

23. SQL Queries

SQL queries are used to retrieve information about various circulation transactions. The Z35 Oracle table stores information about transactions. The field within this table called Z35-EVENT-TYPE has the following values which may be used in SQL commands to retrieve data:

- 31 - Creating New ILL Request
- 50 - Simple Loan
- 51 - Reading Room Loan
- 52 - Offline Loan
- 53 - Advance Booking Loan
- 54 - Routing List Loan
- 55 - ILL Loan
- 56 - Transfer Loan
- 57 - Remote Storage Loan
- Delete Loan
- 59 - Maintenance Loan
- 61 - Return
- 62 - Web OPAC Renewal
- 63 - GUI Renewal
- 64 - Batch Renewal (automatic renewal performed by the services cir-08 and cir-10)
- 65 - Item Declared Lost

- 66 - Item Claimed Returned
- 71 - Expand Hold Request on Available Item
- 72 - Non-Expand Hold Request on Available Item
- 73 - Expand Hold Request on Nonavailable Item
- 74 - Non-Expand Hold Request on Nonavailable Item
- 75 - Reserve Advance Booking Item
- 76 – Rush Cataloging Requests
- 80 - In-house Use
- 81 - Photocopy Request
- 82 - Return of Item Not on Loan
- 90 - Booking
- 91 - Booking turning to Loan
- 92 - Delete Booking
- 95 - Other Institute Fulfillment
- 96 - Item Is Received at Pickup Location

Standard Circulation Reports based on SQL are accessible from the GUI Circulation module.

Circulation transactions will be registered according to the setup of `tab_events.lng` of the administrative library.

24. Audio Warnings

For the Loan and Return windows, you can set the system to assign audio warnings to program events by editing the `circ.ini` file on the client, under the `[ErrorAction]` section.

To define an audio warning:

1. Make sure the sound you want to set is defined in Windows to produce an audio signal. click **Start** > **Settings** > **Control Panel** > **Sounds and Multimedia.** (This might be different depending on the version of Windows that you are using.) For example, for Exclamation, select chimes.wav.
2. Check that you can hear sounds in your PC.

3. In `circ/tab/circ.ini`, in the `[ErrorAction]` section, type `Y,S,` at the end of the program event to which you want to assign an audio warning, together with the name of the sound you want. The syntax is:

<ProgramEvent>=<Y,S,SoundName>

For example:

```
NoUser=Y,S,Exclamation
NoBorrower=Y,S,Asterisk
ItemTransfer=Y,S,Beep
```

The following are the names of the program events for which you can define an audio warning:

```
NoUser=N
NoBorrower=N
LoanError=N
ReturnError=N
UserError=N
ItemError=N
ClaimReturn=N
LostReturn=N
LoanDenied=N
ReturnDenied=N
ItemTransfer=N
ItemRequested=N
LoanSuccess=N
ReturnSuccess=N
OnHold=N
UserVerification=N
RemoteReturn=N
ReturnIllLend=N
OnHoldReturn=N
```

The options for every sound are:

1. `SoundName=N` – No sound.
2. `SoundName=Y,S,` System sound name - This setting uses a system sound for the sound name. The list of available system sounds is listed below.
3. `SoundName=Y,F,` Sound file name - This setting uses a sound file for the sound name. The sound file should be placed in the `Alephcom\Sound` directory.

For example:

```
NoBorrower=N
NoBorrower=Y,S,Beep
NoBorrower=Y,F,RINGOUT.WAV
```

The available sounds at the moment are Windows standard sounds:

- QUESTION
- EXCLAMATION

- OK
- HAND
- BEEP
- ASTERISK

25. Item Process Status

Items with a defined process status are called "in process". These items are not available for loans as long as they are in process. Therefore, such an item should be registered as col 6 = N in `tab15.lng`. Furthermore, they may be requested, and in this case, they are registered as col 8 = Y or col 8 = C in `tab15.lng`. When such an item is requested, it will be shown in waiting position instead of printing a hold request slip since it cannot become available through a return action. When the item process status is changed from "in process" to "not in process", the item becomes available.

You can configure the system to remove specific process statuses automatically when the item is circulated by its barcode. This is useful if the process status has been placed to signify that the item is lost, missing or otherwise removed from the library's stack. The item being circulated by its barcode should therefore trigger removing such a process status.

The process statuses that will be removed are set in the tab100 variable LOST-PROC-STATUS. Any of the process statuses that are listed in this variable will be removed if one of the following applies to the item:

- Item is loaned or returned.
- Item information is changed in the Circulation GUI.
- Item is requested by using its barcode.
- Item is renewed by its barcode.

The circ.ini flag NotifyProcessStatus (in the [General] section) controls whether or not the staff will be notified of the process status removal.

26. Overriding a Block for Loans and Renewals

The override mechanism for Loans and Renewals works according to the following mechanism:

In order to be able to override the error messages you must have the following properties defined (in order of priority):

1. Z305-OVER-PERMISSION (patron permission).
2. USER PERMISSION to do an override (user_functions).
3. A USER LEVEL high enough that covers all your error messages (this value is a field in the record Z66, Z66-USER-CIRC-LEVEL.

This field can be updated through the Staff Privileges function available from all GUI modules. To access the Staff Privileges functionality right-click the key icon on the Operations bar (at the bottom right of the screen) and select Staff Privileges.

In the table `check_circ_override`, you can set the level a staff user has to have in order to be able to override a specific error message.

If one or more of the above mentioned properties is missing, then the Override button will be disabled.

27. Client Setup (circ.ini)

The circ.ini file defines settings for the Circulation client. This chapter presents and explains the following sections of the circ.ini file:

- BookingList on page 83
- ChangeDueDates on page 83
- CircAdminTree on page 84
- CourseReading on page 84
- ErrorAction on page 84
- FastCirc on page 85
- [FilterProfileByLibrary] on page 85
- General on page 83
- Hold on page 85
- HoldFilledList on page 85
- HoldList on page 86
- ItemHoldList on page 87
- ItemList on page 87
- ItemBookingList on page 87
- ItemSchedList on page 88
- Loan on page 88
- LoanList on page 88
- Lost on page 89
- MagneticMedia966 on page 90
- OfflineCirc on page 90
- Patron on page 91
- Payment on page 91
- RemoteRegistration on page 92
- Return on page 92
- [ReturnSession] on page 92
- ReturnHold on page 92
- RfidMedia on page 93
- StationReport on page 94
- TitleReqList on page 94
- UserList on page 94

Explanations of other setting definitions, relating to more than one module, can be found in the General System Librarian chapter of the ALEPH User Guide.

[General]

```
DatesDefaultSubLibrary=WID

UpdateDatesDefaultSubLibrary=Y

NotifyProcessStatus=
```

DatesDefaultSubLibrary
Determines the default sublibrary that displays when the Circulation GUI's Active Due Dates node in the Loan tab is accessed.

UpdateDatesDefaultSubLibrary
Determines whether or not a change in the displayed sublibrary the Circulation GUI's Active Due Dates node in the Loan tab is accessed is saved after the GUI is closed. Setting this variable to N causes the sublibrary defined in the DatesDefaultSubLibrary variable to always be used when the GUI is opened.

NotifyProcessStatus
This variable defines whether or not the user will be notified when a process status is automatically removed from an item. This may happen if an item that is being circulated by its barcode has a process status that is defined in the tab100 variable LOST-PROC-STATUS for removal.

[BookingList]

```
WantDeleteLetter=Y

WantDeleteLetterNote=Y
```

WantDeleteLetter=Y
Determines whether or not the system sends a letter to the patron if a booking request is deleted from his Booking List.

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting the booking request. This is only operative if the parameter is set to `WantDeleteLetter=Y`.

This parameter is relevant for booking requests deleted from the patron's Booking List.

[ChangeDueDates]

```
[ChangeDueDates]

DoCheck=N
```

Enables the display of the window informing how many loans match the range of dates before changing their due dates.

When a staff user is using the "change due date of current loans" functionality, setting DoCheck to Y, will show a prompt telling the patron how many loans were found and how many are going to be changed.

[CircAdminTree]

```
DefaultTree=B
```

This variable sets the first branch that is opened when the Admin tab is clicked in GUI-Circulation.

The possible values are:

B (booking), C (list of courses), J (file list), A (batch log), Q (batch queue), D (print daemon).

The default value is B (Booking).

[CourseReading]

```
DisplayNotesDialog=Y
```

In GUI-Circulation-Course Reading: Adding doc record to course by activating the "Search" button. This variable set the option to automatically display the “Edit Notes" form after choosing a record.

Y-The "Edit Notes" form is pop-ups after selecting a record.

N- The "Edit Notes" form is not displayed.

[ErrorAction]

```
[ErrorAction]
NoUser=N
NoBorrower=N
LoanError=N
ReturnError=N
UserError=N
ItemError=N
ClaimReturn=N
LostReturn=N
LoanDenied=N
ReturnDenied=N
ItemTransfer=N
ItemRequested=N
LoanSuccess=N
ReturnSuccess=N
OnHold=N
UserVerification=N
RemoteReturn=N
ReturnIllLend=N
```

The options for every sound are:

- `SoundName=N` – No sound.

- `SoundName=Y,S,` System sound name - This setting uses a system sound for the sound name. The list of available system sounds is listed below.
- `SoundName=Y,F,` Sound file name - This setting uses a sound file for the sound name. The sound file should be placed in the `Alephcom\Sound` directory.

For example:

```
NoBorrower=N
NoBorrower=Y,S,Beep
NoBorrower=Y,F,RINGOUT.WAV
```

The available sounds at the moment are Windows standard sounds:

- QUESTION
- EXCLAMATION
- OK
- HAND
- BEEP

[FastCirc]

```
[FastCirc]
StartMode=R
```

StartMode sets the default mode for Fast Circulation. If set to "R", the Fast Circulation form is opened with the Return radio button selected. If set to "L", it is opened with the Loan radio button selected.

[FilterProfileByLibrary]

This flag determines whether or not profiles that belong to other libraries are displayed when the profile list is displayed. This is relevant to the Global Patron Information form profiles drop down list.

[Hold]

```
[Hold]
CreateTypeTitle=N
```

CreateTypeTitle
This variable defines whether or not a hold request form will be loaded by default with the 'Title Request' check box selected.

[HoldFilledList]

```
[HoldFilledList]
AutoClose=Y
AutoSelect=Y,2.5
AutoSelectMode=M
WantDeleteLetter=Y
WantDeleteLetterNote=Y
```

This section provides the option to automatically select and close the Hold List window when using "Print Letter - Hold Request filled", when no button is clicked and there is only one hold on the list. The system will automatically select the patron from the list, print a letter for that patron and close the window in 1.5 seconds.

AutoCLose=Y
If you want the window to close automatically after a selection has been made, use Y. If you do not want the window to close automatically, use N.

AutoSelect=Y
If you want the system to automatically select the entry in the List of Hold Requests, use Y. Then set the number of seconds that you want the system to wait before selecting the entry (for example, 2.5 seconds).

If you want to be able to select the entry by yourself, without a time limit, use N.

AutoSelectMode=M
This parameter controls the auto select of the first entry in the Hold Filled List. If the switch is set to S (ingle) the system will automatically print the letter if there is a single entry is in the list, if there is more than one hold on the list the selection must list and no button is clicked. If there is more than one hold on the list the selection must be performed manually. If the switch is set to M (ulti) the system will automatically print the first entry in any case, whether there is only one or more entries in the list.

WantDeleteLetter=Y
Determines when a hold request is deleted, whether to send a letter to the patron notifying him that his request was deleted from the "Hold filled List".

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting the hold request. This is only operative if the parameter is set to WantDeleteLetter=Y.

[HoldList]

```
[HoldList]
WantDeleteLetter=Y
WantDeleteLetterNote=Y
```

WantDeleteLetter=Y
Determines whether the system should send a letter to the patron if a hold request is deleted from his Hold List.

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting the hold request. This is only operative if the parameter is set to WantDeleteLetter=Y.

This parameter is relevant for hold requests deleted from the Patron's Hold List.

ItemHoldList

```
[ItemHoldList]
WantDeleteLetter=Y
WantDeleteLetterNote=Y
```

WantDeleteLetter=Y
Determines whether the system should send a letter to the patron if a hold request is deleted from his Hold List.

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting the hold request. This is only operative if the parameter is set to WantDeleteLetter=Y.

Note

These are the same definitions as those of [HoldList], the only difference being that the definitions in ItemHoldList concern the item list, whereas the definitions of HoldList concern the borrower list.

[ItemList]

```
[ItemList]
FilterType=0
LoanSortType=LOAN-1
ItemSortType=ITEM-2
```

LoanSortType
When only the items on loan are displayed, this parameter sets the sorting order in which they appear. The options are set in `pc_tab_exp_field.lng`.
ItemSortType
When in the item list, all the items are displayed, this parameter sets the sorting order in which they appear as default. The options are set in `pc_tab_exp_field.lng`.

The Item list can be sorted according to either of the following Filter types:

- Show all items=0
- Loan filter=1

The ItemList FilterType can also be changed manually using the "Loan filter" checkbox on the Item List in the Circulation client.

[ItemBookingList]

```
WantDeleteLetter=Y
WantDeleteLetterNote=Y
```

WantDeleteLetter=Y
Determines whether or not the system sends a letter to the patron if a booking request is deleted from the item's Booking List.

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting the booking request. This is only operative if the parameter is set to `WantDeleteLetter=Y`.

This parameter is relevant for booking requests deleted from the item's Booking List.

[ItemSchedList]

```
[ItemSchedList]
WantDeleteLetter=Y
WantDeleteLetterNote=Y
```

WantDeleteLetter=Y If you want the system to automatically generate a letter to the patron informing him that his advance booking reservation has been deleted, type *Y*. If you do not want a letter generated, type *N*.

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting his advance booking reservation. This is only operative if the parameter is set to WantDeleteLetter=Y.

[Loan]

```
[Loan]
LogSize=300
CheckPatronBooking = Y
```

LogSize=300
The log size of the "Loan History" list. The list is available when the Loan window is open.

CheckPatronBooking=Y
This parameter sets the system behavior if during loaning, a booking of the same patron is found on a like item. Setting this parameter to Y will cause the system to warn that there is a booking on a like item. Setting the parameter to N will cause the system to treat the request as if it is on the loaned item, without warning the user.

[LoanList]

```
[LoanList]
WantReturnWarning=Y
WantBorListOnRenewAll=N
RenewAllBorPresentDefault=Y
DisplayRenewAllCheckWindows=Y
OverrideRenewAllChecks=N
RenewOverrideAllSimilar=Y
```

Note that the variables that affect the Renew All function also affect the Renew Marked function when more than one loan is selected.

WantReturnWarning
This parameter determines whether or not a confirmation will be required when a loan is deleted using the Delete button on the List of Loans.

WantBorListOnRenewAll=Y
If the value is **N**, the system does not automatically print a letter after a "Renew All" procedure. If the value is **Y**, printing of the form bor-list-renew-all is automatically triggered.

RenewAllBorPresentDefault=Y
This parameter defines whether or not the check box next to "Is patron present?" in the pop-up window after performing "Renew All" should be checked by default.

DisplayRenewAllCheckWindows
OverrideRenewAllChecks

The following options are controlled by the two above variables:

1. Automatic Override - All
   Automatic renewal of all loans through Renew All even if there are reasons to prevent a renewal. In order for the system to work this way the setup is:

   ```
   DisplayRenewAllCheckWindows=N
   OverrideRenewAllChecks=Y
   ```

2. Automatic Override - None
   Automatic renewal of only those loans which do not have reasons to prevent a renewal. In order for the system to work this way the setup is:

   ```
   DisplayRenewAllCheckWindows=N
   OverrideRenewAllChecks=N
   ```

3. Manual Override
   Renewal is according to the manual actions of the operator while performing Renew All. In order for the system to work this way the setup is:

   ```
   DisplayRenewAllCheckWindows=Y
   OverrideRenewAllChecks=N/Y (not critical)
   ```

   RenewOverrideAllSimilar=Y
   This switch controls whether or not the same error is ignored if it has previously been overridden for the same item.

   - If set to **Y**, the system ignores the same error message if the item is renewed again.
   - If set to **N**, the system displays and relates to an error message that was overridden on the occasion of an earlier renew action. The setting refers to both single Renew action and to a Renew All action.

[**Lost**]

```
[Lost]
ManualReplacement=Y
```
The staff user can manually input the replacement costs for lost material using Manual Replacement, when this parameter is set to Y.

[MagneticMedia966]

Some libraries use theft-detection devices. When an item is loaned, the librarian passes it through a device that demagnetizes the magnetic strip on the spine of the book. This device can erase data that is stored on magnetic media such as CDs or diskettes.

It is possible to provide a warning when magnetic media is loaned. A column in tab25 (material type) is used to mark items as magnetic media. When an item that is defined as magnetic media is loaned, a warning light on the device will light up and the librarian will be able to deactivate the magnetization device

This option is only available with the 3M 966 magnetization/demagnetization device.

```
[MagneticMedia966]
Port=0
NeedResponse=N
MagnetizeSequence=DDD
DemagnetizeSequence=RRR
MagneticMediaSequence=SSS
```

When an item is loaned, the librarian passes it through a device that demagnetizes the magnetic strip on the spine of the book. This device can erase data that is stored on magnetic media such as CDs or diskettes.
It is possible to provide a warning when magnetic media is loaned. A column in tab25 (material type) marks items as magnetic media. When an item that is defined as magnetic media is loaned, a warning light on the device will light up and the librarian will be able to deactivate the magnetization device.

Port
This is the port number to activate the warning for magnetic media.

NeedResponse
NeedResponse determines whether a response from the 966 magnetization device is necessary or not. If set to **Y**, then the Circulation client waits until the device responds.

MagnetizeSequence=DDD
DemagnetizeSequence=RRR
MagneticMediaSequence=SSS
These three parameters are required by the 966 magnetization device.

[OfflineCirc]

```
[OfflineCirc]
StartMode=L
BlockSize=5
```

StartMode=L
Determines the default mode of the offline circulation window (loan or return).

BlockSize=5
Determines how many transactions per file are sent to the server when the offline circulation is finished.

[Patron]

```
[Patron]
PictureDirectory=C:\Temp
FastPatronRegistration=Y
```

PictureDirectory=C:\Temp
This determines the drive and subdirectory that is opened by default when you are adding or updating a patron's picture from the GUI.

FastPatronRegistration=N
This flag determines which of the New Patron Registration forms will be displayed when the New Patron option is activated. If this flag is set to N, the Global Patron Information form will be displayed. If the flag is set to Y, the Patron Registration form will be displayed.

[Payment]

```
[Payment]
CashNoReceipt=1
AllowWaiveOnTransfered=N
AllowPayOnTransfered=N
PaymentMode=00
```

CashNoReceipt=1
Number of copies of payment receipts (up to 9) to print. If you do not want any printed, type zero ("0").

The actual printer that will be used is defined in the PRINT.INI file, with the function name PayLetter or LineLetter. If the Pay Selected option is used, then the LineLetter entry is used. If the Pay Sum option is used, then the PayLetter entry is used.

AllowWaiveOnTransfered=N
This switch determines whether or not the system allows the waiving of cash transactions transferred to AR (Accounts Receivable).

AllowPayOnTransfered=N
This switch determines whether or not the system allows the payment of cash transactions transferred to AR.

PaymentMode=00

This variable sets the default payment mode that is used when the Pay Sum or Pay Selected buttons are clicked.

[RemoteRegistration]

```
UseGlobalPassword=N
```

This section determines how the system will operate regarding the operator's user name and password when using the Remote Registration option.

When set to N, the system does not perform a preliminary check and immediately shows the ALEPH Temporary Password window upon choosing the Remote Registration option.

When set to Y, the system first tries to check the current operator's password in the remote library. The ALEPH Temporary Password window will only be shown if there is no such user name and password in the remote library.

[Return]

```
[Return]
LogSize=300
DateOverrideActive=5
ReturnLoadPatron=N
```

LogSize=300
The log size of the "Return History" list. The list is available when the Return window is open.

DateOverrideActive=5
Defines after how many minutes of the station being inactive, there will be a message informing the librarian that the Date Override option is still in use.

ReturnLoadPatron=N
This variable enables the option of loading patron information after the return action is completed. Default: N.
Y – After the return action is completed, the GUI loads the patron context. Users may switch to Patron tab to view the patron information.
N – Return action does not load patron context.

[ReturnSession]

RemoteStoreMessage=Y

This section gives the option set behavior of the return action.

RemoteStoreMessage=Y
If you want to display the “item must be returned to remote storage facility” message, use Y or leave the entry unspecified. If you do not want to get this message, use N.

[ReturnHold]

```
[ReturnHold]
AutoClose=Y
AutoSelect=Y,0.1
AutoSelectMode=M
```

```
WantDeleteLetter=Y
WantDeleteLetterNote=Y
```
This section gives the option to automatically select and close the Hold List window when the item is returned and there is only one hold on the list. The system will automatically select the patron from the list, print a letter for that patron and close the window.

AutoCLose=Y
If you want the window to close automatically after a selection has been made, use Y. If you do not want the window to close automatically, use N.

AutoSelect=Y
If you want the system to automatically select the entry in the List of Hold Requests, use Y. Then set the number of seconds that you want the system to wait before selecting the entry (for example, 0.1 seconds).

If you want to be able to select the entry by yourself, without a time limit, use N.

AutoSelectMode=M
This parameter controls the auto select of the first entry in the Return Hold List. If the switch is set to S (ingle) the system will automatically print the letter if there is a single entry is in the list, if there is more than one hold on the list the selection must be performed manually. If the switch is set to M (ulti) the system will automatically print the first entry in any case, whether there is only one or more entries in the list.

WantDeleteLetter=Y
Determines when a hold request is deleted, whether to send a letter to the patron notifying him that his request was deleted from the "Returned item hold list".

WantDeleteLetterNote=Y
Determines whether or not the system allows the operator to add a message to the letter sent to the patron, specifying the reason for deleting the hold request. This is only operative if the parameter is set to WantDeleteLetter=Y.

[RfidMedia]

```
[RfidMedia]
ActivateReader=Y
SuccessMessage=Y
```

The [RfidMedia] section is required when library items are RFID-tagged. For more information on using RFID equipment, see Aleph Support for RFID on page 176. This section include information on the following RFID products: Nedap, 3M, and Bibilichip (Bibliotheca).

ActivateReader=Y
This variable determines whether or not the RFID Reader is updated when the relevant GUI actions are triggered.

SuccessMessage=Y

This variable determines whether or not a message indicating success is issued when the RFID Reader update action succeeds.

[StationReport]

```
[StationReport]
FromDate=19990518
FromHour=1700
ByIdIp=ID
BySubLibraryTarget=S
```

When running the Payment Report from the Circulation client, the librarian can choose to save the date and time of the last report, and later use it as the beginning time and date of the next report.

FromDate
The last date the Payment Report was run.

FromHour
The last hour the Payment Report was run.

ByIdIp
This field sets the default of the "Create Report Using" checkbox in the Payment Report window.

BySubLibraryTarget
This field sets the default of the "Sort Report By" checkbox in the Payment Report window.

[TitleReqList]

```
WantDeleteLetter=Y
WantDeleteLetterNote=Y
WantDeleteLetter=Y
```

Determines if the system sends a letter to the patron if a title request is deleted from his Title Request List.

WantDeleteLetterNote=Y

Determines whether or not the system allows the operator to add a message to the letter sent to the p⇨atron, specifying the reason for deleting the title hold request. This is only operative if the parameter is set to WantDeleteLetter=Y.

This parameter is relevant for title hold requests deleted from the Patron's Title Hold List.

[UserList]

```
[UserList]
BorrowersOnly=N
FilterType=0
```

BorrowersOnly=N
Defines if the "Display Local Patron Only" check box in the Patron List is selected by default:

- If set to **Y**, only patrons with at least one local record are displayed. Whether or not patrons whose only local record is ALEPH are

displayed depends on the setup of the user_list_aleph_local parameter in pc_server_defaults.

- If set to **N**, all patrons are displayed.

FilterType=0
The Patron List can be sorted according to one of the following parameters: Patron ID, Patron Name or Patron Barcode. This parameter sets the default value for the sorting of the Patron List.

- FilterType=1 sorts list by Patron ID
- FilterType=0 sorts list by Patron Name
- FilterType=2 sorts list by Patron Barcode

The sort of the Patron List can be changed manually using the "Sort by" option in the Patron List in the Circulation client.

28. Circulation Values of pc_server_defaults

pc_server_defaults is an ALEPH configuration file located under $alephe_root. This configuration file defines the default values for the PC server.
Following are the relevant entries for GUI-Circulation:

1.28.1.1. setenv borname_style_u 01

1.28.1.2. setenv borname_style_s 02

1.28.1.3. setenv borname_style_f 03

Style for patron's name:

The letters denote the interface:

- U - Patron input
- S - Screen display
- F - Form

The numbers denote format

- 01 - name, title
- 02 - name
- 03 - title name

1.28.1.4. setenv overdue_delinq_period

Sets the threshold value for global block (Code 70). Code 70 is registered in the block field of the global patron record. It goes into effect for a patron who returns an item more than a certain number of days late. This happens automatically, so either comment out or remove this line if not required.

Possible values: 00-99

- 00 - code 70 is never attributed
- 01-99 - code 70 is attributed if an item is returned after being overdue for *nn* number of days.

1.28.1.5. setenv bor_list_address

Defines which address lines display.

Possible values:

- 00 - name only
- 02,03 - displays patron name together with the second and third lines of the address (the first line is the patron name, copied automatically from the name field)

1.28.1.6. setenv user_list_aleph_local

Defines whether the ADM patron is to be considered a Local Patron or not.

Possible values:

- Y – ADM level is considered a local patron.
- N – ADM level is not considered a local patron.

If set to N, patrons whose only Z305 local patron record is the ADM level record are not displayed in the Patron List if the checkbox "Display Local Patron Only" is selected. Therefore it is recommended to set user_list_aleph_local to Y.

1.28.1.7. setenv circ_user_z37_sort_routine

Defines the sort of the patron's hold list.

Possible values:

- 00- "standard" sort using priority, request date, open date & hour
- 01 - sort using sublibrary, item-status, collection, open date & hour
- 02 - sort using sublibrary, item-status, collection, status, open date and hour
- 04 - sort using group ID and sequence, priority, request interest date and request date

If the standard sort is sufficient then this line can be commented out.

1.28.1.8. setenv circ_user_z37_sort_order

Defines the sorting of the patron's Hold Request list.

Possible values:

- A - Ascending
- D - Descending

If the standard sort is sufficient then this line can be commented out.

1.28.1.9. setenv pc_z37_date_to.

Syntax is <x><nnn>, where the first position is the time unit code, [M]onths, [D]ays or [Y]ears, and the second is the numeric value. For example: M006 for six months. The variable determines the default period between the placing of a hold request from the GUI and the end of the period of interest. If this variable is not defined, or is defined incorrectly, the default "date to" will be three months.

1.28.1.10. setenv circ_user_z38_sort_routine

Defines the sort of the patron's photocopy request list.

Possible values:

- 00- "standard" sort using priority, request date, open date & hour
- 01 - sort using sublibrary, item-status, collection, open date & hour
- 02 - sort using sublibrary, item-status, collection, status, open date & hour

If the standard sort is sufficient then this line can be commented out.

1.28.1.11. setenv circ_user_z38_sort_order

Defines the sorting of the patron's photocopy request list.

Possible values:

- A - Ascending
- D - Descending

If the standard sort is sufficient then this line can be commented out.

1.28.1.12. 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 the patron defined as CATALOGER is created.

Possible values: 000-999

1.28.1.13. setenv default_lock_period

Locked circulation records are automatically unlocked after a period defined in this section .

Possible values: 000-999 - The period is defined in seconds.

Note 1: By default (if not otherwise defined), the variable has been set to lock records for 300 seconds.

Note 2: The value entered in setenv default_lock_period is also shared by Items, Acquisitions and ILL records.

1.28.1.14. setenv default_recall_type

This variable sets the default value for Recall Type field in the Hold Request screen in the Circulation module. Possible values:

- 01 = regular recall
- 02 = rush recall
- 03 = no recall

If no value is defined, the system defaults to "03" (= No Recall).

1.28.1.15. setenv display_lines_patron_list

This variable defines the number of lines that are loaded in the Patron List window when scrolling up and down. It can be used by the library to speed up the patron list display.

Possible values: 10-99. If no value is defined, the system defaults to 20.

If the value is defined above 99, the system sets to 99.

If the value is defined under 10, the system sets to 10.

29. Circulation Values of aleph_start

aleph_start is an ALEPH configuration file located under $alephe_root. This configuration file defines default values when starting the system. Following are the relevant entries for GUI-Circulation:

setenv fine_factor_fix
You can 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 (for example, 10,000 lira).

Sensible values: 10, 100, 1000

setenv fine_rounding

This rounds a 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.

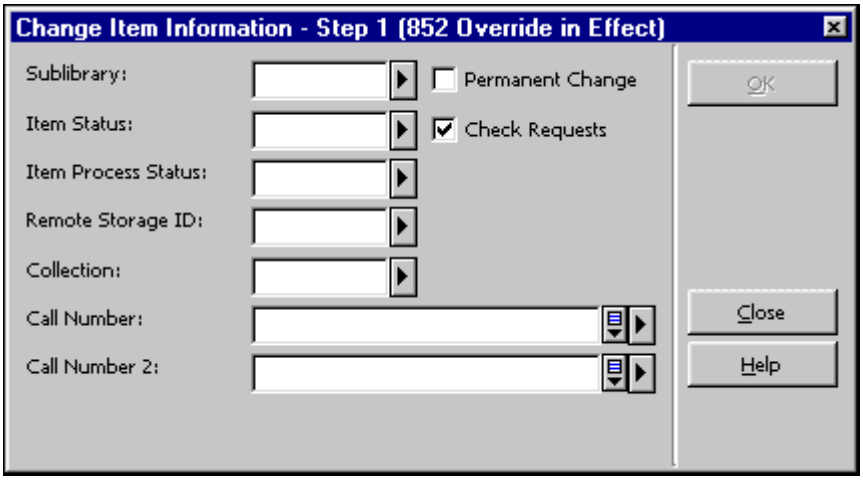
Possible values:

- Y - rounds sums up or down
- N - leaves decimal values

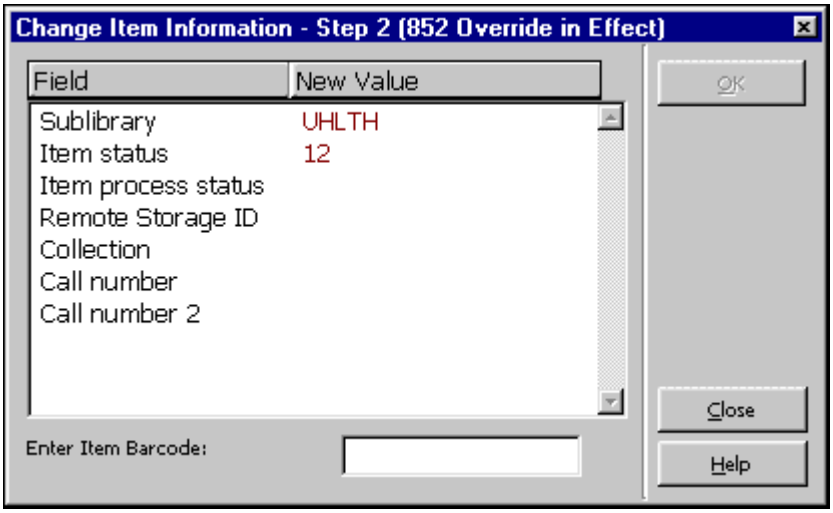
30. Change Item Information Display

You can change several parts of an item's information from the Change Item Information utility. You access the Change Item Information utility as follows:

1. From the main menu, click Items/Change Item Information. The following window opens:



2. Enter your changes and click OK. A second window (Change Item Information - Step 2) opens that displays the changes you want to effect.



The text in the first column ("Field") of this window is controlled by the table: `$aleph_root/error_lng/circ_item_change`. The order of the lines is hardcoded in order to match the order of the first window (Change Item Information - Step 1).

Note:
If you change an item status to an advance booking item status, advance booking slots are automatically created for two weeks. In order to open more slots as defined in www_server.conf and pc_item_schedule, run p-cir-60 after implementing the changes to the item status.

If you change an item status from an advance booking item status to another type of item status, the advance booking schedule of the item is automatically deleted.

31. Tables

31.1. Library Tables

1. **cash_09_translate**
   The table cash_09_translate translates sublibrary codes into campus or department codes for cash export reporting purposes in conjunction with the service Cash Export Report (cash-09).

2. **check_circ_override**
   The table defines the minimum password level that is required in order to be able to override circulation transactions that have been trapped.

3. **form_sub_library_address**
   form_sub_library_address table defines address types for Acquisitions, ILL & Circulation forms. form_sub_library_address works in conjunction with column 2 (address type) of `tab_sub_library_address.lng` (alephe table).

4. **pc_tab_circ_fast_cat.lng**
   pc_tab_circ_fast_cat.lng defines fields for the quick cataloging option, Catalog Item function, in the Circulation GUI module. The table in the BIB library defines the bibliographic fields. The table in the ADM library defines the administrative fields.

5. **pc_tab_fast_circ**
   pc_tab_fast_circ 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 that are connected to the main library via modem.

6. **tab100**
   tab100 is the central configuration table for system-level, server-level and library-level variables. See tab100-related entries in Circulation on page 111 and tab100's header.

7. **tab14**
   tab14 allows the library to indicate the time it takes for a returned item to get back on the shelf. The definition can be on the level of the sublibrary, collection and item status. When an item is returned, the due date will display "reshelving" (line 0357 in `$aleph_root/error_lng/www-heading` and line 8007 in `$aleph_root/error_lng/circ_status_holding`) for the time defined in `tab14`.

8. **tab15.lng**
   `tab15.lng` defines the item statuses and various parameters connected to these statuses.

9. **tab16**
   `tab16` defines the rules for computing due dates and fine rates for each combination of patron/item status. The table also defines limits placed on the number of hold requests and the maximum number of loans a patron with a certain status can have of items with a specific status or a global limit placed on a patron status not connected to a specific item status. Parameters for recalls are also held in this table. Note that the item process status is not taken into consideration when creating due dates and due hours (columns 6, 9, 16 and 18) but is taken into consideration for all other columns.

   Special Note: When a change is made to tab16 that affects due dates - in order for the changes to take immediate effect - the due dates table (Z301) should be dropped and recreated. Use either the service Rebuild Due Dates (cir-01) or UTIL A/17/1, Drop and Create Table and Index, to accomplish this. Note however, that if the table is not dropped and recreated via the service Rebuild Due Dates (cir-01) or UTIL A/17/1 Drop and Create Table and Index, the changes automatically take effect the next day.

10. **tab17**
    `tab17` defines the open days and hours of a library/sublibrary. It enables the system librarian to set the due date to the next open date and to skip "closed" days when computing fines, depending on the tab16 column 14 (fine method) setup.

11. **tab18.lng**
    `tab18.lng` defines the function against which Cash Control will be activated and the Cash amount. It is sensitive to sublibrary/patron status.

12. **tab24**
    `tab24` is used for Remote Storage. It defines the maximum time required for delivery of items from the RS to the pickup sublibrary and the set times that items sent from the Remote Storage arrive at the pickup sublibrary.

13. **tab25.lng**
    `tab25.lng` defines the list of possible values for Z30-MATERIAL (item material type). "ISSUE" has special functionality.

14. **tab27**
    `tab27` defines if an item that was requested by a patron through a hold, a booking or an ILL request is kept on the "hold shelf", is sent to a reading room, or is immediately loaned to the patron when it becomes available.

15. **tab30**
    `tab30` enables the library to define defaults for two elements regarding the expiration of patron registration.

    1. How many days before the expiration date do you want the system to warn you of approaching expiration (in CIRC). If a patron borrows an item within the defined period, the system gives a warning "expiration date approaching" together with the expiration date. The librarian can then make sure that the due date of the item will not fall beyond the expiration date.

    2. The default renewal period. This can be seen online when the renew button (on the Update Local Patrons screen) is activated.

16. **tab31**
    `tab31` defines default privileges 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. The status must be defined here in order to be valid in the system.

17. **tab_z31_sort**

    The table defines the sort procedures and sort orders (ascending or descending) for lists of cash records. The sort procedures will be displayed in a drop-down list that is defined in the pc_tab_exp_field.lng table under the menu CASH-SORT-TYPE.

18. **tab32**

    `tab32` defines overdue letter intervals for patron/item statuses.

19. **tab34**

    `tab34` is an optional table that computes the charges that are added to the patron's cash record in case an item is changed to lost through the LOST function in the Circulation module, or as a result of Overdue and lost letter. The charges are defined for each item-patron combination.

20. **tab37**

    `tab37` defines the configuration for hold requests. This configuration includes:

    - Target library for hold request
    - Item status
    - Patron status
    - Pickup library

    If the pickup location is not defined the request will not be allowed.

21. **tab37_booking_pickup**

    This table defines the pickup locations allowed for booking requests. Which locations are allowed depend on item status, item process status, patron status, material type and sublibrary of item. The list of locations are valid sublibraries in `tab_sub_library`.

22. **tab37_booking_delivery**

    This table defines the delivery locations allowed for booking requests. Which locations are allowed depends on item status, item process status, patron status, material type, and the sublibrary of item. The list of locations is made up of valid locations in tab_delivery_locations.lng. Note that a line with hashes in columns 1-5 defines the list of delivery locations that will be shown wherever a drop-down list of delivery locations is shown in the GUI.

23. **tab37_campus_filter**
    tab37_campus_filter is used in conjunction with the service Hold Requests Router (cir-24). This table allows you to remove hold requests that have certain characteristics from being considered by the Hold Requests Router (cir-24).

24. **tab37_campus_preferred**
    tab37_campus_preferred is used in conjunction with the service Hold Requests Router (cir-24). This table allows you to determine the order of preference for supplying items that have been requested to a certain pickup location.

25. **tab38**
    tab38 defines the configuration for photocopy requests. This configuration includes:

    - Target library for photocopy request
    - Item status
    - Patron status
    - Pickup library
    - Delivery method

    If the pickup location is not defined the request will not be allowed.

26. **tab39**
    tab39 supports the configuration of the daemon for handling hold requests (ue_06). This includes the capability to print requests on different printers.

27. **tab40.lng**
    tab40 defines collection codes and names. It defines the expansion of the collection code as entered in the Z30-COLLECTION FIELD in items, to the expanded form to be displayed in the Web OPAC.

28. **tab41**
    tab41 supports the configuration of the daemon for handling photocopy requests (ue_06). This includes printing requests on different printers.

29. **tab43**
    tab43 defines the setup for advance booking of short loans. It is a table for defining advance booking parameters. Tab43 includes definitions for the start day and hour, as well as end day and hour. The table

enables you to define "closed" times, independent of `tab17` (library hours).

30. **tab44**
    `tab44` defines how long requested items will be kept on the hold shelf.

31. **tab_attr_sub_library**
    tab_attr_sub_library defines (by IP address or station ID) circulation loan/return authorizations. In other words, which sublibraries will accept loans/returns of other sublibraries and which will not.

32. **tab_block_circ**
    This table is the default table used to determine the alternative block values for checks 9_a, 9_b, 10_a and 10_b in `tab_check_circ`. 9_a and 10_a are for checks at the ADM level, 9_b and 10_b are for checks at the sublibrary level.

33. **tab_booking**
    This table defines the following times, required for booking requests: Head time, tail time, release time, delete interval, future booking limit.

34. **tab_bor_address**
    `tab_bor_address` provides an option for defining the validity period of a patron address.

35. **tab_bor_id.lng**
    `tab_bor_id` defines all system-wide behavior relating to the Z308 (patron's ID) records. In this table, you can define the keys that can be used to access patron records in the GUI applications and in the Web OPAC. You can also define the additional IDs that can be updated in the GUI, their verification codes, and whether the ID is displayable. The table is managed from the tab directory of the user library.

36. **tab_check_circ**
    `tab_check_circ` defines which check routines to run when activating loan/renewal.

37. **tab_circ_log.lng**
    `tab_circ_long.lng` is used to configure which events will be logged by the circulation logger, and which events can be manually added to the logger. Note that the texts of the logged events are set in `$aleph_root/error_lng/circ_logger`.

38. **tab_delinq.lng**
    tab_delinq.lng allows the library to define which circulation privileges will be blocked and/or restricted for a defined block code (in the Global Patron Information form or the Local Patron Information form).

39. **tab_delivery_locations**
    This table is where the delivery locations that are allowed by this ADM are defined. The table lists the delivery locations and their addresses. These delivery locations can be defined in `tab37_booking_delivery` as delivery locations of booking requests.

40. **tab_events.lng**
    `tab_events.lng` of the administrative library defines which transactions will write a record to the Z35 events Oracle table. The transactions relate to Circulation (loans, returns, holds, photocopy requests). The event type identifier is written in Z35-EVENT-TYPE of the Z35 table. The `tab_events.lng` table of the bibliographic library also has Web OPAC-related event types that write records to the Z69 Oracle table (types 20-35) and to the Z35 oracle table (types 10, 11, 12).

41. **tab_fast_patron_register**
    This table is found in the `data_tab` directory of USR00. The table defines the default values that will be used for creating the following patron records fields, when the Fast Patron Registration option is used :

    Z303-ILL-TOTAL-LIMIT

    Z303-ILL-ACTIVE-LIMIT

    Z303-EXPORT-CONSENT

    Z303-SEND-ALL-LETTERS

    Z304-DATE-TO (for address type "01")

    Z305-BOR-TYPE

    Z303-PLIF-MODIFICATION

    This table contains lines in the format "FIELD=VALUE", for example:

```
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
Z303-PLIF-MODIFICATION = A B D E 1
```

    The value of Z304-DATE-TO can be in one of the following formats:

    If the value of the Z304-DATE-TO field in the table is eight digits only (for example Z304-DATE-TO=20041231), this value will

be translated to be the 'valid to' value of the new patron's address. In this case it is a simple date of the form yyyymmdd.

If the value of the Z304-DATE-TO field in the table is eight digits suffixed by an upper case letter, the value will be translated to be an offset from the current date. If the suffix is Y the offset will be in years, if it is M the offset will be in months, and if it is D the offset will be in days.

For example:

Z304-DATE-TO=00000002Y means that 2 years will be added to the current date and stored in the field Z304-DATE-TO of the newly created patron.

The Z304-ADDRESS-TYPE sets the type of address that will be created. Possible values are 01,02,03 etc.

The Z303-PLIF-MODIFICATION field sets which fields should be protected when updating borrower records using the PLIF - Patron Loader (p-file-20) service. If you want all fields to be unprotected, leave the z303-PLIF_MODIFICATION= line blank.

The field can contain a list of codes delimited by a space. The codes are:

- A = Do not modify the borrower's address records (Z304).
- B = Do not modify the borrower's status.
- D = Do not modify the borrower's expiry date.
- E = Do not modify the borrower's e-mail address (Z304-EMAIL-ADDRESS) in the address record (Z304).
- 1 = Do not modify all borrower's records (Z303, Z304, Z305 and Z308).

42. **tab_hold_request**

`tab_hold_request` is used to define which checks should be performed by the system when a hold request is placed. Definitions in the table determine:

- If the hold request link should be displayed in the WWW OPAC ("PRE" checks).
- Which checks should be performed when the request is actually placed ("POST").

- Which checks should be performed in the circulation client (CIRC).

- In which cases information should be displayed when a request is initiated in the WWW OPAC (INFO).

43. **tab_hold_request_form**

    `tab_hold_request_form` 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 variable in `tab100`.

    If no match is found, or `tab_hold_request_form` does not exist, the system will display the hold request forms without the suffix.

44. **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 in the History node of the Items tab in the Cataloging 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 node.

45. **tab_label.lng**

    `tab_label.lng` defines the prefixes that will be printed on the spine label.

46. **tab_late_return**

    `tab_late_return` defines the periods and sums for late return fines in conjunction with fine method F in tab16, column 14.

47. **tab_map_privileges**

    This table defines the type of global ALEPH Z305 record that is created when the patron locally registers in a sublibrary. This table will be functional only in cases where column 22 of tab31 is configured to Y, meaning that an ALEPH record is to be created.

48. **tab_photo_request**

    `tab_photo_request` 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 photocopy requests for the Web OPAC determine whether the "photo" link appears on the item line.

49. **tab_roster**

    This table determines how the roster of potential suppliers will be created when a title level hold request is submitted. Each pickup location is assigned groups of potential suppliers that can fulfill the request. The order in which the potential suppliers are listed in this table is the order in which they will be created in the title request roster of the potential suppliers. This table can be defined either in the data_tab of the pickup location ADM library or in alephe_tab.

50. **tab_service_hour**

    This table defines the service hours for hold and photocopy requests. The request submission time (day+hour) is checked against the pickup location service hours that are set in tab_service_hour. It lists service hours for days of the week (00-Sun thru 06-Sat) or a specific date (YYYMMDD).

    Note that the system applies tab_service_hour only when check_hold_request_sh and check_photo_request_sh are set in tab_hold_request and tab_photo_request respectively. Only days+hours defined in this table are considered service hours. Days+hours that are not set are labeled as non-service hours. If the tab_service_hour table is empty (with no lines set) or does not exist, request services are off (there are no available service hours).

    If a request is submitted when the pickup location is out-of-service, the request is not accepted and a message is displayed indicating that the pickup location is currently out of service and that the request cannot be submitted. For setup and additional information, see Request Service Hours Checks (tab_service_hour) on page 50.

51. **tab_sub_library_address.lng**

    `tab_sub_library_address.lng` defines addresses of sublibraries for printouts (for example, orders to vendors, overdue letters to patrons, and so on.)

    Note: This table can be defined either in the `tab` directory of `alephe` library or in the tab directory of each ADM library.

52. **tab_ue_pickups**

    tab_ue_pickups lists the photocopy pickup locations that are handled by the requests daemon UE_06. Photocopy pickup locations that are not listed in the table are skipped by the requests daemon.

53. **tab_z30_sort**

    `tab_z30_sort` defines sort options for items. Different sort routines can be used in different parts of the system.

54. **tab_z31_sort**

    This table defines the sort procedures and sort orders (ascending or descending) for lists of cash records. The sort procedures are displayed in a drop-down list that is defined in the `pc_tab_exp_field.lng` table under the CASH-SORT-TYPE menu.

55. **tab_z36_sort**

    This table defines the sort procedures and sort orders (ascending or descending) for lists of loan records in various instances. The sort procedures are displayed in a drop-down list that is defined in the `pc_tab_exp_field.lng` table under the LOAN-SORT-TYPE menu.

56. **tab_z38_sort**

    This table defines the sort procedures and sort orders (ascending or descending) of the Patron's Photocopy Request list. The sort procedures are displayed in a drop-down list that is defined in the `pc_tab_exp_field.lng` table under the PHOTO-SORT-TYPE menu.

57. **user-cash**

    `user-cash` defines the information fields that will be displayed in the Cash pane of the Circulation module.

31.2. alephe Tables

library_relation
`library_relation` defines relation between libraries by library type.

tab_days
`tab_days` defines of the days of the week in different languages. The table is used in the Web OPAC for the short loan reservations.

tab_ip_sub_library
This table is now called `tab_attr_sub_library`.

tab_month
`tab_month` defines 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.

tab_roster

This table determines how the roster of potential suppliers will be created when a title level hold request is submitted. Each pickup location is assigned groups of potential suppliers that can fulfill the request. The order in which the potential suppliers are listed in this table is the order in which they will be created in the title request roster of the potential suppliers. This table can be defined either in the data_tab of the pickup location ADM library or in alephe_tab.

tab_sub_library.lng
`tab_sub_library.lng` defines the sublibrary codes of all the libraries on the server or in a network of servers.

tab_sub_library_address.lng
`tab_sub_library_address.lng` defines addresses of sublibraries for printouts (for example orders to vendors, overdue letters to patrons, and so on.)

Note: This table can be defined either in the tab directory of the alephe library or in the tab directory of each ADM library.

32. tab100-related entries in Circulation

`tab100` is the central configuration table for system-level, server-level and library-level variables. A few lines of the table are shown below:

```
!!!
```

```
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
CHECK-UNIQUE-NAME-BIRTH=Y
CHECK-ORDER-ISBN-ISSN=N
CREATE-ITM-FORM-ORDER-M=Y
BARCODE-DELETE-SPACES=Y
BOR-EXPIRY-DUE-DATE=Y
OVERDUE-LETTER-STYLE=Y
Z30-PRICE-FROM-ORDER=N
OVERDUE-LETTER-NO=1
CHECK-BARCODE=Y
MARC-TYPE=1
FAST-CAT-HOLD-ID=
BOOKING-DELIVERY-TIME=D
NOTE-ONPICKUP-ONLY
```

- ADDITIONAL-ID-TYPE

  This variable is related to fast patron registration. The type of ID that is added in the 'Additional ID' field is set in this variable.

- AVAILABILITY-ROUTINE

  Determines whether an item is available or is on loan, on hold or has a hold request placed on it.

  Possible values are: 1 or 2.

  1 = item is considered non-available if it is on loan, on hold, or has been requested (Z37-SOURCE is "C" or "O").

  2 = (site specific) item is considered non-available if it is on loan, on hold, or has been requested (Z37-SOURCE "O" only).

  The default value is 1.

- BARCODE-DELETE-SPACES
  Determines whether or not spaces are deleted when the barcode data is transferred to the system.

  Possible values are: Y or N.

  Y = deletes spaces when barcode data is transferred to the system. This refers to the item barcode only.

  The default value is Y.

- BLOCK-RATIO

  Defines the factor by which the system multiplies the late days for a borrower that has returned an overdue item in order to calculate the number of days on which the patron will be blocked from performing other loans. This is applicable or fine methods 5, 6, 8 and 9.

  Possible values are 1-9.

  The default value is 1.

- BOOKING-DELIVERY-TIME

  Defines the delivery time that will be added to a booking request that is required for delivery.

  Possible values: The value is a three-digit number that can express minutes, hours, days or weeks, depending on the variable's prefix. The prefix can be 'M' for minutes, 'H' for hours, 'D' for days or 'W' for weeks.

- BOR-EXPIRY-DUE-DATE
  This variable controls the shortening of due dates according to the patron's expiration date.

  Possible values are: Y, N, or H.

  Y = sets the due date for loan and renew and the end hold date for hold request to the local patron expiry date if the local patron expiry date is earlier than the computed due date.

  H = sets the due date for loan and renew to the local patron expiry date if the local patron expiry date is earlier than the computed due date.

  N = the due date is not adjusted.

  The default value is N.

- BOR-STATUS-FOR-RENEWAL
  Defines the patron status when an item is loaned or at the time the loan is renewed.

  Possible values are: L or C.

  L = when an item is renewed, the patron status at the time of the original loan is used. This is the default if the variable is left blank.

  C = when an item is renewed, the patron's current status (at the time of renewal) is used, and the Z36-BOR-STATUS is updated accordingly.

  This variable works whether the item is renewed through the Circulation module or through the Web OPAC.

  The default value is Y.

- CHECK-BARCODE

  Determines whether the barcode for the item record is checked or disabled.

  Possible values are: Y or N.

  N = disables (the default) check for barcode on the item record, thereby allowing item records without barcodes. If the library opts to disable the barcode check for items, the z30_id1 entry in the file_list must be adjusted accordingly, since it assumes that bc is an index, and will not allow a non-unique index key ("blank" is an index key).

  The default value is Y.

- CHECK-BOR-DEFAULTS

  Determines whether or not there is a discrepancy in the values set in the local patron record and the values set in tab31 of the patron status.

Possible values are: Y or N.

Y = when updating Z305 (local patron record) in 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, a warning (which can be overridden) is displayed.

The default value is Y.

- CHECK-UNIQUE-NAME-BIRTH

  Determines whether the combination of name and birthdate is unique.

  Possible values are: Y or N.

  Y = in patron Z303 record the combination of name and birthdate must be unique. Umlaut letters are considered to be identical to their non-umlaut values for this purpose. The non-umlaut value depends on the translation of the umlaut in the Unicode table.

  The default value is Y.

- CREATE-Z30H

  Determines when to create an item history record when an item is updated or deleted. There can be multiple Z30H records for a single item.

  Possible values are: Y or N.

  Y= create Z30H (ITEM HISTORY) record when item is updated or deleted, if tab/tab_item_history.lng is set for creating a history record. There can be multiple Z30H records for a single item; a Z30H record can be reinstated as the current record.

  The default value is Y.

- CREATE-Z36H

  Determines whether or not the system stores information about previous loans and displays them in the Item History window which is accessible from the Item List. For each loan, a history record (Z36H) is created when the loan record (Z36) is deleted.

  Possible values are: Y or N.

  Y= transfer completed loan records to a history file.

  The default value is Y.

- CREATE-Z37H

  Determines whether or not the system stores information about previous hold requests and displays them in the Item History window which is accessible from the Item List. If you want the system to store the information, enter *Y*. For each hold request, a history record (Z37H) is created when the hold request record (Z37) is deleted.

  Possible values are: Y or N.

  Y= transfer completed loan records to a history file.

  The default value is Y.

- DEFAULT-BOR-ID

Determines the default value of the Borrower ID.

Possible values are: Y or N.

Y = create default field value, using last-bor-id counter from z52.

The default value is Y.

- DEFAULT-ADDRESS

  Determines which address type and validity range will be used as default when a new patron is created. The variable format is TT DDD, meaning:

  - TT - Two digits representing the default patron address Type
  - DDD - A three digit number, representing the number of days for which the default address is to be valid after its creation date. When this variable is set to 999, it is interpreted by the system as permanently valid.

  For example, setting the variable to 03 090 means that the default address type is 03 and it is valid for 90 days.

  The variable's default is 01 030.

- DEFAULT-BOR-ID-1

  Determines the default value of the Borrower ID.

  Possible values are: Y or N.

  Y = create default field value, using last-bor-id-1 counter from z52 (UTIL G/2). The default value is Y.

- DEFAULT-BOR-VERIFY-1

  Determines the default field value for the Borrower ID.

  Possible values are: Y or N.

  Y = create default field value, using last-bor-verify-1 counter from z52.

  The default value is Y.

- DEFAULT-BOR-VERIFY-2

  Determines the default field value for the Borrower ID.

  Possible values are: Y or N.

  Y = create default field value, using last-bor-verify-2 counter from z52 (UTIL G/2).

  The default value is Y.

- EMAIL-VALIDATION
  Determines whether or not a patron's e-mail validation check is performed when it is populated.

  Possible values are: Y or N.

  Y = the system performs a patron's e-mail validation check and displays a message if the e-mail address is not valid (warning message, not a block). The check is activated when the patron address is updated and the e-mail field is

populated. It is performed when the address is updated via GUI, OPAC, PLIF loader, DLF-API, and X-services.

The default value is N.

If the flag is set to Y, the validation includes:

1. Syntax check

   a. Existence of only one “at” symbol (@).

   b. Existence of a dot after the “at” symbol (@).

   c. No space in the e-mail address.

2. Domain existence check - a check confirming that the remote server exists.

- ENCRYPT-Z308

  Determines whether or not the patron identifying numbers are encrypted.

  Possible values are: Y or N.

  Y = indicates that the patron identifying numbers are encrypted.

  The default value is N.

- FAST-CAT-HOLD-ID

  Determines the ID that will be used for automatically creating a hold request on a fast cataloged item in Circulation.

  Possible values are all 12 uppercase letter IDs that are valid patrons in the ADM.

  If this entry does not exist then:

  If the library uses user-sharing then the ID will be "CATALOGER".

  If the library does not use user-sharing then the ID will be a concatenation of "CAT-" and the library name.

- HOLD-FULFILL-NOTIFY

  This variable sets the system behavior when an item is transferred from the owning library to the pickup library.

  Possible values are 1,2.

  1 = Notify the patron about the request fulfillment only when the item arrives at the pickup location.

  2 = Notify the patron about the request fulfillment when the item is sent from the owning library.

  The default value is 1.

- HOLD-REQ-PROCESS-STATUS

  Determines which "like copies" can satisfy a hold request.

  Possible values are: Y or N.

  Y - Only items with the same process status as the item on which the hold request was made can satisfy the request.

N - All items can satisfy the request, regardless of their process status.

The default is Y.

- HOLD-REQUEST-COLLECTION

  Determines whether the system will try to fill a hold request with a copy from the same Collection as the copy selected in the Item List.

  Possible values are: Y or N.

  Y = If you enter Y, the system will fill in the Collection field of the Hold Request form with the same Collection as the copy selected in the Item List. Only copies with the same collection will satisfy the request.

  N = If you enter N, the system will enter the # character in the Collection field to indicate that it does not matter which Collection is used to fill the request.

  The default value is Y.

- HOLD-REQUEST-ITM-STATUS

  This table determines whether the system will try to fill a hold request with a copy having the same item status as the copy selected in the Item List.

  Possible values are: Y or N.

  Y = If you enter Y, the system will fill in the item status field of the Hold Request form with the same item status as the copy selected in the Item List.

  N = If you enter N, the system will enter the # character in the item status field to indicate that it does not matter which item status is used to fill the request.

  The default value is Y.

- ITEM-BARCODE-OR-CALL-NO

  Disables or enables item retrieval by call number when using the Item Bar (Circulation and Cataloging GUI).

  Possible values are: Y or N.

  Y - Item retrieval by call number is enabled (in the Item Bar in Circulation or Cataloging GUI).

  N - Item retrieval by call number is disabled (i.e. there is an attempt to retrieve the item by barcode; if it fails, no further attempt is made to retrieve by call number).

  The default value is Y.

- ITEM-BOR-INFO-CIRC-ONLY

  Determines whether or not circulation information that is displayed in modules other than the Circulation module will include patron information. This includes current loan information and loan and hold history information that is displayed in the Cataloging and Acquisitions modules.

  Possible values are 'Y' and 'N':

  Y - Do not display patron information outside the Circulation GUI.

  N - Display patron information outside the Circulation GUI.

  The default value is 'N'.

- LATE-RET-FINE-WHEN-LOST=N
  This variable is used to set up the library's policy with regard to lost items and overdue charges.

  Y = When an item is declared lost, charge any overdue fines that have been accrued up to this point of time. In addition, charge the lost-item charges according to the regular tab34 definitions. If the item is later returned, refund the lost-item charges according to the regular LOST-LOAN-CREDIT-METHOD and REFUND-RATE tab100 variables policy. Do not create any new overdue fines at this point.

  N = When an item is declared lost, do not charge any overdue fines that have been accrued up to this point of time. Do charge the lost-item charges according to the regular tab34 definitions. If the item is later returned, refund the lost-item charges according to the regular LOST-LOAN-CREDIT-METHOD and REFUND-RATE tab100 variables policy. In addition, create overdue fines at this point for the entire overdue period, including the period of time in which the item was declared lost.

- LOST-LOAN-CREDIT-METHOD

  Determines which cash transactions are credited when a lost loan is returned.

  Possible values are:

  1 = when a lost loan is returned (that is an item which has a Z36 record with status "L"), all associated cash transactions except those with "waived" status are credited. This means "C"losed, "T"ransferred or "O"pen transactions are credited. In addition, partially waived transactions are not credited at all.

  2 = when a lost loan is returned, all "T"ransferred or "O"pen transactions are credited. Closed transactions are not credited automatically. In addition, partially waived or partially closed transactions are fully refunded, regardless of the tab100 variable REFUND-RATE.

  3 = like 1 but REFUND-RATE only applies for "C"losed transactions. Other transactions are fully refunded.

  4 = like 1 except that Open transactions are credited by waiving the lost bill debit transaction.

  5 = like 2 except that Open transactions are credited by waiving the lost bill debit transaction.

  6 = like 3 except that Open transactions are credited by waiving the lost bill debit transaction.

  The default value is 1.

- LOST-PROC-STATUS

  Determines the process statuses that will be removed from an item if the item’s barcode is read in a circulation action.

The format of the variable is 'P1 P2 P3', where 'P1', 'P2' and 'P3' are item process statuses.

Default Value: Spaces

- OFFLINE-CASH-TRANS

  Determines whether or not offline circulation registers cash transactions. This depends on the OFFLINE-CASH-TRANS variable.

  Possible values are: Y or N.

  Y = The system registers cash transactions triggered by circulation transactions effected through offline circulation.

  N = The system does not register cash transactions triggered by circulation transactions effected through offline circulation.

  The default value is Y.

- ONLINE-RECALL

  This variable sets the library policy regarding online recall.

  Possible values are: Y or N

  Y = any request that is created in the GUI and is required for recall will trigger a recall when the request is submitted.

  N = a recall will be triggered only when the 'Recall Items on Loan (cir-13)' service is run.

  The default value is N.

- OVERDUE-LETTER-NO

  Defines the number of overdue letters sent to be considered as a block in circulation functions.

  Possible values are: 0, 1, 2, 3 or 4.

  Defines "overdue" for tab_check_circ_1 block. A loan is overdue if the due date has passed and "n" or more notices have been sent.

  The default value is 0.

- OVERDUE-LETTER-STYLE

  Determines whether or not an overdue notice should be standard or sensitive to the letter number.

  Possible values are: Y or N.

  N = standard overdue notice, without special text.

  Y = overdue notice text is sensitive to letter number.

  The default value is N.

- OVERDUE-RECALL-RATIO

  Defines whether the values in col.5 of lines 0050-0055 in tab18.lng (fines relating to recalled items) are to be treated as a ratio, as an actual amount or as a combination of both.

Possible values are: Y, N or B.

Y or blank = values are ratio multiplicator of the basic fine set in col.11 of tab16.

N = values are actual amount per day

B = Both - values are actual amount added to the basic fine set in col.11 of tab16.

Example:

Using fine method 7 (fine for total days including recall fining) in col.14 of tab16:

OVERDUE-RECALL-RATIO set to Y:

```
col.4 tab18:      Y
col.5 tab18:      2
col.11 tab16:   0.50
```

The daily fine charged is $0.50 after the due date. After the item is recalled, the daily fine is still $0.50. After the recall due date has passed, the daily fine is 0.50x2=$1.

Using fine method 7 (fine for total days including recall fining) in col.14 of tab16:

OVERDUE-RECALL-RATIO set to N:

```
col.4 tab18:      Y
col.5 tab18:      2
col.11 tab16:   0.50
```

The daily fine charged is $0.50 after the due date. After the item is recalled, the daily fine is still $0.50. After the recall due date has passed, the daily fine is $2.

Using fine method 7 (fine for total days including recall fining) in col.14 of tab16:

OVERDUE-RECALL-RATIO set to "B":

```
col.4 tab18:      Y
col.5 tab18:      2
col.11 tab16:   0.50
```

The daily fine charged is $0.50 after the due date. After the item is recalled, the daily fine is still $0.50. After the recall due date has passed, the daily fine is 0.50+2=$2.50.

The default value is Y.

- PATRON-LIST-ID-TYPE

  Determines which Z308 key will be used for building the ID in the patron list (Z353). In this way, additional IDs can be used as IDs in the patron list instead of the Z303 ID.

  The possible value of this variable is any two-digit number that is a valid Z308 type in the system.

  By default, the system uses the Z303-ID as the patron list ID.

- PHOTO-REQUEST-STYLE

  Determines whether the system will delete the photocopy request after the slip is printed. If the slip is printed by ue_06 or by cir_22, the request is not deleted but its status is changed to H.

  Possible value is H.

If you leave the adjacent column blank, then the photocopy request will remain in the system and the librarian will be able to determine and record the number of pages used for photocopying and the corresponding fee.

H = If you enter the letter H, the request will be deleted and no follow-up will be possible.

There is no default value.

- PHOTO-SLIP-STYLE

  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 variable relates to the list of items that are included in the print slip.

  Possible values are: 1, 2 or 3.

  1 = list all "like" items in the photocopy request slip. This means that availability is not checked.

  2 = list only available "like" items in the photocopy request slip. This implies that availability is checked.

  3 = list the first of the available "like" items in the photocopy request slip. This implies that availability is checked.

  The default value is 1.

- PICKUP-LIST-METHOD

  This variable controls how the pickup list will be created when a title request form is loaded.

  Possible values are 1,2,3,4

  1 = Create a joint pickup list that is based on the pickup locations of all of the available items that are linked to the requested record.

  2 = Only the dispatch library that is defined for the patron is a valid pickup location.

  Values 3 and 4 are reserved for future use.

  The default value is 1.

- PICKUP-SORT-BY-ITEM

  When presenting pickup location lists for hold requests or photocopy requests, the patron's home library is always used as default. This variable defines which sublibrary will be used as the default when no home library is defined.

  The possible values are Y and N:

  Y = When Z303-HOME-LIBRARY is not defined, the item’s sublibrary will be the default pickup location.

  N = When Z303-HOME-LIBRARY is not defined, the default pickup location will be the first sublibrary in the tab37 or tab38 setup.

- RECALL-METHOD

  Determines whether or not renewals and recalls will update the due date.

Possible values are: 1, 2 or 3.

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 is later.

The default value is 1.

- REFUND-RATE <x> <y> <z>

  This variable determines the refund-rate for lost item handling/replacement/notice charges.

  When an item that has been declared lost is returned by the patron, the charges that were made when the item was declared lost are refunded. Using this variable can set what percentage of each charge will be refunded.

  Possible values are three numbers separated by a single space, each one up to three digits long. The numbers are the refund rates of the handling cost (transaction number 40), the replacement cost (transaction number 41) and the notice cost (transaction number 42) respectively. A value of 0 means that no refund will be made, and a value of 100 means that a full refund will be made.

  Default values are 100 100 0.

- RENEW-DURING-LOAN

  This variable controls the behavior of the system when trying to effect a loan on an already loaned item.

  Possible values are: Y or N.

  When trying to effect a loan for an item that is already on loan for the same patron the system consults RENEW-DURING-LOAN.

  When trying to effect a loan for an item that is already on loan for another patron the system consults RENEW-DURING-LOAN.

  Y = If RENEW-DURING-LOAN is Y the system renews the loan for this patron.

  N = If RENEW-DURING-LOAN is N the system consults RETURN-DURING-LOAN.

  The default value of RENEW-DURING-LOAN is Y.

- RESET-LETTER-NUMBER

  This variable sets whether an online recall of an item will reset the loan's Z36-LETTER-NUMBER field.

  Possible values are: Y or N

  Y = online recall of an item will reset the loan's Z36-LETTER-NUMBER field

N = online recall of an item will not reset the loan's Z36-LETTER-NUMBER field

Default value is N

- RETURN-DURING-LOAN

  This variable controls the behavior of the system when trying to effect a loan for an item that is already on loan for another patron.

  Possible values are: 0, 1 and 2.

  0 = when an item is being loaned, and the system detects that the item is currently on loan, a message is displayed indicating that the item must 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.

  The default value is 0.

- SHOW-USR-VERIFICATION

  Determines whether or not to display the patron verification fields in the GUI Circ and ADMIN clients.

  Possible values are: Y or N.

  Y = display patron verification fields when updating Z303 (Patron) in GUI CIRC.

  N = mask out view of patron verification fields when updating Z303 (Patron) in GUI CIRC.

  The default value is N.

- STORE-CALL-NO-DOLLAR

  When an item's information is overridden by a temporary change, the original information is saved. This variable defines what string will replace a $ sign in the call number when the call number field is saved. We suggest you use a string that will not appear as part of the call number data.

  Possible values are all characters, making up to a 10-character long string.

  The default value is the string U+0036.

- SUB-LIBRARY-DIVISION

  Determines whether the system will try to fill a hold request with a copy from the same sublibrary as the copy selected in the Item List.

  Possible values are: Y or N.

  Y = If you enter Y, the system will fill in the sublibrary field of the Hold Request form with the same sublibrary as the copy selected in the Item List.

  N = If you enter N, the system will enter the # character in the sublibrary field to indicate that it doesn't matter which sublibrary is used to fill the request.

The default value is: N.

- TIMEZONE-ADJ

  This variable is used to set the "current time" of a library in relation to the server time. The variable's format is a two digit number, prefixed by either a '+' sign or a '-'sign. The number is the difference in hours between the server time and the time zone of the ADM library, and the sign sets whether the ADM library is ahead (+) or behind (-) the server time. This adjustment affects circulation-related activities in an ADM library.

  The default value is +00.

- UPDATE-RECALL-DATE

  This variable determines if, when an item is loaned, and the item already has a recall hold request, the loaned item should immediately be considered "recalled", or whether a loaned item becomes "recalled" only through the Recall Items on Loan service (cir-13).

  Possible values are: Y or N.

  Y = item loan can trigger "recall". The z36-recall-date and z36-recall-due-date are automatically set at the time of the loan if there is a matching "O" (Original copy/Only selected item) request.

  N = recall of a loaned item (update of z36-recall-date and z36-recall-due-date) is triggered only by cir-13.

  The default value is N.

- USER-ADDR-PERMISSION

  This variable defines which types of patrons can have their addresses updated by the staff user.

  Possible values are: 0, 1, or 2

  0 - Update of patron's addresses is based on standard permissions. Any staff user that has privileges to update and view a patron's addresses can view and update the patron address.

  1 – Staff user can update patrons' addresses only if they have one or more Z305 records in the ADM.

  2 – Staff user can update patrons' addresses only if they have one or more Z305 records other than the ADM level record.

  The default value is 0.

  Note that the patron's address can be viewed even if it cannot be updated.

- USER-HOME-PERMISSION

  Determines whether or not the system checks the staff user library (z66_user_library) against the patron's home library, when a staff user modifies patron records (Z303, Z304, Z305)

  Possible values are: Y or N

Y- A check is performed on the staff user's library (z66_user_library) against the patron's ADM library (derived from the patron's home library field).

The staff user is only permitted to modify the patrons records (Z303, Z304, Z305) if a match is found.

If the patron's Z303 record does not include a value in the Home library field, it can be modified by any staff user (z66_user_library field, may include any value)

N - No check is performed and any staff user (with any value in the z66_user_library field) can modify the patrons record (Z303, Z304, Z305)

The default value is N.

- USER-IDS-PERMISSION

  This variable defines which types of patrons can have their additional IDs viewed and updated by the staff user.

  Possible values are: 0, 1, 2 or 4.

  0 - Update of patron IDs is based on standard permissions. Any staff user that has privileges to update and view a patron's ID can view and update the patron's IDs.

  1 – Staff user can view/update patron's IDS only if the patron has one or more Z305 records in the ADM.

  2 – Staff user can view/update patron's IDS only if the patron has Z305 records other than the ADM level record.

  4 - The staff user can view and update the patron's IDs only if he is connected to a library that includes one of the sublibraries in the patron's list of locally privileged libraries.

  The default value is 0.

  Note that if the patron's IDs cannot be updated by the staff user, they will not be viewable either.

- USER-PERMISSION

  This variable defines which types of patron records can be updated by the staff user.

  Possible values are: 0,1 or 2

  0 - Update of global patron records is based on standard permissions. Any staff user that has privileges to update global patron records can update the patron's record.

  1 - Staff user can update Z303 only if it has one or more Z305 records in the ADM.

  2 - Staff user can update Z303 only if it has one or more Z305 records other than the ADM level record.

  The default value is 0.

  Note that patron records are viewable even when they can not be updated.

- USER SHARING

The User Sharing variable determines whether the patron records participate in the shared or non-shared section of the database (USR00).

Possible values are: Y or N.

Y = If the ADM library is defined as USER-SHARING=Y, then the patrons created by this library will be displayed in the Patron List when a staff user is connected to ANY library that is defined as "Y" in the USER SHARING variable.

N = Patrons belonging to an ADM library which defines the variable USER-SHARING=N in tab100 will not be included in this Patron List. If the ADM library defines USER-SHARING=N, then the patrons created by this library will be displayed in the Patron List only when a staff user is connected to this specific library.

The default value is N.

- NOTE-ONPICKUP-ONLY

  Determines whether the message "Item(s) on Hold for this Patron" will be displayed if a patron has hold requests and the work station is in a different sub library than the item's pickup sub library.

  N= display the message, regardless of the sublibrary to which the item belongs.

  Y= default. Display only if the patron's on-hold item is on the hold shelf that belongs to the sublibrary of the circulation desk station.

33. Circulation Values of www_server.conf

setenv www_item_schedule 14
Defines the number of days that a patron will be able to book an item in advance. For example, if the parameter is set to 14, a patron will be able to book an item up to 14 days in advance.

setenv www_z37_recall_type 03
Defines 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 valid recall types are 01, 02 or 03, where 01 is regular recall, 02 is rush recall and 03 is no recall.

Recall is related to item status in tab15, col.13 and 14, according to which an item can be recalled or rush recalled. This relates only to the Recall Items on Loan (cir-13) service.

Not all libraries use the recall feature, and not all libraries want every request to automatically be recalled. Some libraries want the recall to be in effect only if the person placing the request comes to the Circulation desk and asks for recall.

This is why the default variable is needed. 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, it should be set to 03.

setenv www_z37_date_to M002
Defines the end of the time slot when creating a hold request through the Web OPAC.

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, for example, M006. This will create a hold request that starts from the current date and ends in six months.

If the variable is not defined, or is defined incorrectly, the default "date to" will be three months.

setenv www_z36h_max_record_limit 100
Defines the maximum number of Z36H records displayed in the Web OPAC's User Information - Loan History list

Sort Order of Loans, Holds and Photocopy Requests
The following display parameters in www_server.conf allow you to configure the sort order of loans, holds and photocopy requests in the Web OPAC:

```
setenv www_user_z36_sort_routine      00
setenv www_user_z36_sort_order        A

setenv www_user_z37_sort_routine      00
setenv www_user_z37_sort_order        A


setenv www_user_z38_sort_routine      00
setenv www_user_z38_sort_order        A
```

The available routines are:

z36 (Loans):
00 = z36-due-date
01 = z36-sublibrary z36-status z30-collection z36-due-date

z37 (Holds):
00 = z37-priority z37-request-date z37-open-date z37-open-hour
01 = z30-sublibrary z30-item-status z30-collection z37-open-date z37-open-hour
02 = z30-sublibrary z30-item-status z30-collection z37-status z37-open-date z37-open-hour

z38 (Photocopy):
00 = z38-open-date

The available sort orders are:
A = Ascending Order
D = Descending Order

34. Return Receipt

The calculation of the value of the fine in the column by this name in the return receipt is dependent on the existence of a matching Z36H record. Libraries that do not

keep Z36H records should not use this column. If there are no Z36H records, the fine will appear as 0.00.

35. Column Headings (pc_tab_col.lng)

pc_tab_col.lng defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (USM01) table `pc_tab_col.lng`. For more information about `pc_tab_col.lng`, see the ALEPH User Guide - General chapter - Using and Customizing Your GUI.

The following is a list of the Circulation windows which use pc_tab_col.lng for formatting data, and their identifiers (Column 1 in pc_tab_col.lng):

Unique Code	**Circulation GUI Windows**
ADMIN_PROFILE_LIST	User Profile List
CHECK_CIRC	Item loan blocks
CHECK_HOLD_REQ	Hold request blocks
PC_CASH_CASH_EXPAND	Patron Information - Cash Expand
PC_CASH_SUB_TOTAL	Cash Subtotal by Sublibrary*
PC_CASH_SUMMARY	Patron Information - Cash transaction > tab *
PC_CIR_B_HOLD_LIST	Patron Information - Hold List
PC_CIR_B_ILL_LIST	Patron Information - ILL Requests*
PC_CIR_B_LOAN_LIST	Patron Information - List of Loans*
PC_CIR_B_PHOTO_LIST	Patron Information - List of Photocopy Requests*
PC_CIR_B_TITLE_LIST	Patron Information – Title Request List
PC_CIR_BOOKING_LIST	Item and patron booking list
PC_CIR_BOR_LIST	Global Patron Information/Local Patron Information
PC_CIR_CHANGE_ITEM	Change Item List
PC_CIR_D_HOLD_LIST	Item List - List of Hold Requests*
PC_CIR_D_LOAN_LIST	Item List - List of Loans
PC_CIR_D_PHOTO_LIST	Item List - List of Photocopy Requests
PC_CIR_DEL_HOLD_LIST	Item Delete Hold Request List
PC_CIR_HOLD_LIST	Hold Request List*
PC_CIR_ITEM_LIST	Document Item List

Unique Code	Circulation GUI Windows
PC_CIR_LOAN_EXPAND	Patron Information - Loan Expand
PC_CIR_LOAN_INFO	Loan/Return window - Patron Information in upper part
PC_CIR_LOAN_LINE	Loan window - Individual loan line in bottom part
PC_CIR_RENEWALL_LOG	Patron Information - Renew All Log*
PC_CIR_RESTORE_ITEM	Restore Item List
PC_CIR_RETURN_INFO	Return Window - Item Info in bottom part
PC_CIR_RR_LOAN_LIST	Reading Room Items
PC_CIR_RR_SUBLIBS	Reading Room Sublibrary List
PC_CIR_S_LOAN_LIST	Reserve Items
PC_CIR_S_LOAN2_LIST	Patron Information - Advance Booking
PC_COM_ADDR_LIST	Patron Information - Address List
PC_COM_BOR_IDS_LIST	Patron Information - Global Information - Update IDs - IDs Information
PC_COM_DOC	Brief List - Display BIB and ADM Record
PC_COM_HOLD_H_LIST	Item Circulation History - Hold Request List
PC_COM_HOLD_H_LIST_1	Item Circulation History - Hold Request List in ACQ and CAT if ITEM-BOR-INFO-CIRC-ONLY=Y
PC_COM_LOAN_H_LIST	Item History - List of Loans
PC_COM_LOAN_H_LIST_1	Item History - List of Loans in ACQ and CAT if ITEM-BOR-INFO-CIRC-ONLY=Y
PC_COM_SCAN	Scan for an Administrative Record
PC_COM_SCAN_I	Scan for an Administrative Record (Scan on Index)
PC_COM_SCAN_W	Scan for an Administrative Record (Scan on Words)
PC_COM_USER_LIST	Patron List
PC_USER_PROXY_LIST	Patron Information - Proxies/Sponsor
SHORT_LIST	Search for an Administrative Record - Brief List

* In this GUI table, an optional color/font can be used by the system for color/font differentiation between values in the same column. The alternative font and color can be defined in Column 8 and 9 of `pc_tab_col.lng`.

36. Overnight Loan Setup

In order to set up rolling loan periods overnight the special setup described below is required:

Reservation of the item for overnight loan is allowed, and the item is loaned according to a defined overnight time slot. During the day, the item is loaned for a given period (for example two hours), and cannot be booked.
The day loan is set in "tab16" as shown in the example below:

```
16A   35 ## ## + 00000000 00 + 0200
```

This example of "tab16" means that:
items from libraries of group ID "16A", in status "35" and in any process status - can be loaned to patron of any status – for two hours (0 days , 0 "grace days", + 2 hours).

There could be a problem when the day loan approaches the time of the overnight reservation, for example, when the overnight loan starts at 17:00 and a patron comes to loan the item at 16:00. The item should be returned at 17:00 and not at 18:00 as it will be using tab16.

The solution can be set as follows:

Set two tab16 tables called tab 16.1 and tab 16.2. The first one will include the definitions for the two-hour loan periods during the day:

```
16A   35 ## ## + 00000000 00 + 0200
```

For the item status "35", the item is loaned for 2 hours.

The other tab16 table will define for the same item status an actual due hour at 09:30:

```
16A   35 ## ## + 00000000 00 A 0930
```

In $alephe_proc there are two procedures, which switch between the tables.
In $alephe_tab/job_list, list the $aleph_proc/p_switch_tab16_1 and p_switch_tab16_2 procedures on the relevant days at the relevant times. This can be done using UTIL E/16/1 (Edit job_list):

```
W1 05:00 N tab16_1.log  csh -f /aleph/proc/p_switch_tab16_1
USM50
W1 15:00 N tab16_2.log  csh -f /aleph/proc/p_switch_tab16_2
USM50
```

The above settings will run the procedure that switches between the tab16 tables, at 05:00 and at 15:00, respectively.

Note that these procedures cannot be run from the UNIX utility "chron" but only from job_list.

37. Printing in Circulation

The circ.ini settings for printing in Circulation are described in the CIRC.INI section of the Printing System Librarian chapter.

38. Patron Picture

A picture of a patron is displayed in the Circulation GUI if a matching picture file is present in the relevant subdirectory on the server. The picture's file name is the Patron's ID. The following file formats can be used: *.jpg, *.jpeg, *.gif, *.bmp,*.jif, *.rle and *dib. If the file name contains letters, these must be saved in lowercase. When loading a patron picture through the GUI, the system converts the uppercase letters to lowercase automatically, so uppercase letters in the file name do not present a problem.

The picture is displayed in a square window. If the original picture does not matching this format, the picture is compressed to fit into the GUI picture window. It is best to use pictures that have a width/height ratio of 1:1 to get the optimum results.

The picture files reside in the [usr_library]/bor_pics subdirectory with the usr_library as defined in the ./alephe/aleph_start table in the line:

```
setenv    usr_library        USR00.
```

If the bor_pics directory has reached the limit of 1000 pictures, new files will be stored in numbered directories under bor_pics, such as bor_pics/1, etc.

The file name of the picture is the Patron's ID plus extension. For example, a picture file with the name 1234567.jpg will be displayed with the patron record of the patron with the ID 1234567. If there is more than one file present, the system displays the picture of the file it first encounters. If there is no matching picture file, the system displays the default_picture.* file, if present. If this is also not present, the system displays the .\alephcom\bmp\DefBor.gif file which is stored on the client after the installation of ALEPH.

A picture can be added or updated through the GUI "Update Picture" utility. Click on the Update Picture button to open a dialog box which lets you load a picture file from the local PC or from the network and see its preview. The default subdirectory that is opened is set in circ.ini. After confirming by clicking **OK** in the dialog box, the new picture file is saved on the server in the bor_pics directory of the user library (for example, ./usr00/bor_pics) under the patron ID plus the extension of the file from the PC. For example, if a file such as "mypicture.jpg" is loaded for patron 1234567, the new file will be "1234567.jpg" (under bor_pics). Note that at this point, all other picture files of that patron are deleted (for example, "1234567.gif").

In addition to the Global Patron Information tab, the picture is refreshed upon updating, in the loan and return tabs.

When deleting a patron, whether individually through the GUI, or through the Delete

Patrons (cir-23) service, all picture files with the patron's ID, regardless of their extension, are deleted. The default_picture.jpg file will not be deleted even if a patron with ID DEFAULT_PICTURE is deleted.

39. Expand Display

The information in expand display panes is generally taken from one specific Oracle record field. For example, the patron's name is always taken from Z303-NAME. In some instances the information is taken from different sources, some of which are Oracle record fields.

39.1. Display of Workstation Identity

There are three possibilities for displaying the workstation's identity in an expand window.

The station ID

The staff user's name

The IP address

Whether the station identity is displayed as a station ID / staff user's name or as an IP address, depends on the user's choice. If the user set up a station ID, the station ID or staff user's name is used, otherwise the workstation's IP address is used.

The station ID is set using the Set Worksation Identifier function, available by clicking the key icon at the bottom right of the screen. It is saved locally as an encrypted value in `.\alephcom\tab\stationid.dat`.

The user's staff name is taken from the Oracle record field Z67-USER-NAME.

The station's IP address is taken from the network setup.

40. Circulation Values of prof_library.conf

ue_06_active_hours
This variable determines the time period in which the requests daemon can actually be run. Note that this setup is by ADM library.

41. Duplicate Patron IDs

Duplicate IDs for the same patron are always allowed. There are certain conditions where it is also possible to use the same ID for different patrons.

The system checks for duplicate IDs as follows:

IDs that are NOT used as the key to retrieve Z303 in the GUI (N in column 5 of `tab_bor_id.lng`) can be duplicated for different patrons.

In the case where there is a duplicate ID date for two different patrons, the system checks that:

The IDs are of a different type (Z308-KEY-TYPE). It is not possible to enter the same ID for different patrons with the same ID type.

The IDs have different verification fields (Z308-VERIFICATION). It is not possible to have two identical pairs of ID and verification in the system.

Duplication is not possible between IDs that are not used as the key to retrieve Z303 in the GUI and those that are used as the key in the GUI (Y in column 5 of `tab_bor_id.lng`).

Duplication will only be allowed between ID types that are used by the patron to sign in to the Web OPAC.

There is a possible scenario where there are two identical IDs with two different verifications. If one of the patrons tries to change his password through the Web OPAC to the same one as the other patron, the system will alert the patron and will not let him save the new password until he enters a password that is different from the password of the other patron that has the same ID.

42. Remote Registration

The remote registration option can be used to create new patron records by copying a patron record from another ALEPH library outside your system in the same version, or from or to a "non-shared" ADM library in your own system. The expected scenario is that of a patron who is registered in the Education library at one university, walks in to apply for services at another (non-shared) library at the same institution, or at another institution. Provided that the libraries cooperate, and that the needed settings exist, the second library will be able to derive the patron's records from the institution where he is already registered, and the barcode on the patron's library card can be used at any of the libraries at which he is registered. The records that can be copied are the patron's global record; the patron's ID records; the patron's address records and the patron's local record. Note that the matching element for checking if the patron is already registered in your system is the patron's barcode. When a match between the patron's barcode in the remote system and the local system is found, the procedure is stopped. If no match is found, the system will copy the patron records into the local database.

42.1. Remote Registration Settings

There are a number of elements that need to be defined in order for this functionality to work:

All the libraries that work in cooperation and participate in this environment must be registered in the library.ini file located under the client's ALEPHCOM/TAB/. For example:

```
USM50 - USMARC Administrative  USM50 10.1.235.39:6515
USM51 - USMARC Administrative  USM51 10.1.235.39:6595
```

These libraries must also be registered in the rmt_reg.ini file located under CIRC/TAB. This file lists the libraries that will appear in the drop down menu in the Circulation client when opening the Remote Registration option from the Patrons main menu. For example:

```
!--------------------------------------------
!
!   List of remote registration libraries
!
!----->
USM50
USM51
TST50
```

There is no limit on the number of libraries that can be listed in this file.

The operator must have a valid password defined in the remote library in order to be able to derive the patron records from that library. This password must include the following permissions:

```
Patrons, Global/Display List  (USR/LIST)
Patrons, Global/Patron Record - Display (USR/SHOW)
```

The operator must also have the appropriate permissions in the local library so he can work with these records.

There are two possible ways for the system to operate regarding the operator's user name and password when using the Remote Registration option:

- The system automatically checks the operator's user name and password in the remote library. If the same user name/password is found in the remote library, the system will proceed and show the Remote Patron List. When there is no such user in the remote library, the system will display the ALEPH Temporary Password window for entering the password that is valid on the other system.

- When choosing the Remote Registration option from the Patrons main menu, the system displays the ALEPH

Temporary Password window for entering the password to work with. No check is performed prior to that.

This is determined in the circ.ini file in the following section:

```
[RemoteRegistration]
UseGlobalPassword=N
```

When set to N, the system does not perform a preliminary check and immediately shows the ALEPH Temporary Password window when the Remote Registration option is chosen.

When set to Y, the system first tries to check the current operator's password in the remote library. The ALEPH Temporary Password window will be shown only if there is no such user name and password in the remote library.

When copying patron records from a remote library, you can perform checks on the patron that is targeted for import records.

The checks should be listed in the RMT-REG section of `tab_check_circ` (located under . `/xxx50/tab`). The definition should appear in each of the ADM libraries from which the records are copied.

The relevant checks in this section are the ones that check the global aspects of the patron records. For example, the global patron record blocks, the blocks on the patron's ALEPH record, etc. Following is an example from the table:

```
RMT-REG    check_circ_1_a
RMT-REG    check_circ_1_b
RMT-REG    check_circ_1_c_a
RMT-REG    check_circ_1_c_b
RMT-REG    check_circ_10_a
```

The checks are performed when the patron is selected. If error messages are received, the operator can nevertheless override them and copy the patron records to the library.

When copying patron records from a remote library, the Home Library field in the global record will be populated with the active ADM library of the local system. The original home library from the remote system will be saved in the Note 3 field of the global record for follow up.

The text for the note referring to the patron's original home library is set in the `window.dat` file located under `CIRC/TAB/<lng>`. The following entry in the `window.dat` file is consulted:

```
RemoteRegistration.Field3                REMOTE REGISTRATION:
HOME-LIBRARY=%s RMT-LIBRARY=%s
```

Patron Addresses

If the operator authorization in the remote library includes Patrons, Global/Addresses

Display (USR/ADDRESS-GET), then the addresses are copied from the remote library to the current library.

Patron Local "ALEPH" Record

If the operator authorization in the remote library includes Circulation/Patron, Local Display (CIRCULATION/BOR-SHOW) and if the patron status in the remote library's ALEPH record is a valid patron status in the current library, an "ALEPH" local library record, with this status, is created in the current library. The settings in the local "ALEPH" record are dependent on the definitions for the borrower status in the current library's `tab31`.

43. Different Hold Request Forms (tab_hold_request_form)

You can define the system to use different hold request forms when the patron places a hold request through the Web OPAC. Two definitions are required:

Create the desired settings in the `tab_hold_request_form` table with a new suffix.

Copy the hold request forms to files with the same names but with the suffix from `tab_hold_request_form`.

43.1. tab_hold_request_form

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 variable in `tab100`.
If no match is found, the system will display the hold request forms without the suffix. The same will happen if the table does not exist. Note that as in most of ALEPH table, specific lines should come before the general ones. For example:

```
32 ##### DP ## # depo32
## ##### DP ## # depo
```

Following is an example of the table:

```
! 1    2  3  4  5    6
!!!!!-!!-!!-!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ULINC 42 ## ## N nonav
##### 32 DP ## # depo32
##### ## DP ## # depo
ULINC ## ## 03 # 03ulinc
```

```
##### ## ## 03 3 03
```

Following are details about each column:

Column 1: Sublibrary, the item's sublibrary.

Column 2: Item Status

Column 3: Item Process Status

Column 4: Patron Status

Column 5: Item Availability Status. The optional values in this column are Y or N. The item availability is set according to the value in the AVAILABILITY-ROUTINE variable of `tab100`.

Column 6: WWW Form Suffix. This is the suffix to be used to identify the special set of hold request forms in the `/alephe/www_f_lng` directory.

43.2. Creating a Different Set of Hold Request Forms

After a suffix is created in `tab_hold_request_form` for a specific combination of the available parameters, copy the following pages to files with the same name but with the created suffix:

item-hold-request-body
item-hold-request-head
item-hold-request-tail
booking-page-delivery-form-head
booking-page-pickup-form-head
booking-page-return-pickup-head
booking-page-return-delivery-head
booking-page-form-body
booking-page-return-delivery-tail
booking-page-delivery-form-tail
booking-page-pickup-form-tail
booking-page-return-pickup-tail

For example:
item-hold-request-body-depo
item-hold-request-head-depo
item-hold-request-tail-depo
booking-page-delivery-form-head-depo
booking-page-pickup-form-head-depo
booking-page-return-pickup-head-depo
booking-page-return-delivery-head-depo
booking-page-form-body-depo
booking-page-return-delivery-tail-depo
booking-page-delivery-form-tail-depo

booking-page-pickup-form-tail-depo
booking-page-return-pickup-tail-depo

This example was created to define a different set of hold request forms for remote storage items.

Note that in order for the hold request page to be fully displayed, all of these pages need to have this suffix. Change the pages as desired to create different sets of hold request forms. Note that you can change the text that appear on the pages, the location of the various place holders or remove a place holder. The place holders are the same through the different type of pages and it is not possible to add place holders.

44. Different Photocopy Request Forms (tab_photo_request_form)

You can define the system to use different photocopy request forms when the patron places a photocopy request through the Web OPAC. Two definitions are required:

1. Create the desired settings in the `tab_photo_request_form` table with a new suffix.
2. Copy the hold request forms to files with the same names but with the suffix from `tab_photo_request_form`.

44.1. 43.1 tab_photo_request_form

This table sets the option to configure the system to use different photocopy request forms when the patron places a request through the Web OPAC. When a request is placed, the system looks for a matching line in the table. The correct line is selected according to one of the following 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 variable in `tab100`.
If no match is found or the table does not exist, the system displays the photocopy request forms without the suffix. Note that as in most of the ALEPH table, specific lines should come before the general ones. For example:

```
32 ##### DP ## # depo32
## ##### DP ## # depo
```

The following is an example of the table:

```
! 1    2  3  4  5    6
!!!!!-!!-!!-!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ULINC 42 ## ## N nonav
##### 32 DP ## # depo32
##### ## DP ## # depo
ULINC ## ## 03 # 03ulinc
##### ## ## 03 3 03
```

The following are details about each column:

Column 1: Sublibrary, the item's sublibrary.

Column 2: Item Status

Column 3: Item Process Status

Column 4: Patron Status

Column 5: Item Availability Status. The optional values in this column are Y or N. The item availability is set according to the value in the AVAILABILITY-ROUTINE variable of `tab100`.

Column 6: WWW Form Suffix. This is the suffix to be used to identify the special set of hold request forms in the `/alephe/www_f_lng` directory.

44.2. Creating a Different Set of photocopy Request Forms

After a suffix is created in `tab_photo_request_form` for a specific combination of the available parameters, copy the following pages to files with the same name but with the new suffix:

- item-photo-request-body
- item-phot-request-head
- item-photo-request-tail
- item-photo-request-tail-an (if used)

For example:

- item-photo-request-body-depo
- item-phot-request-head-depo
- item-photo-request-tail-depo
- item-photo-request-tail-an-depo

This example was created to define a different set of photocopy request forms for remote storage items.

Note that in order for the photocopy request page to be fully displayed, all these pages need to have this suffix. Change the pages as desired to create different sets of photocopy request forms. Note that you can change the text that appear on the pages, the location of the various place holders or remove a place holder. The place holders are the same through the different type of pages and it is not possible to add place holders.

45. Patron Retrieval by Concatenation of ID and Verification

You can retrieve a patron by using the ID (that is, Z308-KEY-DATA) and the password/verification (Z308-VERIFICATION).

Z308-KEY-DATA comprises of a patron name (for example, "NANCY"), a

concatenation character (for example, a hyphen "-") and the verification (for example, "X"):

```
Z308-KEY-DATA : NANCY-X
Z308-VERIFICATION : X
```

This lets a patron log in with a user name and its verification. For example:

User Name: NANCY
Verification: X

The program looks for Z308-KEY-DATA "NANCY" first. If it is not found, it looks for "NANCY-X".

In order to enable this, the following configurations must be set:

The concatenation character must be defined in `./alephe/aleph_start`:

```
setenv bor_id_concat_char     -
```

Column 8 in `tab_bor_id.lng` must be set to "N" for the codes used for this purpose, to prevent patrons from changing their verification.

46. Defining the Zip Code Format for a Patron's Address in Printouts

ALEPH supports both the USA and Canadian ZIP code formats. Which format is used, is defined in the XML translation file `global.trn` in the `form_eng` directory.

Enter the line:

```
z302-zip                              ZIP-CAN    S
```

to use the Canadian format (123 456).

Enter the line:

```
z304-zip                              ZIP-USA    S
```

to use the USA format (12345-6789).

For any other format, do not enter any line in `global.trn`. The ZIP code will be displayed exactly as it was entered.

In addition, the ZIP code can be appended to the last address field, instead of being shown in a separate field. This can be done for all three zip formats (that is, USA, Canadian and unformatted) by configuring `global.trn` (or any form-specific `trn` file) in the following way :

```
z302-address-0                  AP-ZIP-USA S z302-zip z302-address-1
z302-address-1                  AP-ZIP-USA S z302-zip z302-address-2
z302-address-2                  AP-ZIP-USA S z302-zip z302-address-3
z302-address-3                  AP-ZIP-USA S z302-zip z302-address-4
z302-address-4                  AP-ZIP-USA S z302-zip
z302-zip                        TO-BLANK   S
```

Using this setup will append the ZIP code in USA format to the `z302-address-0` field if it is not empty and the z302-address-1 field is empty or to the z302-address-1 if it is not empty and the z302-address-2 field is empty and so on.

The following line deletes the Z302-zip field so that it will not show up in the form twice:

```
z302-zip                              TO-BLANK   S
```

You can append the ZIP code to the last address field while using Canadian format by using AP-ZIP-CAN instead of AP-ZIP-USA. You can also append the ZIP code to the last address field without formatting by using AP-ZIP instead of AP-ZIP-USA.

47. Log File Expansion

The `pc_server` log file can be expanded to include information on the status of the checks that were performed on a hold request. This means that the exact number of errors caused by each check configured in `tab_hold_request` can be logged. This is done by setting the environment variable DEBUG_CHECK_HOLD to Y prior to `pc_server` initialization in the following manner:

```
setenv DEBUG_CHECK_HOLD Y
```

If this configuration is set, the log file will include information such as :

```
check_hold_request "CIRC      " :

After check_hold_request_a BUF-ERROR-NO-ERRORS=01

After check_hold_request_b BUF-ERROR-NO-ERRORS=02

After check_hold_request_c BUF-ERROR-NO-ERRORS=02
```

48. Booking Requests

In order to set booking request policy, you must consider the following issues:

- Booking staff and patron permissions
- Item booking availability
- Like item policy
- Booking checks
- Booking head, tail and delivery time
- Booking release time
- Booking delete interval
- How far into the future requests can be made
- Delivery policies
- Cash policies

The setup for each of these issues is described below.

48.1. Patron Booking Permissions

The following booking-related default patron permissions are set in the ADM library's `tab/tab31` table:

- Item Booking Permission - This is set in column 20 of the table. If this column is set to 'Y' then booking is allowed for all available items, meaning that the item has not been defined as N in column 16 of `tab15.lng`. If it is set to 'N'

then booking is permitted only for items that have been defined in `tab15.lng` (column 16) as available for all (Y).

- Ignore Closing Hours – This is set in column 21 of the table. Setting this column to 'Y' has two consequences:
  - The release time will not be taken into account for requests that were placed by a patron with this patron status. The requested item will therefore be kept until the patron comes to pick it up, no matter how late he is.
  - Item booking limitations concerning library closing or opening hours are disregarded when the patron places a booking request.

Note

Patron permissions are checked only if check_booking_a has been defined in the proper BK- section of the tab_hold_request table for regular bookings or in the AB-POST section for advance booking requests.

48.2. Staff Booking Permissions

The following staff permissions are defined in `alephe/tab/user_function.eng`

CIRCULATION	BOOKING-VIEW	Booking request – View
CIRCULATION	BOOKING-UPDATE	Booking request - Update
CIRCULATION	BOOKING-CREATE	Booking request - Create
CIRCULATION	BOOKING-OVERIDE	Booking request - Override Blocks
CIRCULATION	BOOKING-DELETE	Booking request - Delete
CIRCULATION	BOOKING-DELIVER	Booking Request – Deliver

Note 1

The staff permissions that are required for advance booking requests are

CIRCULATION	SCHEDULE-LIST	Advance Booking - List of slots
CIRCULATION	SCHEDULE-REQUEST	Advance Booking - Create

Note 2

The staff user must also be registered as a patron in order for the **Availability** button to be usable by the staff user when creating a booking request from the GUI.

48.3. Item Booking Ability

Definitions of the item booking capabilities are defined in the ADM library's `tab/tab15.lng` table. You must define the following in this table:

Reloaning limit – This is set in column 15 of the table. The value denotes the number of hours during which the same copy of an item cannot be re-loaned or re-booked to the same patron starting from the time he returns it or the end time of an existing booking request. The values "00" or blanks are used for unrestricted re-loaning and re-booking.

Item booking capability – This is set in columns 16 and column 17 of the table.

In column 16 three different options can be used to define the different booking restrictions of an item status:

- Y - Booking is possible for all patrons, regardless of their booking permissions.
- N - Booking is not possible
- C - Booking permission is conditional. Only patrons that have been granted booking permissions in their local privileges record can book the item.

In column 17 the item's availability with regard to library open hours is set. Three options are configurable:

- A - Booking is possible at all times (both open and closed hours)
- O - Booking is possible only during open hours
- C - Booking is possible only during closed hours

While the reloaning limit is hardcoded and is always checked for both types of bookings, the item booking capability is checked only if it has been defined for checking in the BK- section of tab_hold_request for regular bookings and in the AB-POST section for advance booking requests.

48.4. Maximum Number of Bookings and Maximum Booking Period

Definitions of the patron's restrictions with regard to booking are defined in the ADM library's `tab/tab16` table. Two types of restrictions are defined in this table:

Maximum number of active bookings – This is set in column 26 of the table. The value can be set as maximum per sublibrary and item status, for all item statuses in the sublibrary (by setting the item status to 99), for all item statuses in the same group of sublibraries (by setting the 98 line), or for the entire ADM (by setting the ADM code instead of the group code in column 1).

- **Maximum Booking Period** – A booking request can not be set by the patron to be longer than the maximum period that the item would have been loaned. Columns 5,6,8,9 for regular items and columns 15,16,17,18 for requested items set this maximum period, as it is done for loans.

Note that these checks will only be performed if they have been defined for checking in the BK- section of tab_hold_request for regular bookings and in the AB-POST section for advance booking requests.

48.5. Like Item Policy

Defining what items are considered *like items* is important for calculating item availability. Since an item is considered available if any item that is its *like item* is available, setting this policy will influence the availability of a requested item. Other than having similar material type and enumerations, the items must be *like* in one of the following item parameters in the ADM library's tab/tab100 setup:

SUB-LIBRARY-DIVISION – Setting this variable to 'Y' means that the two items must belong to the same sublibrary to be considered *like*.

HOLD-REQUEST-ITM-STATUS – Setting this variable to 'Y' means that the two items must have the same item status to be considered *like*.

HOLD-REQ-PROCESS-STATUS– Setting this variable to 'Y' means that the two items must have the same item process status to be considered *like*.

HOLD-REQUEST-COLLECTION– Setting this variable to 'Y' means that the two items must belong to the same collection to be considered *like*.

48.6. Booking Request Checks

The ADM library's `tab/tab_hold_request` table sets the checks to be performed by the system when a booking request is placed. Column 1 defines when the check will be performed:

- BK-PRE is used for the 'Booking' link when the list of items is displayed in the Web OPAC
- BK-POST is used when submitting the booking request form in the Web OPAC
- BK-CIRC is used when a request is submitted in the Circulation client
- AB-PRE is used when the list of items is displayed in the Web OPAC. Passing this check is required to show the 'request' link next to advance booking items
- AB-POST is used when submitting an advance booking request from the OPAC or from the GUI.
- REALLOC-AB is used when any booking request is reallocated to another like item.

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.

When setting which checks to use, take into account the system resources required for checks that are made when the items list is displayed (PRE checks). If the checks require comparing item records and existing requests, and if there are many item records, this can create an undesirable load on the server. If a check is listed for PRE, it need not be listed for POST, since the POST scenario will not happen. Checks that relate to the patron usually appear in POST.

CHECK	**Action**	**$aleph_root/error_lng/ check_hold_request**	**Suggested use**
booking_a	If tab15 col. 16 = N item can not be booked	1210	BK-PRE, BK-CIRC

CHECK	Action	$aleph_root/error_lng/check_hold_request	Suggested use
booking_b	Checks if tab37_booking_pickup and tab37_booking_delivery return empty list.	1106	BK-PRE, BK-CIRC
booking_c	Item cannot be booked if the same tem is already booked by the same patron at the same day and Z305-MULTI-HOLD=N.	1222	AB-POST
booking_e	Booking span is longer than the defined loan period for this item. This check is hard coded in requests that are submitted from the Web.	1251	BK-CIRC
booking_h	tab16 col.26, maximum number of booking requests per Sub Library and Item Status	1330	BK-POST, BK-CIRC
booking_h_1	tab16 col.26, maximum number of booking requests per Sublibrary & Item Status. This check is similar to check_booking_request_h, but also includes in the check existing loans that were originally booking requests.	1330	BK-POST, BK-CIRC
booking_k	Checks if the booking period complies with `tab15` policy. For example, if the booking request is during the night, `tab15` must allow it	1342	BK-POST, BK-CIRC
booking_l	`tab_booking`, Col8: `max_booking_start_date`. Limits how far into the future bookings can be made.	1310	BK-POST, BK-CIRC
booking_q	`tab16` - Limit of Bookings per Sublibrary	1311	BK-POST, BK-CIRC
booking_q_1	tab16 – Limit of Bookings per Sublibrary This check is similar to check_booking_request_q, but also includes in the check existing loans that were originally booking requests	1311	BK-POST, BK-CIRC

CHECK	Action	$aleph_root/error_lng/ check_hold_request	Suggested use
booking_r	patron expiration date has been reached	1320	BK-POST, BK-CIRC
booking_t	tab16 col.13 check for maximum number of hold requests for the ADM library (ADM line(s) in tab16)	1312	BK-POST, BK-CIRC
booking_t_1	tab16 col.26 check for maximum number of booking requests for the ADM library (ADM line(s) in tab16) This check is similar to check_ booking _request_t, but also includes in the check existing loans that were originally booking requests.	1312	BK-POST, BK-CIRC
booking_x	tab16 - Limit of Bookings per Sublibrary Group	1313	BK-POST, BK-CIRC
booking_x_1	tab16 - Limit of Bookings per Sublibrary Group This check is similar to check_ booking _request_x, but also includes in the check existing loans that were originally booking requests.	1313	BK-POST, BK-CIRC
hold_l	patron-id has not been entered	1140	BK-PRE

The text of messages associated with the checks is defined in `$aleph_root/error_lng/ check_hold_request`.

48.7. Booking Request Default Times

The default times that are related to a booking request are set in the ADM library's `tab/tab_booking` table. These are set per sublibrary, item status and material type. The following times are set:

- **Head Time and Tail Time** – these are the times that will be added at the beginning and end of a request for request preparation purposes. The format of these columns is [M/H/D]*nnn*, where 'M' stands for minutes, 'H' stands for hours and 'D' stands for Days. *nnn* is a three digit number that sets the amount of time that will be used. Note that these are default times and that they can be overridden by a staff user.

- **Release Time** – This is the length of time that passes from a request becoming active that will release the item to be loaned to patrons other than the booked one. The format of this column is [M/H/D]*nnn*, where 'M' stands for minutes, 'H' stands for hours and 'D' stands for Days. *nnn* is a three digit number that sets the amount of time that will be used. Note that these are default times and that they can be overridden by a staff user.
- **Delete Interval** – This is the length of time before a request becomes active, up to which the patron will be able to delete his request. After this time deleting a request by the patron will not be possible. The format of this column is [M/H/D]*nnn*, where 'M' stands for minutes, 'H' stands for hours and 'D' stands for Days. *nnn* is a three digit number that sets the amount of time that will be used. Note that these are default times and that they can be overridden by a staff user.
- **Time in Advance** – This is the latest time in the future for which a patron can place a request. Placing a request beyond this time is not possible. The format of this column is [D/W/M/Y]*nnn*, where 'D' stands for days, 'W' stands for weeks, 'M' stands for months and 'Y' stands for years. *nnn* is a three-digit number that sets the time that will be used.
- **Preview Period** - This is the amount of time in advance that a booking can be executed. For example, if the booking period is between 12:00 and 18:00, and this time is defined as 5 minutes, the patron can execute the booking starting from 11:55. The column format is [M/H/D]*nnn*, defining minutes, hours or days.

Note

These times are used also for advance booking requests.

48.8. Delivery and Pickup Location Policies

Booking functionality includes an option to request that the material be transferred to another location. Two types of transfer are provided for:

Pickup – that is, the service desk of library in the system (including the location of the item)

Delivery – that is, transfer to a non-library location, such as a classroom

Delivery/Pickup location policies are set in tab37_booking_delivery and tab37_booking_pickup tables.

48.8.1. tab37_booking_pickup

The ADM library's `tab/tab37_booking_pickup` table sets which pickup locations can be requested, based on the item's sublibrary, item status, item process status, item material type and patron status. Pickup locations must also be defined as sublibraries in the `./alephe/tab/tab_sub_library.lng` table. Up to ten pickup locations can be set in each row. If more than ten pickup locations are required, the list can be continued in the next row, leaving columns 1-5 empty.

For example:

```
WID   01 IP 03 WID   LAW   LIB1  LIB2  LIB3  LIB4  LIB5  LIB6  LIB7  LIB8
                       LIB9  LIB10
```

48.8.2. tab37_booking_delivery

The ADM library's `tab/tab37_booking_delivery` table sets which delivery locations can be requested for a sublibrary, item status, item process status, item material type and patron status. The locations must also be defined in the ADM library's `tab_delivery_locations.lng` table

Note

You can define that all of the delivery locations in `tab_delivery_locations.lng` are valid delivery locations for a specific sublibrary, item status, item process status, item material type and patron status. This is done by setting column 6 of `tab37_booking_delivery` to '*'.

Up to ten delivery locations can be set in each row. If more than ten delivery locations are required, the list can be continued in the next row as is described for `tab37_booking_pickup` above.

48.8.3. tab_delivery_locations.lng

The ADM library's `tab_delivery_locations.lng` table lists the codes and addresses of the delivery locations that are allowed by the ADM library. The delivery location codes that are used in `tab37_booking_delivery` must be defined in this table.

48.8.4. tab27

Delivery handling settings are defined in the ADM library's `tab/tab27` table. This table sets how deliveries are handled per sublibrary, item status and pickup location. Settings 06, 07, and 08 in column 5 define actions that are applied to booking requests.

06 in column 5 defines that when delivery is performed, the item is immediately loaned to the requesting patron. If option 08 is used, then a reading room record will also be created. Setting 07 in column 5 means that upon delivery, the item will be placed in transit to the pickup location and will be loaned to the patron only upon arrival to the pickup location. Note that these are the only options for a booking request. Also note that this table is used also for hold requests, and column 1 differentiates between the two request types ('H' means hold requests and 'B' means booking requests).

48.9. Cash Transactions

The following cash transactions are defined in the ADM library's `tab/tab18.lng` table for booking requests:

0091 - Booking Request Created

0092 - Booking Request Filled (loaned)
0093 - Booking Request Delivered
0094 - Booking request not filled (not picked up)

These can set to be either active or non-active, like all other defined transactions.

48.10. Policy of a Booking Request Fulfillment

Booked items can be loaned out by using the Booking nodes **Loan** button. In this case, the loan is always considered a fulfillment of the booking request. If the loan is made from the loan tab and it is made within the preview period of the booking request as set in `tab_booking`, it will also be considered as a booking request fulfillment. However, if the loan is made from the loan tab and the loaning patron has a booking request which is not in its preview period, the tab100 variable OVERLAP-BOOKING, will set the system behavior. You can set up the following options:

- 01 = Extends loan until end of Booking Period. Considers it a booking request fulfillment
- 02 = Deletes booking request and executes loan regularly using the calculated due date
- 03 = GUI functionality: Allows staff to choose between option 01 and option 02. An offline loan, such as Self-check, X-Server or Web and batch renewals: the same as option 01.
- 04 = GUI functionality: the same as option 03 (for GUI). An offline loan, such as Self-check, X-Server or Web and batch renewals: the same as option 02.

Note:
Tab100 OVERLAP-BOOKING is consulted only if the booking end time is later than the calculated loan due date. If the booking end time is earlier then the loan calculated due date (based on tab16) – than tab100 OVERLAP-BOOKING is ignored. In this case, the due date will be the booking end time.

49. Displaying Item Information

The format of the Item Information and the Circulation Summary that are displayed in the Items tab of the Circulation module is defined in HTML-like files. These files, called item-expand and item-current-summary, are found under the pc_display_lng directory that is found under the ADM library's root directory. They can be easily configured to change and format the data that is displayed. These files are also used for item information display in the Cataloging module Items tab and in the Acquisitions module Orders and Serials tab.

When the tab100 variable ITEM-BOR-INFO-CIRC-ONLY is set to 'Y', separate files are required for the Circulation module and the Acquisitions and Cataloging module, so that patron information will be displayed only in the Circulation. For this purpose two separate files exist. The file item-expand-no-bor is used to display the item-expand information without patron information, and item-current-summary-no-bor is used to display the item-current-summary information without patron information.

50. Patron Direct Queue

Patron Direct Queue (PDQ) is the name of ALEPH's direct consortial borrowing function. This chapter describes the configuration required to optimize its capabilities.

The following are prerequisites for using PDQ:

- All the institutions must share a single ALEPH catalog (single BIB), which is either Union View, or which has shared bibliographic records.
- There must be unique item barcodes across all participating libraries.
- The participating institutions must have defined their patrons as shared.
- Patron IDs must be unique across the participating institutions.
- There must be cross-institution agreements on the meaning of the patron statuses of the shared privileges records.

50.1. Patron Consortium Privileges

50.1.1. Creating the Consortium Privileges

A patron is given a single general ALEPH privileges record, which is used by all of the sublibraries in which the patron does not have specific privileges.

Use of the ALEPH privileges depends on a sublibrary's definitions in columns 9-13 of alephe_tab/tab_sub_library. If the sublibrary defines ALEPH in one of these columns, then the ALEPH local privileges record is used according to the hierarchy defined in this table. A sublibrary can define in this table that if no sublibrary privileges record is found, the ALEPH privileges record is used. Another sublibrary can require explicit registration in the sublibrary by omitting ALEPH from columns 9-13 of tab_sub_library. Such a sublibrary will give circulation privileges only to patrons that have explicitly registered in the sublibrary.

Note that ADM-level privileges (the ADM local privileges record) are granted automatically when a patron registers in a sublibrary. The exact privileges that will be granted in this record depend on the matching line that is defined in tab31.

A patron is granted consortium-wide privileges when registering in any of the consortium's sublibraries by being granted privileges in the ALEPH local privileges record. Receiving the ALEPH privileges depends on the following:

- Column 22 of tab31 – You can use this column to determine, for every patron status, whether or not registration in the sublibrary also grants ALEPH privileges, and thus consortium-level privileges. For example, an line of:

  ```
  WID   10 Y Y Y N Y Y Y N F + M 00000006 9999999999 N Y Y 21 Y N Y
  ```

  Means that whenever a patron registers in sublibrary WID with a status of 10, an ALEPH local privileges record is created.

- tab_map_privileges – This table defines which ALEPH patron statuses are given to every patron that registers in a sublibrary. For example, this table can define that a patron that registers in the Medicine Library is granted ALEPH

privileges with status 01 and a patron that registers in the Humanities Library is granted ALEPH privileges with status 02.

For example, a line of:

```
WID  10 04
```

Means that whenever a patron with status 10 registers in sublibrary WID, the ALEPH record that will be created will have the status 04.

ALEPH privileges should use an agreed range of patron statuses. The same range should be used by all of the participating institutions, and the meaning of each status should also be agreed on by all of the participating institutions. These statuses should be added to the circulation policy tables of all institutions

For example, all institutions agree to use statuses 01-04 for ALEPH patron privileges. They all agree that 01 is 'First Year Student'. Each library can then define the specific privileges for a first year student who is not registered within the institution (and is therefore using ALEPH 01 privileges) according to its own policies. See *50.2 Setting the Library's Policies* below.

- The privileges that a patron will be granted in the ALEPH record depend on the tab31 definitions that are defined for the new patron status (as defined in tab_map_privileges) and for ALEPH. The tab31 table that is used is the one that belongs to the ADM library in which the registration was made. For example, the line:

  ```
  ALEPH   04 N N N N Y Y Y N F + M 00000006 9999999999 N Y Y 21 Y N Y
  ```

  Defines the privileges that an ALEPH record with status 04 will be granted.

Note that the Automatic Creation of ALEPH Z305 Records (cir-83) batch job may be used to automatically open up ALEPH Z305 records after the above described setup has been done.

50.2. Setting the Library's Policies for PDQ

50.2.1. Setting Library Global Policies

The patron privileges record that is used when the patron circulates sublibrary items is defined in the alephe/tab tab_sub_library.lng table. Columns 9-13 of this table define a hierarchy of sublibraries. For example, the following line:

```
MED   1 USM50 L Medicine Library              15A  16A   17A   MED   ALEPH
```

Means that when a patron accesses an item that belongs to the Medicine Library, he must have MED privileges or ALEPH privileges. A line such as

```
MED   1 USM50 L Medicine Library              15A  16A   17A   MED   USM50
```

Means that the patron must have MED privileges or USM50 privileges. In this example, the sublibrary does not use ALEPH privileges, so that patrons of other ADM libraries may not have access to MED items. This can be used by sublibraries that do not want to allow patrons of other libraries to access their collection.

50.2.2. Setting Patron-specific Policies

A sublibrary may want to share its resources with other libraries, but restrict the resource sharing to specific item collections or to specific patron statuses, or apply special due date and fine policies to external patrons. This is managed by configuring tab16 to force the special policies, as described below.

As mentioned above, the ALEPH cross-institution patron privileges use an agreed range of privileges. For example, if an agreement has been made that the statuses 01-04 are used for ALEPH privileges, each institution and sublibrary is responsible for avoiding the use of these statuses for their own patrons. To enable this, you can define a special line for non-local patrons. For example, a tab16 line of:

```
16A   ## ## 04 + 00000024 03 A 2400 0000 00050 000 000 D + 00000014 A
2400 007 014 05
```

will block all external patrons with status 04 from loaning and requesting items. This is because this line refers to patron status 04, which is always an ALEPH status, and it sets columns 12 and 13 so that the maximum number of loans and requests is 0. If loans are allowed, special due dates can be set in this line, forcing separate due date policies for local patrons and for external patrons.

50.3. Creating a Title Request

50.3.1. The Pool of Items

When a title request is created, any item that belongs to the requested title may potentially fulfill the request. The pool of items that may potentially fulfill the request includes items that match the following conditions:

- The items belong to the requested title. This can be items that belong to one ADM library, items that belong to multi-ADM libraries that share the same bibliographic record, or items that belong to multiple bibliographic records that are united in a Union View.
- The items share the same enumeration and chronology information, and have the same material type.
- The items are available according to the 'TITLE-REQ' section of tab_hold_request. Note that if the items belong to more than one ADM library, each item will be checked according to the tab_hold_request table of its owning ADM library.
- The item may be sent to the requested pickup location. This is determined in the tab37 table of the owning ADM library. Correct configuration of the pickup location table is explained in 50.3.2 The Pickup List section below.

50.3.2. The Pickup List

Setting the Allowed Pickup Locations

The title request form displays a list of the pickup locations that the patron may choose from. The pickup location list is a union of the pickup locations of all of the items in the pool of potential items, as explained above.

Any ADM library can locally control two variables that affect the pickup location list:

- Where its own items may be sent to. This can be sensitive to the type of item and the status of the requesting patron.
- Which sublibraries in the ADM library can serve as pickup locations for items that belong to other ADM libraries.

Setting the pickup location list is done in the ADM library's tab37 table. A sublibrary can define that all of its items can only be picked up within the sublibrary. For example, if LB50A and LB50B are sublibraries in USM50:

```
LB50A ## ## ## # LB50A
LB50B ## ## ## # LB50B
```

These lines mean that LB50A and LB50B allow their items for pickup only within the sublibrary. If the requested title has items in LB50A and in LB50B, both will be displayed as pickup locations. If the patron selects LB50A, only the LB50A items can fulfill the request.

A sublibrary may also decide that certain patron statuses may request special items for pickup in another ADM. For example:

```
LB50A 01 ## 02 # LB50A LB51A
LB50A ## ## ## # LB50A
LB50B ## ## ## # LB50B
```

This means that patrons with status 02 may request LB50A items with status 01 for pickup in a sublibrary that is in another ADM; in this case, in LB51A which is a sublibrary in USM51. If the title has items in both LB50A and LB50B, the LB50A item has a status of 01, and the patron has a status of 02, the pickup location list will be LB50A, LB51A, LB50B.

The sublibrary may leave the work of defining the pickup locations that are available in another ADM library to the pickup ADM library. In other words, instead of USM50 having to define LB51A, which is a USM51 sublibrary, as an available pickup location for its items, it may simply define that USM51 is an available pickup library, USM51 will define, in its own tab37, which of its sublibraries may serve as pickup locations for items that have been transferred from USM50.

For example:

```
LB50A ## ## ## # LB50A USM51
LB50B ## ## ## # LB50B
```

This setup means that the LB50A may be picked up in USM51. USM51 may define which pickup locations are allowed for USM50 items by setting its tab37 table as follows:

```
USM50                LB51A
```

Note that columns 2-5 are spaces and that column 1 is the code of the owning ADM. This line means that items that belong to USM50 and are allowed for pickup in USM51 may use LB51A as a pickup location. The allowed pickup locations will therefore be LB50A, LB51A, LB50B, as in the previous example.

USM51 may also define which pickup locations are allowed for USM50 items by setting its tab37 table as follows:

```
USM51                LB51A
```

This setup means that all items that do not belong to USM51 and that are allowed by their owning library for pickup in USM51 may use LB51A as a pickup location.

Another parameter that affects the pickup location list is the tab100 variable PICKUP-LIST-METHOD. If this variable is set to 2, the only allowed pickup location will be the sublibrary that is defined as the patron's dispatch library. This pickup location must still be allowed by the tab37 definitions, as described above.

Paging Rules

Paging rules are another consideration in creating the pickup list. This means that a library may be defined not to be used as a pickup location if an item is available at the sublibrary. This policy definition is achieved by omitting the sublibrary from the tab37 line that has Y in column 5. For example:

```
LB50B ## ## ## N LB50C LB50D LB50B
LB50B ## ## ## Y LB50C LB50D
```

This setup means that sublibrary LB50B cannot be a pickup location if there is an available item in LB50B. If an available item exists in sublibrary LB50B, the pickup location LB50B will be removed from the list of possible pickup locations.

Dispatch Library as Single Pickup Location

The pickup location list may be narrowed down to allow only a single pickup location for every patron. To do this, set the PICKUP-LIST-METHOD tab100 variable to 2. In this case, only the patron's dispatch library, as defined in the patron's global record, will be an allowed pickup location. In order to activate this functionality, the HOME-LIBRARY field in the GLOBAL PATRON INFORMATION (z303) record should contain the ADM library in which the patron is registered.

50.3.3. The Roster

The setup of the roster defines the order in which the potential suppliers are added to the queue.

The roster is defined in terms of groups. In other words, a few groups of potential suppliers can be defined for a specific pickup location. If a title request specifies a given pickup location as the desired pickup location, all of the potential suppliers in the groups are added to the roster. As each group is assigned a priority in column 2, all of the suppliers in the first group are added before all of the suppliers of the second group. Ordering within a group can either be random or subject to pre-defined ordering, depending on the setup of column 3. The roster is set in the tab_roster table.

For example:

```
SUB1 01 Y SUB2 SUB3
SUB1 02 N SUB4 SUB5
```

This setup means that for a title request that specifies SUB1 as a pickup location, the potential suppliers SUB2 and SUB3 will always come before SUB4 SUB5. This is because SUB2 and SUB3 belong to group 01 while SUB4 and SUB5 belong to group 02. Yet, while the order in which SUB2 and SUB3 will be arranged is random, SUB4 will always come before SUB5. This is because group 02 is defined in column 3 as

non-random. This means that for a request in which SUB1 is the pickup location, the roster may be one of the following:

```
SUB2 SUB3 SUB4 SUB5
SUB3 SUB2 SUB4 SUB5
```

Note that all of these potential suppliers will be added into the roster only if they have available items that are connected to the requested title. This means that if SUB3 does not have a available item the roster will be created as:

```
SUB2 SUB4 SUB5
```

Also note that the potential suppliers (columns 4-13) may be either ADM libraries or sublibraries. If ADM libraries are defined then only one item from that ADM will be put in the queue. If sublibraries are used then all of the sublibraries that have matching items will be put in the queue, creating multiple slots in the queue that belong to the same ADM but to different sublibraries.

The tab_roster table may be defined either in alephe_tab or in the data_tab of the ADM library of the pickup location.

Note that if a supplier is willing to send an item to the desired pickup location (as defined in its tab37 setup) but is not defined in the pickup location's roster in tab_roster, the supplier will be added as last in the queue.

50.3.4. Creating the Request

A title request can be created only if the patron has not exceeded the title request limit that has been defined in the patron's global record.

When a title request is submitted, the checks that are defined in the TITLE-REQ section of the tab_hold_request tables are performed. Each item is tested with the checks that are defined in this section of the item's ADM library table. Roster stops will be placed only on items that pass the checks.

50.4. Fulfilling the Request

When an item is transferred to a pickup location that is not the sublibrary of the item, the patron may be notified either when the item is sent from the owning sublibrary or when it arrives at the pickup location. This is determined by the value of the tab100 variable HOLD-FULFILL-NOTIFY:

- If the variable is set to 1, the system will notify the patron that the request has been fulfilled only when the item arrives at the pickup location
- If the variable is set to 2, the system will notify the patron that the request has been fulfilled when the item is sent from the owning library.

50.5. Loan and Return Anywhere

Where PDQ prerequisites are fulfilled, an item may be loaned and returned anywhere in the consortium. The tab_attr_sub_library table governs a circulation station's cross-institution loan and return permissions just as it governs a circulation station's permissions for loans and returns within the ADM library. Cross-ADM loan-return

permissions can be defined at either the ADM or at the sublibrary level. For example, for a station that belong to USM51 to be able to loan and return items that belong to WID sublibrary of USM50, the following setup is possible in the USM51 setup of tab_attr_sub_library:

```
*.*.*.*        1 LINC5 GDOC5   WID
               3 LINC5
```

Or

```
*.*.*.*        1 LINC5 GDOC5   USM50
               3 LINC5
```

50.6. Staff Permissions

The following permissions control staff privileges to PDQ functions:

- CIRCULATION / TITLE-REQ-DELETE - Permission to delete title requests
- CIRCULATION / Circulation TITLE-REQUEST – Permission to create title requests
- CIRCULATION / Circulation CROSS-INST-RETURN – Permission to return an item not in its owning library
- CIRCULATION / Circulation CROSS-INST-LOAN – Permission to loan an item not in its owning location
- CIRCULATION / TITLE-REQ-UPDATE – Permission to update a title request.

51. Managing Lost Items

The following configurations may be used to set the library's policies with regard to lost items.

tab34

tab34 computes the charges for an item which is declared "lost" either through the LOST function on the Loans tab of the Patron Information in the Circulation module, or as a result of the services Overdue and Lost Billing Notices (cir-50) and Overdue and Lost Billing Summary Letter (cir-51). The table sets the default Notice Cost, Handling Cost and Fixed Price per:

- Sublibrary
- Item Status
- Collection
- Patron Status
- Call Number or Call Number Range
- Item Material Type

The table may be set to use a Fixed Price that is defined in the lost item's record (Z30).

tab18

The table may be used to switch the Notice Cost, Handling Cost and Fixed Price on or off. In addition, other parameters such as VAT rates may be set in this table. Additional charges for lost items that have been previously recalled may be set by using the transaction types 0054 and 0055 and the OVERDUE-RECALL-RATIO tab100 variable (see below).

tab35 and tab_circ_log.lng

These tables may be used to configure the Circulation Logger (Z309) events and statistic events (Z35) that are created when an item is declared lost.

Tab100

The LOST-PROC-STATUS variable may be used to remove a 'lost' process status automatically when the lost item's barcode is scanned into the Circulation system.

The REFUND-RATE variable may be used to set what percentage of the lost charges will be refunded when a lost item is eventually checked it.

The LOST-LOAN-CREDIT-METHOD variable may be used to determine which cash transactions are refunded when a lost loan is eventually returned, and whether the refunding method will be by creating balancing credit charges or by waiving existing debit charges.

The LATE-RET-FINE-WHEN-LOST variable may be used to set whether overdue charges will be charged when the lost item is declared lost or only if and when it is eventually checked in.

Circ.ini

The `LostReturn` parameter in the [`ErrorAction`] section may be used to set and audio warning if an item that is being checked in has been previously declared lost.

The ManualReplacement parameter in the [Lost] section may be used to control whether the Fixed Price may be manually amended by the librarian when the item is declared lost.

52. SMS Messaging

ALEPH Circulation module may be configured to send SMS messages to library patrons in one of the following instances:

- Patron hold request is fulfilled.
- Overdue letter is sent to patron.
- An item that is on loan to a patron is recalled

Note: Sending SMS messages depends on configuring and activating the SMS Server. Please relate to SMS Proxy documentation for information on the required configurations for the SMS Server.

52.1. Sending SMS from ALEPH

The SMS message are sent when the relevant printout is created if:

- The patron has a valid SMS number in his patron card. The SMS number may be input using the Address Information form of the Circulation GUI or using the Patron Registration and Address Information forms in the Web OPAC.
- The patron has agreed to receive SMS messages. The patron agreement is registered in the Global Patron Information and Patron Registration forms of the Circulation GUI, or in the Patron Registration and Address Information forms in the Web OPAC.
- The library has configured the form to trigger an SMS message, as described below.

The SMS related library configurations are:

1. In aleph_start, the location of the SMS server must be configured. For example.

```
setenv server_sms          "10.1.235.14:1894"
```

2. In the ADM library's data_tab directory, the form_sms table is used to set which forms will trigger SMS sending. The following forms currently support SMS sending:
    - hold-request-letter-*xx*. The *xx* is the hold request send action.
    - loan-recall-*x*. The *x* is the recall letter type (1 or 2).
    - overdue-letter-*x*. The *x* is the optional letter number.
    - overdue-summary-*x* . The *x* is the optional letter number.
    - overdue-sum-single-*x*. The *x* is the optional letter number.

The table includes the following columns:

Column 1 – Name of the form

Column 2 – The form number

Column 3 – The SMS text number. This number must be aligned with the codes used in the $aleph_root/error_lng/sms_messages table, as described below.

Column 4 – Whether or not the SMS message will be sent

3. In the $aleph_root/error_lng directory, the sms_messages table includes the SMS messages text. Column 1 of this table is the message text, and must be aligned with the code that is used in column 3 of the $data_tab/form_sms table.

4. If fee charging is required, tab18.lng must be set with the relevant charge. The SMS charge is 0072.

52.2. Sending SMS by Using Third Party SMS Servers

Third party SMS servers may be used for the actual SMS sending. If the $data_tab/form_sms and the $aleph_root/error_lng/sms_messages are set up as explained in the previous chapter, a <sms-message> tag will be created in the printout XML file. This text, along with the <z302-want-sms> and <z302-sms-number> tags information, may be used to activate third party SMS servers for the purpose of sending SMS messages.

53. Patron Name in Japanese – Kanji and Katakana

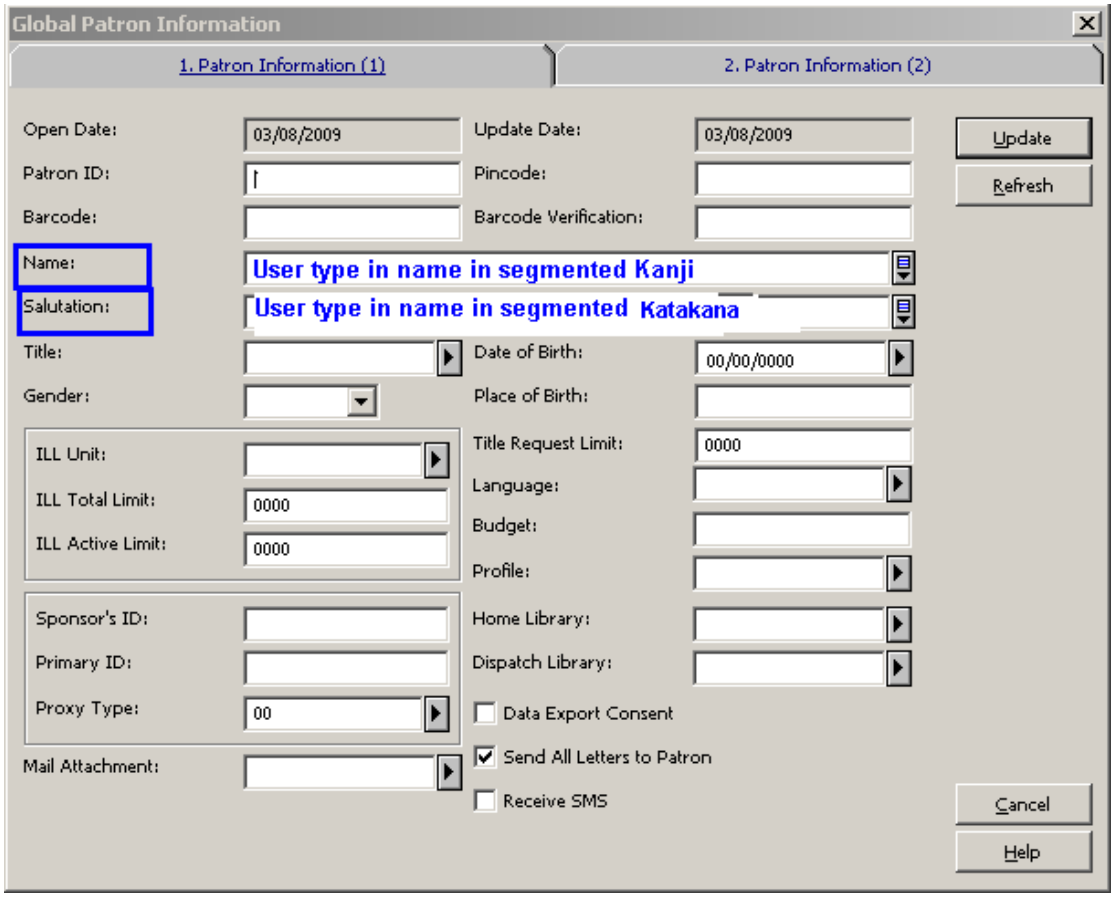
This chapter describes the following activities for using patron names in Kanji and Katakana:

- Storing patron name in Kanji and Katakana
- Sorting patron name list in Kanji
- Searching for patron name in Kanji and Katakana

53.1. Storing Patron Name in Kanji and Katakana

The patron record can maintain patron names in both Kanji and Katakana.

Kanji – The patron name is stored in Patron Record: Name field, Z303-NAME, in segmented Kanji.
Katakana – The patron name is stored in Patron Record: Salutation field, Z303-SALUTATION, in segmented Katakana.



The patron Name (Kanji) is a mandatory field. Users cannot register/update a patron record without populating the Patron Name field.

The Salutation field (Katakana name) can becomes a mandatory field upon setting up a system variable located in ./alephe/aleph_start:

```
setenv  JAPANESE_ENABLED  TRUE
```

The variable can be set to the values FALSE (default) or TRUE.
Setting the variable to TRUE enables special Japanese features such as defining the Patron's Salutation field as a mandatory field.

Upon setting this variable, users cannot register/update a patron record without populating the Patron Salutation field. If this field is missing, the following message is displayed: "Missing Katakana Name".

The ability to define the Salutation field as mandatory using the system variable applies for the GUI-CIRC-Patron Global Information form and Patron Loader (file-20 service).

When JAPANESE_ENABLED is TRUE, the options to register a patron via GUI-Circ- Fast Patron Registration and the Web OPAC-Self Registration forms are disabled (make sure the fast registration flag is turned-off in circ.ini: FastPatronRegistration=N).

:

Notes:

- **Segmented Patron Names** – The patron names must be entered in a segmented Kanji/Katakana. The system is not capable of identifying word breaks for non-segmented phrases.
- **Katakana Name field Caption** – The library may rename the caption of the Salutation field to Katakana Name.

This can be achieved by updating the field name in the PC file .\circ\tab\lng\window.dat

Replace the line:

UserGlobalInfo.Static.Salutation #Salutation:

With:

UserGlobalInfo.Static.Salutation Katakana Name:

- **Letters to Patron Templates** – Some of the letters to patron template files (xsl files) include patron salutation (Z303-SALUTATION). The patron salutation string have to be cleaned-out from the XSL templates so that the Katakana name is not mistakenly quoted in the letters.

53.2. Sorting Patron Name in Kanji

Sorting of the patron name list is set according to the Kanji name.

The patron names list is sorted by Z303-NAME-KEY which is created from Z303-NAME - Kanji name.

The filing of the patron name for sorting purposes is applied as in the following setup:

The sorting form is created according to the BORROWER-NAME-KEY procedure as set in the following configuration tables: tab_filing and tab_character_conversion_line.
Section 55 of ./usr_library/tab/tab_filing should be set with the instance: BORROWER-NAME-KEY as follows:

```
!1 2        3                                    4
!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
55   char_conv            BORROWER-NAME-KEY
```

The instance BORROWER-NAME-KEY should be set in ./alephe/unicode/tab_character_conversion_line

```
          1              2   3        4                                5
!!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
BORROWER_NAME_KEY    ##### # line_utf2line_utf              adm_name_key
```

The above sample setup activates the filing (sorting) routine of the patron name as defined in ./alephe/unicode/adm_name_key.
The library may activate any other filing routine by setting up other routine over BORROWER-NAME-KEY instance.

Note:

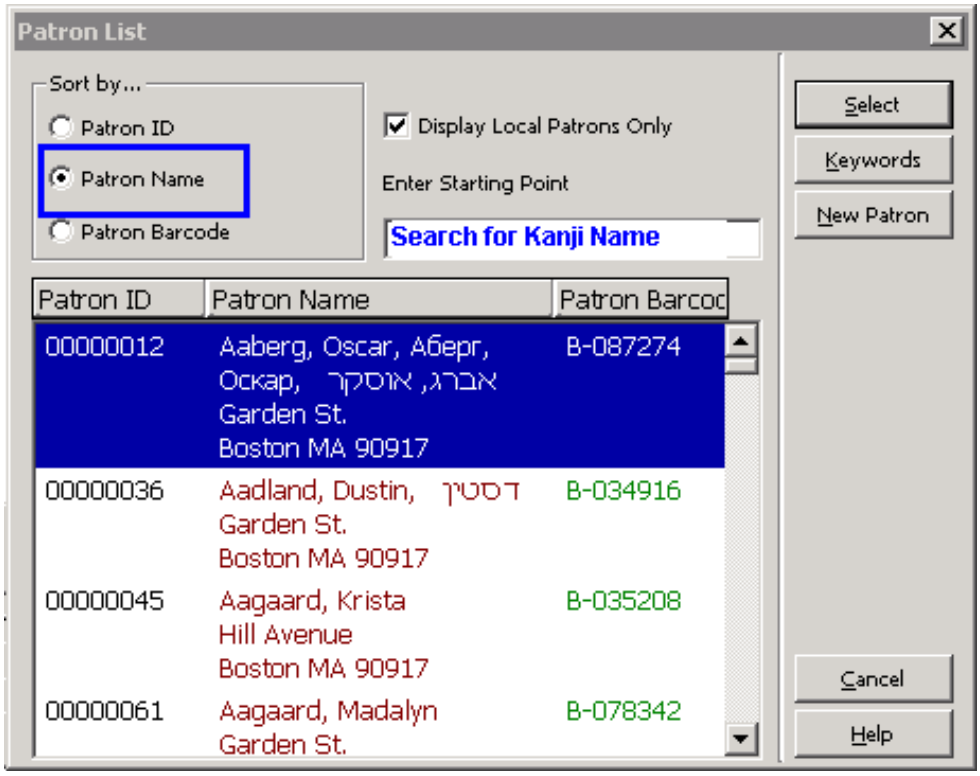
Z303-NAME-KEY length is 50 bytes; hence the sort will not distinguish between names that start with the same 16 Kanji characters.

53.3. Searching for Patron Name in Kanji and Katakana

The search for patron name is enabled for both: Kanji and Katakana.

53.3.1. Search for Patron Kanji Name

Searching for the Kanji version of the name can be done by selecting "Patron Name" in the Patron List dialog box and entering the Kanji name in the “Starting Point” text box..



The result list is sorted according to the Z303-NAME-KEY build-up which uses on the Kanji name.

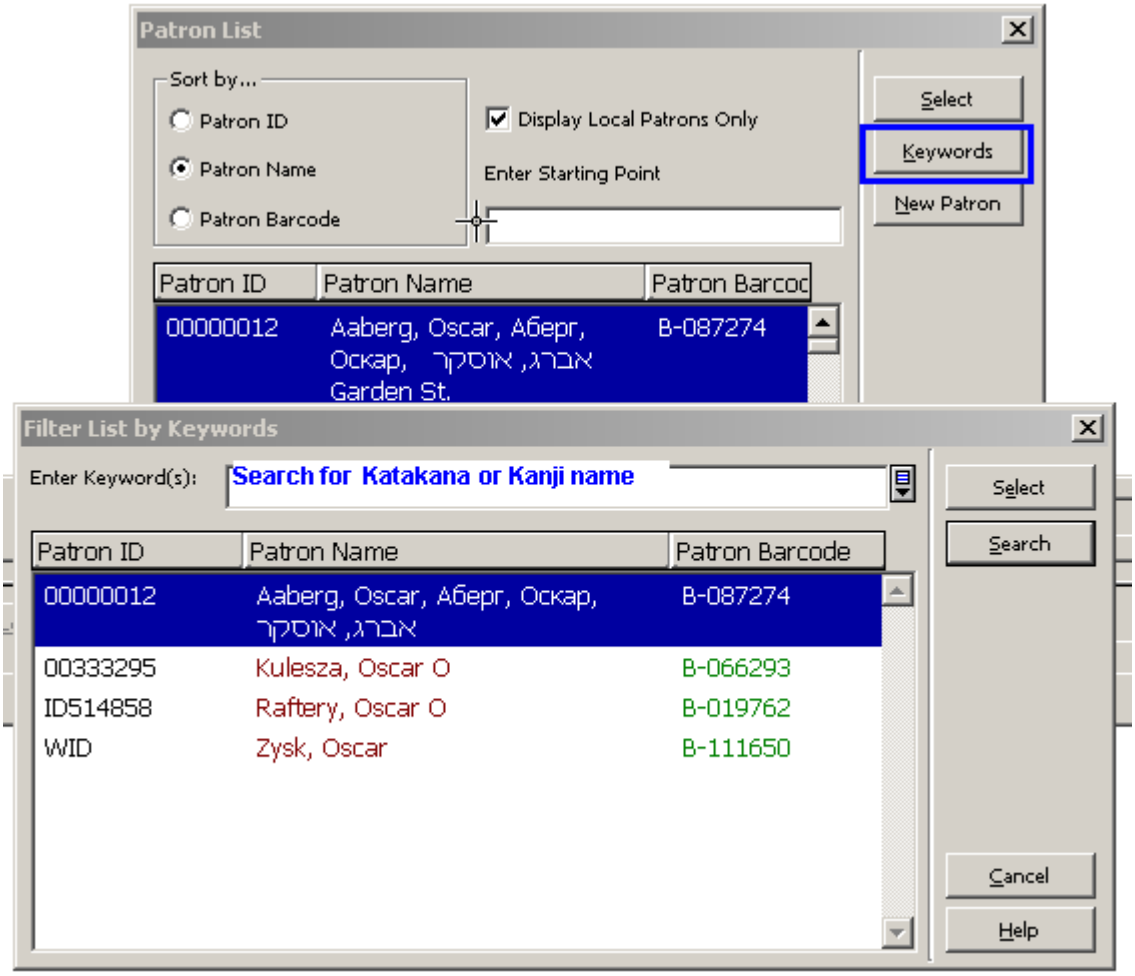
53.4. Patron Keywords Search – Kanji and Katakana

Searching for both Kanji and Katakana versions of the name can be performed by using the "Keywords" feature.

The Z111 keywords feature enables the indexing of keyword out of the Z303-SALUTATION field (Katakana name) and the Z303-NAME field (Kanji name).

Z111 indexes phrases included in certain Z303 fields which enables searching for any words within the indexed string (not just the "Start with" string).

Searching for the Katakana and Kanji version of the name can be done by activating the "Keywords" button on the Patron list and by typing the Katakana/Kanji name at the "Enter Keyword/s" field.



The configuration table ./alephe/tab/z111_index enables the library to decide which fields, out of a pre-defined list of fields, are indexed and searched using Keywords.

53.5. Z111_index table

./alephe_tab/z111_index, allows the library to set for which fields Z111 keywords are created so that only required fields are index when a record is updated or when p-manage-111 batch service is run (build keywords).

The table contains a single column:
Column 1: field to index (30 characters).
This column can contain the following fields (fields which are listed in column 1 are indexed and available for "Keywords Search"):

Z303-NAME
Z303-SALUTATION
Z303-PROXY-FOR-ID
Z303-PRIMARY-ID

Z308-KEY-DATA

Z304-ADDRESS
Z304-ZIP
Z304-EMAIL-ADDRESS

Z310-BARCODE
Z310-ID

Z13-TITLE
Z13-AUTHOR

Z70-VENDOR-NAME
Z70-VENDOR-CONTACT
Z70-COUNTRY
Z70-MATERIAL-TYPE
Z70-ADDITIONAL-VENDOR-CODE
Z70-CONTACT-5

Z700-REC-KEY
Z700-NAME
Z700-GEN-INFO

Z76-EXTERNAL-BUDGET
Z76-CURRENCY
Z76-NAME
Z76-DEPARTMENT

Z72-VENDOR-ADDRESS

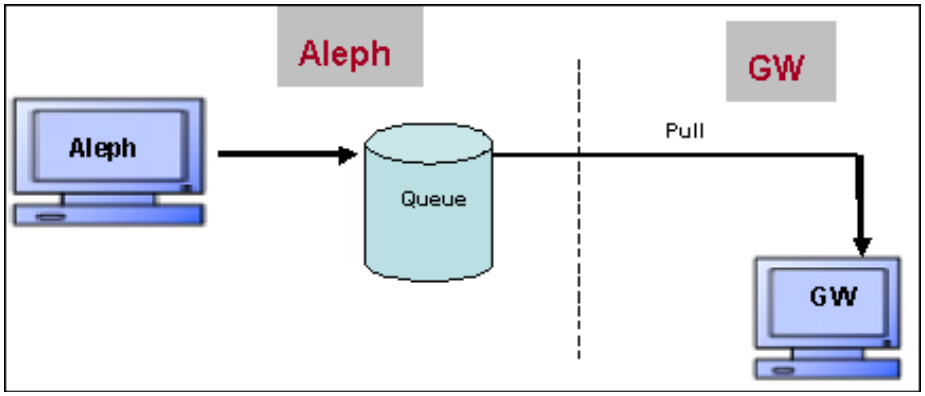
For the patron record, it is highly recommended to index only the Kanji and Katakana versions of the name (Z303-NAME and Z303-SALUTATION) and disable the indexing of other fields (e.g. Z304-ADDRESS), as indexing of other fields introduces inconsistent noises.

54. Aleph – Gateway Compatibility

A library may employ a third-party gateway system that controls patron access to library services. Aleph is capable of making information available to be used by the gateway system. This information is stored in the Oracle Advanced Queue, which resides in the Aleph database (Z502).

Aleph only makes the information available.The gateway system is responsible for accessing the information from the Aleph queue, analyzing it according to its own criteria, and determining whether to allow or block access to a patron.

The following diagram illustrates the architecture of **the** Aleph-Gateway compatibility:



54.1. Events that Trigger Messages for the Gateway

Aleph can be configured to have the following circulation events trigger a message for the gateway system.

- Photocopy (Z38) request created
- Photocopy request supplied
- Photocopy request refused
- Photocopy request deleted
- Loan (Z36) created
- Loan deleted
- Loan returned

Circulation events are recorded only for relevant users. The relevant users are set in the usr_library table circ_events_users (see details below: Activation and Setup).

Aleph can be configured to have the following user events trigger a message for the gateway system.

- User ID (Z308) created
- User ID deleted
- User barcode updated
- Local patron record (Z305) created
- Local patron expiry date changed
- Local patron status changed
- Patron's block created (Z303)
- Patron's block updated
- Patron's block deleted

54.2. Configuration and Activation

Z502 and 1 aleph_q_gate

To configure and activate Aleph - Gateway compatibility:

1. Create the Z502 Oracle queue table and aleph_q_gate:

   a) Add the Z502 table to file_list to the USR library.

   For example: ./usr00/file_list:

```
!1          2                    3                  4                    5
6
!!!-!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!
TAB z502               333K             0K               ts0
```

   b) In usr_library, create the Z502 Oracle Queue table, using util/a/17/7/1.

   c) In usr_library, create the aleph_q_gate queue in the Z502 Oracle Queue Table, using util/a/17/7/2.

   d) In usr_library, start the aleph_q_gate queue, using util/a/17/7/3.

   Note that the name of the Oracle Queue in util/a/17/7/3 is "aleph_q_gate".

Note:
If the Z502 Oracle queue table already exists, it can be dropped and re-created:
1. Stop the queue using util/a/17/7/4.
2. Drop the queue using util/a/17/7/5.
3. Drop the queue table using util/a/17/7/6.

2. Set the relevant users (circ_events_users) for which circulation events are generated by editing the usr_library table circ_events_users.

The circ_events_users table defines the local user status and the sublibraries (Z305) for which circulation events are created. Circulation events are recorded only for users that have a local user record (Z305) in the relevant sub-library (Z305-SUB-LIBRARY) with one of the defined patron statuses (Z305-BOR-STATUS).

For example: ./usr00/tab/ circ_events_users

```
Column 1 – Patron local sub library (Z305-SUB-LIBRARY).
             Set ##### for all sublibraries.
Columns 2 thru 7 - Patron status (Z305-BOR-STATUS).
             Set ## for all statuses

! 1   2  3  4  5  6  7
!!!!!-!!-!!-!!-!!-!!-!!>
MED   01 02 03 04 11 12
TML   01 02 03 04 05 06
WID   01 02 03 04 05 10
```

3. Activate www_server and pc_server.

55. Aleph and SSO Integration

Aleph is capable of storing information regarding specific events that affect a third-party Single Sign On system (SSO).

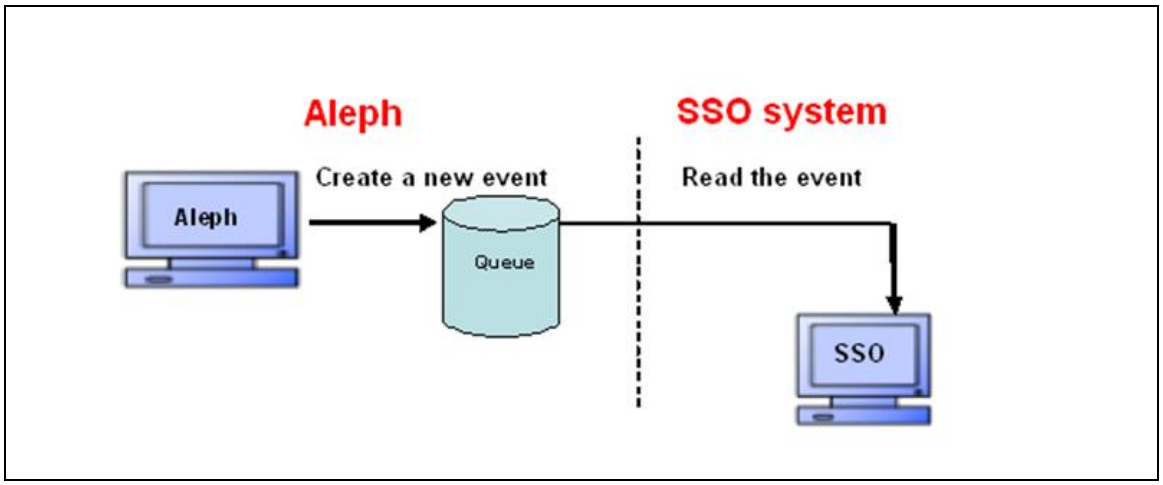
Each time one of the following actions occurs, Aleph stores the relevant patron information in the Z503 Oracle table, Relevant Actions:

- New Patron
- Update Address
- Update Password

This Z503 information is stored in Oracle advanced queue, which resides in the Aleph database.

It is the third-party SSO system's responsibility to pull the information from Aleph's queue, analyze it according to its own logics, and update the SSO data (for example, update patron name and password).

The following diagram illustrates the architecture:



55.1. Events that Trigger Messages for the SSO

The following is a list of patron events that can be configured to trigger a real-time Z503 message for the external SSO system.

- New Patron in OPAC/GUI (event type: NP)
- Address change in OPAC/GUI (event type: UA)
- Password change in OPAC/GUI (event type: UP)

Each Z503 reports the patron relevant data: Z303/304/305/308. Note that there are cases in which a single action triggers several Z503 records, for example, when creating a new patron record via: Circulation> Patron Menu > New Patron (or Circ > New Patron Icon) > Update button.

With the above action, the user creates a new Z303 patron record that affects the patron's password information (Z308) of type 00 and type 01; therefore two "NP" Z503 records are created, one for Z308-type 00 and one for Z308-type 01.

For Z503 record structure, refer to Oracle table documentation: Z503.

55.2. Activation and Setup

Z503 and aleph_q_sso

To create a Z503Oracle queue table and aleph_q_sso:

1) Add the Z503 table to file_list in the USR library.

For example: ./usr00/file_list:

```
!1          2                    3                  4                    5
6
!!!-!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!
TAB z503               333K             0K               ts0
```

2) In the usr_library, create the Z503 Oracle queue table, using util/a/17/7/1.

```
Enter library name  : usr00
You asked to create from library USR00

Enter table name to Create: z503
enter yes to create z503 table for USR00 : yes
This is your last chance!!! Would you like to change your mind? no
```

3) In the usr_library, create the aleph_q_sso queue in the Z503 Oracle queue table, using util/a/17/7/2.

3) In usr_library, start the aleph_q_sso queue, using util/a/17/7/3.
4) In usr_library, create the dequeue user, using util/a/17/7/7.

Note:
If the Z503 Oracle queue table already exists, it can be dropped and re-created:
1. Stop the queue using util/a/17/7/4.
2. Drop the queue using util/a/17/7/5.
3. Drop the queue table using util/a/17/7/6.

56. Log for Patron Record Changes

The library can record in a log changes made to the patron record, the name of the user who implemented the change, and the time and date the change was made.

The Patron Record Changes log is created as follows:

1. The log data is saved in Z307 ORACLE table Z307 which is recorded in the usr_library.

2. The usr_library onfiguration table, tab_patron_log_.lng, defines the actions that trigger patron log creation (Z307).
3. The circulation batch service, Patron Record Log Changes Report (cir-27), produces a report of changes made on patron records.

56.1. tab_patron_log.<lng >

tab_patron_log.<lng> is located in the tab directory of the usr_library. It is used for configuring the action codes (column 1) that trigger the Z307 Patron Log records and their textual description (column 3). Only the actions that are activated (column 2 = Y) can be triggered. Use column 2 to have Aleph ignore action triggers.

The following is a list of possible logs (numeric code–col.1 and text decription-ol.3):

001 - Update of global patron record-Details
002 - Update of global patron record-Blocks
003 - Update of local patron record-Details
004 - Update of local patron record-Blocks
005 - Update of local patron record-Privileges
006 - Update of patron addresses
007 - Update of patron IDs
010 - Delete of global patron record
011 - Delete of local patron record
012 - Delete of patron address record
013 - Delete of patron ID
020 - New global patron record created
021 - New local patron record created
022 - New patron address record created
023 - New patron ID created

Structure of the table:
COL 1. The action type code that cause a Z307 patron change log record to be written
COL 2. Active/Non active log: Y/N
COL 3. Description of the action

The following is an example of the table:

```
!1  2           3
!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
001 Y Update of global patron record-Details
002 Y Update of global patron record-Blocks
003 Y Update of local patron record-Details
004 Y Update of local patron record-Blocks
005 Y Update of local patron record-Privileges
006 Y Update of patron addresses
007 Y Update of patron IDs
010 Y Delete of global patron record
011 Y Delete of local patron record
012 Y Delete of patron address record
013 Y Delete of patron ID
```

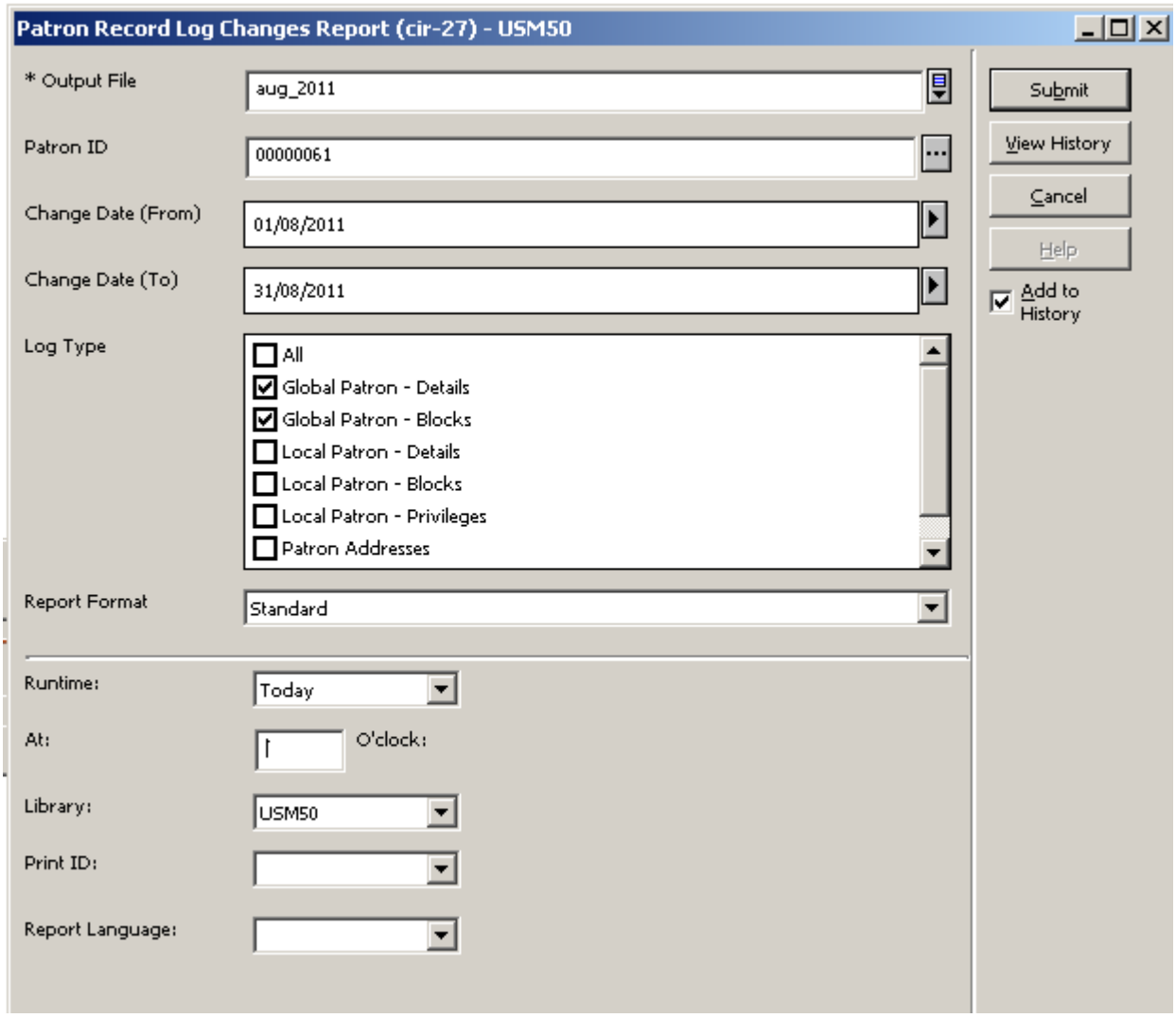
```
020 Y New global patron record created
021 Y New local patron record created
022 Y New patron address record created
023 Y New patron ID created
```

Notes:

- Only action types that are set in the table create a Z307 record.
- Each one of the actions defines a group of fields that when updated, trigger the log.
- If the table does not exist or is empty, no log is created.

56.2. Patron Log Record Changes Report

The patron change log actions can be reported using the Patron Record Log Changes Report (cir-27) batch service. Access this service from **Circulation Services** > **Patron Record**.



This service produces a report of changes made for patron records. It retrieves Z307 records from the usr_libray.

For libraries that share the user file with other ADMs (TAB100-USER-SHARING=Y),the batch service retrieves all of the changes made to patrons that belong to the shred user environment, even if changes were triggered by another ADM library (except for logs that are related to Patron Local Record (Z305) which are retrieved only by the ADM that triggers the change).

For library that has their own user file (non-share user file TAB100-USER-SHARING=N), the batch retrieves only logs related to the library's patrons.

The service produces a report containing the log information. This information includes the Patron ID, the date and time of the change, and the user who implemented the change.

57. Two Patron Name Fields

Your library can use two fields to hold the patron name: Last Name and First Name (instead of a single Name field).

The functionality and interface of the two new fields depend on your library setup, so that only libraries that configure the formatted name structure can use the new functionality.

The setup is implemented by a new ./alephe/tab/tab100 flag: PATRON-NAME-FORMAT. If PATRON-NAME-FORMAT is not defined, the current single name field is used (as it is currently).

The two new fields build the Z303-NAME according to the routine set in the new TAB100 flag.

The migration of current Z303-NAME into the two new fields is done by running the new Convert Patron Name (p-com-04) batch service.

57.1. PATRON-NAME-FORMAT

Update the ./alephe/tab/tab100 flag: PATRON-NAME-FORMAT with the relevant value. This flag determines the Z303-NAME format that is created form the patron first name and last name. It is also used by the Convert Patron Name (p-com-04) batch service to convert the single name field into two name fields.

The tab100 flag can have one of the following values:

- `L F`
- `L,F`
- `L;F`
- `F L`
- `F,L`
- `F;L`

Where F = First name, L = Last name.

57.2. Migrating the Current Single Name Field (Z303-NAME) into Two Name Fields

The Migrate Patron Name (com-04) batch service converts the current Z303-NAME into the two new fields. It is accessed via Services menu > Patron Records page of the Circulation module.

Libraries that want to apply the two new patron name fields should activate the batch service. The goal is to create a proper data and display of the name fields.

The batch can only be run by the library staff after migrating to Aleph 22. It can be run also at a later stage if the library decides to use the two name fields.

You only need to run the batch service once. The batch service must be run before activating the GUI and patron flows so that the two name fields are handled properly (otherwise, there will be no data in the two new name fields).

The batch service assumes that Z303-NAME data is correctly entered. The batch output file reports problematic cases that may require staff verification.

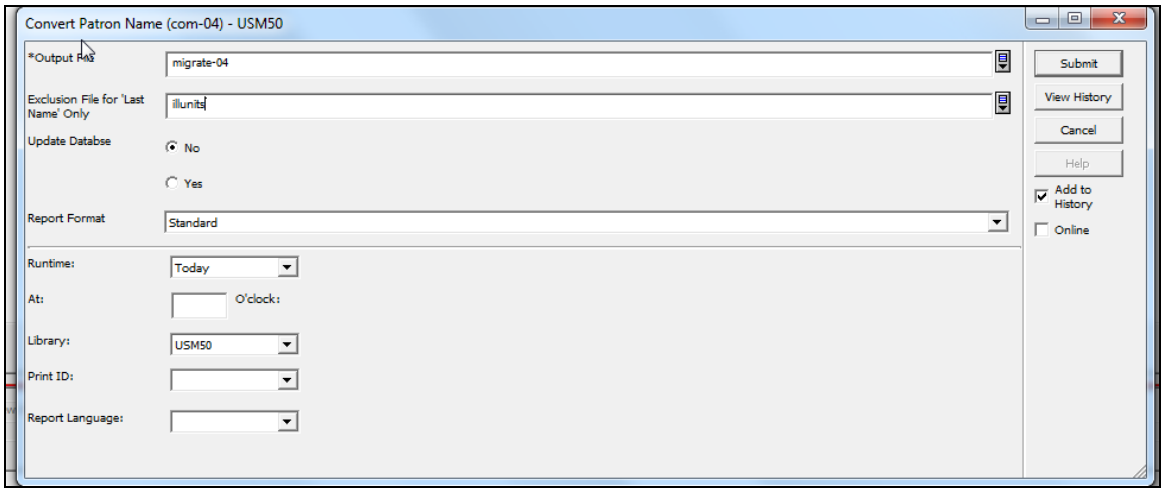
The script looks for PATRON-NAME-FORMAT (if it exists) and migrates Z303-NAME into the two new fields.

If one or more of the two name fields are already populated, the batch does not update existing data.

The migration batch service applies PATRON-NAME-FORMAT to convert Z303-NAME into Z303-FIRST-NAME and Z303-LAST-NAME.

The convert service (com-04) can be run from ADM, and it affects all the patrons in all ADMs.

The following is an example of the batch interface:



57.2.1. Input Parameters

Output file: The report output file name that contains a list of affected names (patron-name-convert.xsl).

Exclusion File for 'Last Name' Only: Input file of patron IDs that are migrated to Last Name only, such as: ILL Unit, Administrative units, and Sublibraries. The file should be located in the ./alephe/scratch directory.

Update Database: Yes/No. To run the service in a preview mode, select 'No'. Select 'Yes' if you want Z303-FIRST-NAME and Z303-LAST-NAME to be updated. The batch applies the TAB100 flag: PATRON-NAME-FORMAT. If the flag is not defined or it is illegally set, and the batch is not executed, an error message is displayed: "tab100 PATRON-NAME-FORMAT is not activated".

If it is correctly set, the batch service populates Z303-FIRST-NAME and Z303-LAST-NAME according to procedure set in the TAB100 flag: PATRON-NAME-FORMAT.

If there is an input file in "Exclusion file for 'Last Name' only", the system migrates Z303-Name into Z303-Last name (leaving Z303-First-name empty). The input file should be set in advanced by the library so that Z303 records, which represent ILL Unit/Sublibraies/etc, is populated with Last name only.

If Z303-FIRST-NAME and/or Z303-LAST-NAME are already populated, there is no override. Neither field is updated. The output file includes these entries with the indication "Already populated".

The batch generates two output reports. For example, if the output file name is "patron", two reports are generated: "patron" and "patron.verify".

The "patron" file include all patrons that were successfully converted (status is "OK" or "OK-Exclusion input file"). "patron.verify" includes all patrons whose name should be verified by staff (status is "verify name" or "already populated").

The output report is sorted by patron's ID (Z303-ID).

57.3. Implementation Notes

1. Set the ./alephe/tab/tab100 PATRON-NAME-FORMAT flag with one of the following values:

- `L F`
- `L,F`
- `L;F`
- `F L`
- `F,L`
- `F;L`

Where F = First name and L = Last name.

2. Update alephe/www_f_lng/bor-new with the two new fields:

In my_reset function:

a. Replace:

```
if (document.form1.elements[i].name == "M_Z303_NAME" ||
```

With:

```
if (document.form1.elements[i].name == "M_Z303_LAST_NAME" ||
document.form1.elements[i].name == "N_Z303_FIRST_NAME" ||
```

b. Add:

```
 <tr>        <td    class=td2    width=20%    nowrap><label
for="last_name">Last Name: </label></td>
  <td class=td2 width=1%>
      <span class=text2 title="Mandatory">*</span>
  </td>
  <td class=td2>
      <input    id="last_name"    size=30    maxlength=200
name=M_Z303_LAST_NAME value="$3000"></td>
 </tr>
 <tr>
  <td        class=td2        width=20%        nowrap><label
for="first_name">First Name: </label></td>
  <td class=td2> </td>
  <td class=td2>
      <input    id="first_name"    size=30    maxlength=200
name=N_Z303_FIRST_NAME value="$3100"></td>
 </tr>
```

c. Remove:

```
 <tr>
  <td  class=td2  width=20%  nowrap><label  for="name">Name:
</label></td>
  <td class=td2 width=1%>
      <span class=text2 title="Mandatory">*</span>
  </td>
  <td class=td2>
      <input      id="name"      size=30      maxlength=200
name=M_Z303_NAME value="$0100"></td>
 </tr>
```

d. Replace:

```
<body topmargin="0"
      onload="window.document.form1.reset();
window.document.form1.N_Z303_NAME.focus()">
```

With:

```
<body topmargin="0"
      onload="window.document.form1.reset();
window.document.form1.M_Z303_LAST_NAME.focus()">
```

4. If localization is needed, add the following lines to $alephe_root/error_lng:

```
a. into pc_com_c0121:
   0051 0007 L tab100 PATRON-NAME-FORMAT is not activated.
   0052 0007 L Last name must be entered.
b. into pc_cir_c0468:
   0050 0001 L tab100 PATRON-NAME-FORMAT is not activated.
   0051 0009 L Last name must be entered.
```

```
c. into www_f_heading:
   4921 L tab100 PATRON-NAME-FORMAT is not activated.
   4922 L Last Name
```

4. If localization is needed, add the following files:
- $alephe_root/error_lng/p_com_04
- /form_lng/patron-name-convert.xsl

5. If localization is required, add the following lines to ./alephe/pc_b_lng/menu-circ.xml:

```
<item>
<display>Convert Patron Name (com-04)</display>
<file>p-com-04</file>
</item>
```

6. If localization is required, add the following line to ./alephe/tab/user_function.lng:

```
PATRON-S               L Patron Services                    COM-04
L Com-04 : Convert Patron Name
```

7. If localization is required, add the following file ./alephe/pc_b_lng/p-com-04.xml

8. If localization is needed, add the following lines to:
./circ/tab/lng/window.dat:

```
 a.   UserGlobalInfo.Static.LastName          Last Name:
      UserGlobalInfo.Static.FirstName         First Name:
 b.   PatronRegDlg.Static.LastName            Last Name:
      PatronRegDlg.Static.FirstName           First Name:
./alephcom/tab/lng/window.dat:
 a.   UserForm.Static.LastName                Last Name:
      UserForm.Static.FirstName               First Name:
```

58. Aleph Support for RFID

58.1. Overview

Aleph supports communication with several RFID products: Nedap, 3M, and Bibliochip (Bibliotheca).

Radio Frequency Identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be attached to or incorporated into a product for the purpose of identification using radio waves. Chip-based RFID tags contain silicon chips and antennas.

RFID tags can be used in various implementations for attaching identification or other information to any object. Once in their receiving range, radio-enabled machines can identify the object or read other information that is attached to the object and use it a large range of purposes.

RFID tags are mainly used in libraries for the following purposes:

- Automatically identifying an item during the check-in and check-out process.

Replacing barcode scanners and other traditional methods of identifying items, an RFID-enabled circulation desk can use the RFID reader to identify the item that is on the check-out or check-in counter by reading its RFID tag. This can save the library staff the time and effort that barcode scanning entails.

- Using security gates to detect an un-authorized check-out.

  RFID-enabled security gates can be placed at the library entrances. The gates can read the RFID tag of every item that comes into their radio range and detect whether or not the item has been tagged as 'allowed for check-out'. If an item leaves the library without the RFID tag being updated, the gates will issue an alarm for the staff to act upon.

- Inventory management

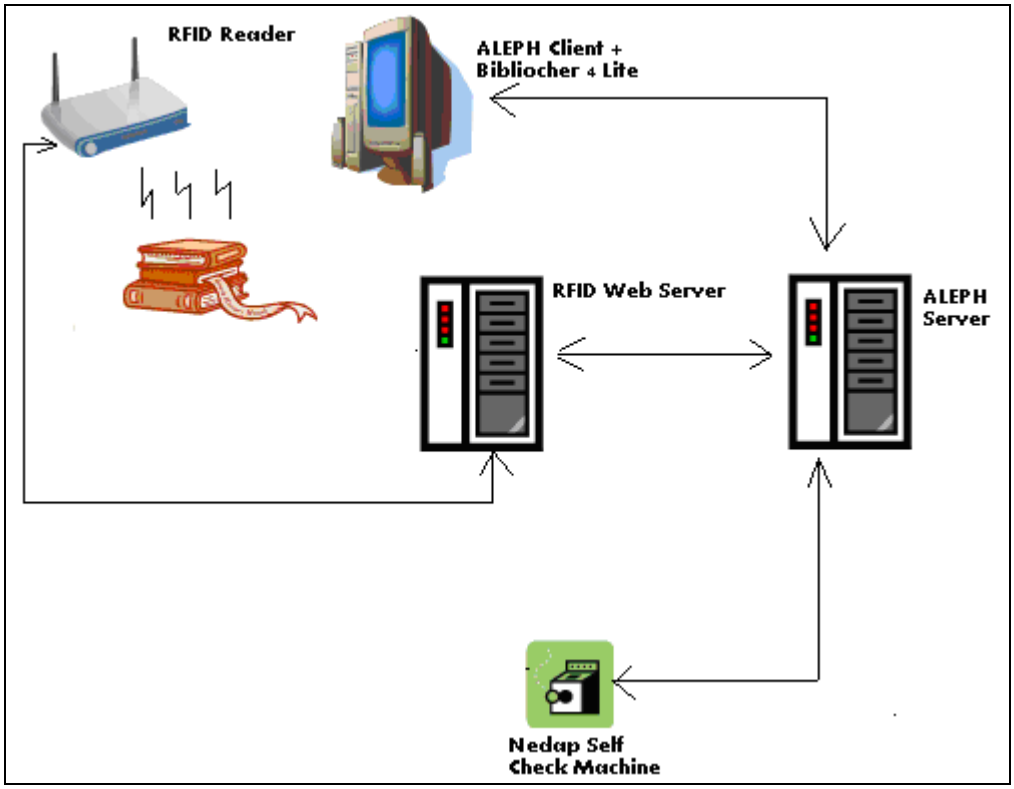
  RFID-enabled machines are very helpful for processing inventory management actions. For example, automatic sorting of items can be done by RFID-enabled machines that sort items based on information stored on the RFID tags. Also, a physical inventory check can be much easier to perform if the item barcodes are read by an RFID-enabled scanner rather than a barcode scanner.

RFID tags and readers of different vendors are gaining popularity. The purpose of this document is to describe how Aleph can be set up to work with in a library that is RFID-enabled, where the RFID environment is based on one of the following RFID tags: Nedap, 3M and Bibliochip.

The RFID-Aleph connection uses the staff workstation's serial interface and requires a proper and fully functioning setup of the RFID software. The Aleph-RFID interface is an integral part of Aleph, so you do not have to install any additional modules.

58.2. System Architecture

The following diagram shows how the Aleph system and the RFID readers interact:



The following is a description of the participating elements and their relationships:

- Aleph Client + the RFID vendor software (Nedap/3M/Bibliochip)

  The Aleph client station is equipped with the RFID vendor software. The vendor software is the component which updates Aleph with the information on the item that has been detected by the RFID Reader.

- RFID Reader

  The RFID reader is able to read the RFID tag information that is found within its radio range and passes the information to the Aleph \RFID client. Communication with the RFID reader is done over UDP.

  Note

  For the system to work properly, only one RFID tag can be in the radio range of the RFID Reader when it is being used.

- Aleph Server

  The Aleph server interacts with the Aleph client to execute the check-out\check-in actions and the item update actions. When actions that affect the RFID tag are triggered, the Aleph server updates the RFID Web Server on the details of the action and on the RFID reader that is responsible for updating the RFID tag.

- RFID Web Server

  This is a server running the RFID Web Server software (Nedap/3M/Bibliochip). This server receives from the Aleph server information on where the updated RFID tag is found (that is, which RFID reader should act on the update request), and on what the update action is (enabling security bit, updating item information, and so on). The RFID Web Server triggers the write action of the RFID Reader,

and obtains success or failure information which is reported back to the Aleph server.

58.3. Functionality

The RFID-enabled Aleph system supports the following RFID-based actions:

- Identifying items
- Checking-out and checking-in items.
- RFID-enabled self check actions.
- Update of the RFID tags when updating the item information.

58.3.1. Identifying Items

The RFID-enabled item is identified by the GUI modules in the following manner:

The Aleph client has the Bibliocheck4Lite RFID software running. The Bibliocheck4LiteSettings.xml file installed on your workstation must contain pre-defined settings such as the following in order to carry out the following actions:

Disclaimer

Third party products such as Nedap/3M/Bibliochip are not under the control of Ex Libris, and Ex Libris is not responsible for any changes or updates to product. Furthermore, Ex Libris is not responsible for providing any end user support with respect to Nedap/3M/Bibliochip. All end user support requests should be addressed to the product's vendor.

- Define the Bibliocheck4Lite software to connect to the RFID Reader using UDP.

```
<RfConnection>
        UDPSocketConnection
</RfConnection>
```

- Define the UDP location of the RFID reader.

```
<UDPSocketConnection>
    <ReaderAddress>
        IP address of the RFID Reader
    </ReaderAddress>
    <ReaderPort>
        79
    </ReaderPort>
</UDPSocketConnection>
```

- Define the Bibliocheck4Lite software to read only the tag identification.

```
<AutoMode>
        Manual
</AutoMode>
```

- Define the Bibliocheck4Lite software to always poll the RFID Reader.

```
<AutoStart>
        True
</AutoStart>
```

- Define the input string's delimiter.

```
<DelimiterString>
\r
/DelimiterString
```

In your Aleph GUI, the cursor is placed in the toolbar in which the item barcode is to be read. For example, if the item is required to be checked out to a patron, the Circulation Loan tab will be activated, and the patron ID will be read. Following that, the cursor is placed in the item bar.

The RFID-tagged item is placed in range of the RFID reader. The item ID is automatically read into the active bar. In the example above, the barcode that has been read by the reader is read into the Item Bar of the Circulation Loan Tab. The Aleph module workflow can continue as if the item barcode was manually scanned in.

58.4. Check-Out and Check-In

Whenever an item is checked out, the RFID Reader is triggered to turn off the security bit.

Whenever an item is checked in the RFID Reader is triggered to turn on the security bit.

If during the check-out or check-in actions the RFID tag has been successfully updated, the GUI can issue a success message such as:

RFID transponder successfully updated.

Whether or not this message is issued depends on the value of the `SuccessMessage` parameter, found under the [`RfidMedia`] section of circ.ini.

If during the check-out or check-in actions the RFID tag has not been successfully updated, the GUI issues an error message, such as

Failed to update RFID Transponder.

The error message is included in a dialog box that allows you to carry out two possible actions:

- To close the error dialog box without further action, click **OK**.
- To retry the RFID update action, by resending the message click **Retry Transponder**.

58.5. Self-Check

Self-check is done by using RFID-enabled self-check machines. The self-check machines can read and update the RFID tags, and update the Aleph server.

58.6. Registering Item Information

When updating the item, the current item's data is written to the transponder which is in range of the RFID reader.

If the RFID tag has been successfully updated, the GUI can issue a success message such as:

RFID transponder successfully updated.

Whether or not this message is issued depends on the value of the `SuccessMessage` parameter, found under the [`RfidMedia`] section of circ.ini, catalog.ini and acq.ini.

If the RFID tag has not been successfully updated, then the GUI will issue an error message, such as:

Failed to update RFID Transponder.

The error message is included in a dialog box that allows you to carry out two possible actions:

- To close the error dialog box without further action, click **OK**.
- To retry the RFID update action, by resending the message click **Retry Transponder**.

58.7. Configuring RFID- Nedap, 3M, and Bibliochip

58.7.1. Configuring RFID Reader Location and Vendor

The [`RfidMedia`] section in alephcom.ini defines the location of the station's RFID reader in the `RfidReaderAddress` variable and the system's RFID vendor in the Vendor variable. The possible values for the Vendor variable are Nedap, 3M, or Bibliotheca.

To define Nedap as the library RFID product, set the Vendor environment to the following:

```
Vendor=Nedap
```

To define 3M as the library RFID product, set the Vendor environment to the following:

```
Vendor=3M
```

To define Bibliochip as the library RFID product, set the Vendor environment to the following:

```
Vendor=Bibliotheca
```

For example:

```
[RfidMedia]
RfidReaderAddress = 10.1.235.173
Vendor=Nedap
```

58.8. Activating Reader and Success Messages

To activate the RFID transponder update from the Aleph client, set the ActivateReader parameter, found under the [`RfidMedia`] section of circ.ini, catalog.ini and acq.ini, to Y.

To issue a message indicating a successful write action to the RFID Reader, set the `SuccessMessage` parameter, found under the [`RfidMedia`] section of circ.ini, catalog.ini and acq.ini, to Y.

To add an additional RFID button to the items list, allowing the updating of the RFID tag without making a change in the item information, set the INITIATED-RFID parameter, found under the [`RfidMedia`] section of circ.ini, catalog.ini, and acq.ini to Y.

For example:

```
[RfidMedia]
ActivateReader=Y
SuccessMessage=Y
INITIATED-RFID=N
```

58.9. Configuring the RFID Web Server

Your Aleph installation contains a table, named tab_rfid.conf, for defining RFID settings. It is located in the ADM library's data_tab directory. Like tab_ncip.conf, the table defines RFID settings in a free `variable=value` format, for example:

```
[General]
rfid_server_address = 10.1.235.173
timeout = 060
Nedap_item_format = FrenchLabel_native
```

rfid_server_address

This variable identifies the IP address on which the RFID Web Server is found.

time_out

This three-digit variable defines the number of milliseconds that the RFID Web server waits for a response from the RFID Reader before issuing a timeout.

Nedap_item_format

This variable identifies the write data format that is used when activating the RFID Web server for an item update action. The possible value is:

FrenchLabel_native - Writes data according to the FrenchLabel data form FR01. The system sends the WriteFrenchLabel_nativeSoapIn and waits for an incoming WriteFrenchLabel_nativeSoapOut message with the action result.

The information to be written into the SOAP message's tags is set in the tab_rfid and tab_rfid_translate tables (see below).

For more information about tab_rfid.conf, refer to the table's header.

58.10. Configuring writing of action data

The data format that is used when updating RFID tag information is configured in the Nedap_item_format variable of the tab_rfid.conf table. The Aleph item information that is written to the tag is configured in the tab_rfid and tab_rfid_translate tables.

58.10.1. tab_rfid

The tab_rfid table defines which fields from the item record are expanded into the transponder record (RFID).

- Column 1 lists the element names of the update-item message. In other words, if the FrenchLabel_native format is defined in tab_rfid.conf, then the configurable elements of the WriteFrenchLabel_native element will be listed in column 1 of this table.
- Column 2 lists the information that will populate the update-item message element. The Aleph information that is available for configuration is derived from the Z30 (Item records) Oracle table. The information is configured in this table by specifying the field name. Constants can also be used as input for SOAP message components.
- Column 3 defines if the code of the item field is to be translated to name (e.g. sublibrary code to sublibrary name). This option is available for:
  - LibraryCode tag – using the SUB-LIB section of tab_rfid_translate
  - LogisticPartGroup tags – using the LOGISTIC section of tab_rfid_translate
- Column 4 defines the character conversion routine that is used. The default value is UTF_TO_RFID.
- Column 5 is for 3M usage only. It defines the field type: (I)Integer or (S)String.

Example of a setup for the FrenchLabel_native format:

```
!
!              1                                2                3         4          5
!!!
```

```
SUB-LIB         02    MED        Medicine Library
SUB-LIB         03    HYL        East Asian Library

LOGISTIC        11    GEN        General Collection
LOGISTIC        02    ARC        Archives Collection
LOGISTIC        01    WID        Main Library
LOGISTIC        05    HYL        East Asian Library
LOGISTIC        01    BOOK       Book
LOGISTIC        44    ISSUE      Issue
```

For more information, refer to the table's header.

System Librarian's Guide – Course Reading and Reserves

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 OVERVIEW 5

2 SETUP 6

2.1 Creating the Course Reading Library 6
2.1.1 Application of Union View 6

2.2 Linking the Course Reading Library to the ADM library 6

2.3 Staff User Privileges 6

2.4 Patron Record for Each Course Reserves Unit 6

3 COURSE RECORDS 7

4 READING LIST RECORDS 8

4.1 Linking fields 9

4.2 expand_doc_course 10

4.3 Grouping of Expanded Fields 10

4.4 expand_doc_merge_union_crs 11

5 COURSE PERIOD AND LOGICAL BASES 11

5.1 Period 11

5.2 Logical Bases 11
5.2.1 Period 11

6 INDEXING 12

6.1 Browse Indexing (tab11_acc) 12

6.2 Search Indexing (tab11_word) 13

6.3 Direct Indexing (tab11_ind) 13

7 E-RESERVES 13

7.1 Introduction to E-reserves 13

7.2 Copyright and Use Permission Form 14

7.3 Usage Permission 14
7.3.1 Usage Permission Configuration 14

7.3.2 Setting up Usage Permissions 15

8 CONFIGURATION INFORMATION 16

8.1 tab_expand 16

8.2 Hold Request Start Date for Pick List 16

8.3 Submitting a List for Course Reading from OPAC 17
8.3.1 E-mail Address 17
8.3.2 Directory for Saved List 17

8.4 Linking 18

8.5 Course Reading GUI Configuration 18
8.5.1 Display Course Information 18
8.5.2 Quick Cataloging 18
8.5.3 Search Form for Course Reading 18
8.5.4 Display Requester IDs 18
8.5.5 pc_tab_col.<lng> 18
8.5.6 pc_tab_exp_field.<lng> 19
8.5.7 View ADAM objects in the Course Reading OPAC 19

9 GUI SERVICES TEMPLATES 19

10 UNION VIEW 20

10.1 expand_doc_merge_union_crs 20

10.2 Configuration for Union View 20
10.2.1 tab100 20
10.2.2 tab_base 20
10.2.3 union_global_param 21
10.2.4 tab_merge_union 21
10.2.5 tab_expand 21
10.2.6 www_tab_short.<lng> 21
10.2.7 edit_doc_999.<lng> 22
10.2.8 Union view batches 22

10.3 Union View Indexing 23

1 Overview

The Course Reading and Reserves operations are managed via the Course Reading node in the Administration tab of the Circulation GUI client. The System Librarian is responsible for setting up the Course Reading library, adapting dropdown lists and display parameters, and maintaining Course Reading Databases.

The building blocks for Course Reading and Reserves are Course records and Bibliographic records. The Course records are the Z108 Oracle table. There are two types of BIB records, and it is important to understand the difference and functionality of each. One type of BIB record is the cataloging metadata record, which contains the description of the work (a book, a journal article, etc.). The other type of BIB record contains linking data, to connect between the course and the bibliographic records which make up its reading list. This second type of BIB record (a connection BIB) is an internal ALEPH structure; in addition to the default fixed fields, it contains only a CNO field (link to course) and an SID field (link to bibliographic record). It is system-created when a title is added to a Course Reading list.

The Course Reading setup requires a separate database (library) parallel to the Administrative library. In the ALEPH demo setup, this is the USM30 library. In a multi-ADM setup, each institution (ADM) that uses Course Reading has a parallel Course Reading library (e.g. USM51/USM31).

True bibliographic metadata records should never contain course links. The links should always be held in the separate connection record. The bibliographic records for the Course Reading and Reserves Collection can be cataloged in the library's main database (in the ALEPH demo setup this is USM01), or they can be cataloged in the Course Reading Database. Using records already cataloged in the regular catalog saves cataloging effort; cataloging in the Course Reading database keeps the main database free of extraneous records (such as photocopies, scanned documents and private material).

The Union View mechanism is applied to the Course Reading database so that titles that are listed for more than one course will be virtually merged into a single record for view in the OPAC. Therefore, records that are linked to more than one course will display as a single de-duplicated record, with the course information from all the de-duped records merged in the single record display.

2 Setup

There are two preliminary tasks for the System Librarian in the setup of Course Reading:

- Creating the Course Reading library (typically as XXX30).
- Linking the new Course Reading library to the Administrative library (in library_relation)

There are two additional initial tasks, not necessarily performed by the System Librarian:

- Creating users, or adding Course Reading privileges to existing users.
- Creating a borrower (patron) record for each Course Reserves Unit, in order to enable placing hold requests for pick lists.

2.1 Creating the Course Reading Library

This library should be based on the ALEPH demonstration Course Reading library called USM30. The document, *How to Open a New Library*, describes the creation of a new library.

2.1.1 Application of Union View

In addition to standard BIB library setup, the Course Reading Library can be set up as a Union View library. Refer to Section 10 Union View.

2.2 Linking the Course Reading Library to the ADM library

When the library has been created, in addition to the standard definitions for BIB, HOL, and ADM in the `./alephe/tab/library_relation` configuration table, the library must also be defined as a Course Reading library as in the following line:

```
CRS USM50 USM30
```

In a multi ADM-library environment, the course reading library will only show items of the related ADM library.

2.3 Staff User Privileges

Refer to Section 2 of the *ALEPH 19.01 Staff User Guide - Course Reading and Reserves* for information on staff permissions that are required for working with Course Reading.

2.4 Patron Record for Each Course Reserves Unit

Hold requests can be placed on items intended for transfer to the Reserves collection. The operator must enter a user ID to which the request will be registered. This user is a standard patron, created specifically for this purpose. The creation of a patron record for a Course Reserves Collection is carried out via the Circulation GUI.

Note

You should assign a status that is not limited for the number of requests and which has privileges for multiple-hold requests.

3 Course Records

A course is identified for the operator by the **course number**. In cases where the course number is not unique, additional information, such as instructor or course dates, is required in order that the operator be able to identify the course.

Internally for the system, a course is unique and identified by the course number to which an internal course sequence number is appended. The sequence number serves to differentiate between different courses that use the same course number.

There are two types of Course Records:

- Common course records

  This type of course is optional. It provides efficient management of reading lists for multi-session courses, enabling one-time registration of reading material for all sessions of a course. It is suitable for use when there is a list of reading material that is used in common by all the instructors of all the sessions of a course. The advantage in using this type of course is that although the material is listed only once, it is included in search and display for all sessions of the course. The common course record does not have instructor or active dates. In addition to material registered to a common list, the existence of a common course does not preclude the option of adding specific reading materials to specific sessions of the course.

- Individual course records

  This is the actual classroom course, and must have an instructor and active dates. An individual course is linked to a common course by virtue of the fact that they have the same course number. Since most courses at a college/university are taught in a single session only, most course records will not be linked to a common course.

  If an individual course is linked to a common course, the reading list for the course includes both the individual course's reading list and all the common course's reading list, displayed in one sequence. Entries from the common course are displayed and indexed for each of its individual courses. The individual course can have additional entries of its own.

  If the course does not have the same course number as a "common" course, it will have only the reading list entries that have been registered for it.

Special Note for Course records that were upgraded from ALEPH WEB Course Reading system to GUI-Circulation Course Reading system:

If Web Course records have no value in the 'Instructor Name' field, the Course Upgrade routine sets the Instructor Name field to INSTRUCTOR.

After the upgrade is completed, you may open the GUI-CIRC-Course List window, retrieve all courses by instructor name, INSTRUCTOR, and update the Instructor Name field with the relevant value.

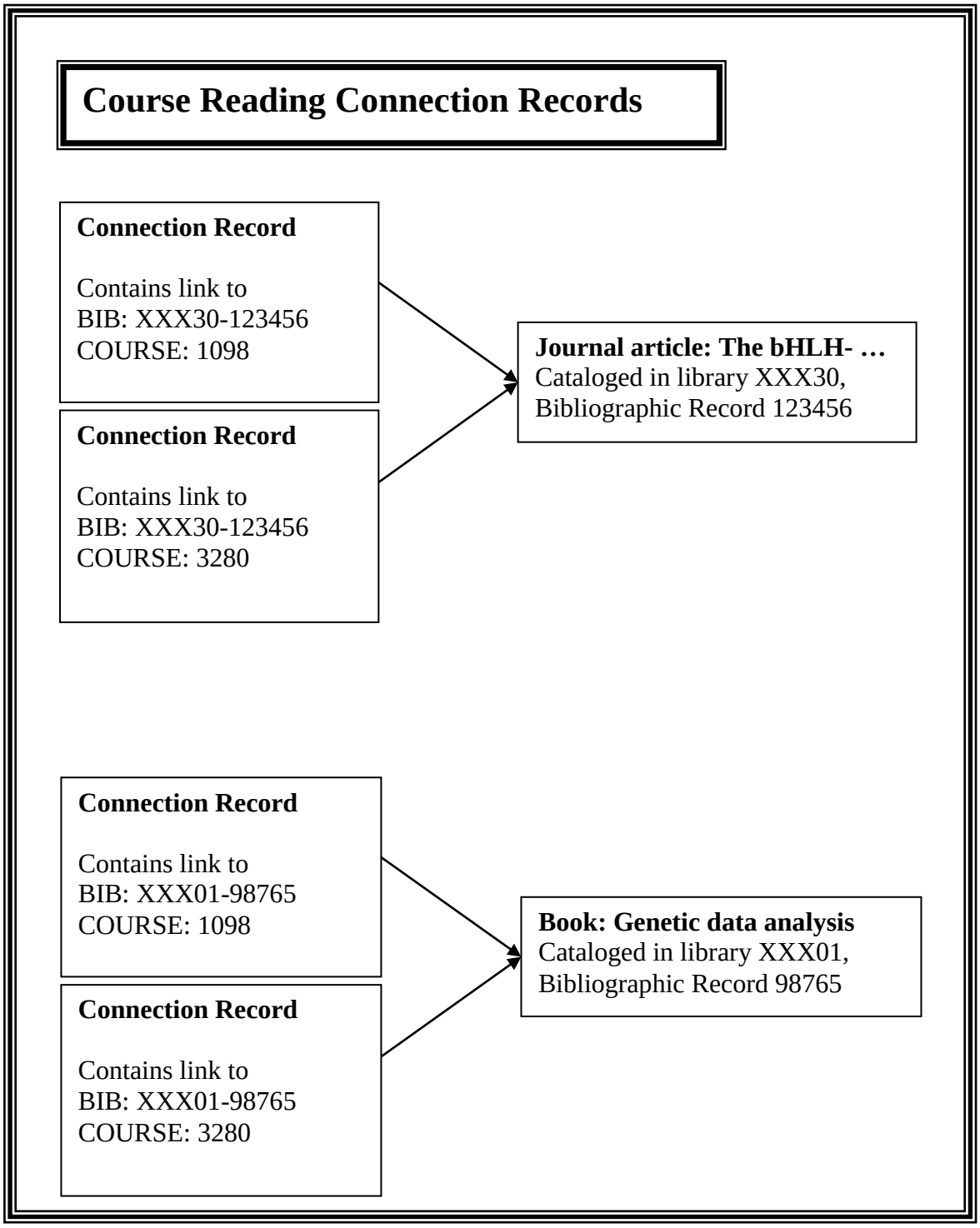
4 Reading List Records

The Reading List record is a BIB record that contains links to a course and to a bibliographic record in the same or another library's catalog. This record includes:

- Default fixed fields
- CNO field (link to Z108 Course record)
- SID field (link to bibliographic record)
- Up to four note fields, specific to the bibliographic record within the context of the reading list (e.g. pages or chapters that are relevant)

When the Reading List record is displayed (in the WEB OPAC, in the Course Reading GUI client, or in the Search tab of the GUI client), the bibliographic data and the course data are expanded into the Reading List record.

The following diagram illustrates the relationship of the records and their links in the Course Reading library.



4.1 Linking fields

The CNO and SID are linking fields that are used by expand-doc-course and expand_doc_merge_union_crs to add Course data and Bibliographic data to the Course Reading linking record. Course data is added using the CNO link, and bibliographic data is added using the SID link. These expand procedures build virtual fields for course name, number, instructor, etc., and full bibliographic information.

4.2 expand_doc_course

This expand program adds all the bibliographic data (retaining bibliographic field tags) and the following course information in "CNO…" fields; the parts of the CNO fields are separated by hyphens, as shown.

These fields can be used for indexing and for display in the OPAC. Refer to Section 6 Indexing, to Section 8 Configuration Information, and to the USM30 setup of tab11_word, tab11_acc, tab00.eng and edit_field.eng for examples of their use.

CNO-0	$$a Z108-COURSE-NUMBER $$b Z108-COURSE-NAME $$c Z108-INSTRUCTOR-NAME $$*n* (repeated)*Z108-PERIOD $$e Z108-DEPARTMENT
CNO-1	$$a Z108-COURSE-NAME
CN0-2	$$a Z108-COURSE-NAME – Z108-INSTRUCTOR-NAME
CNO-3	$$a Z108-INSTRUCTOR-NAME
CNO-4	$$a Z108-INSTRUCTOR-NAME – Z108-COURSE-NAME
CNO-5	$$a Z108-DEPARTMENT
CNO-6	$$*n* (repeated)*Z108-PERIOD
CNO-7	$$a Z108-UNIT
CNO-8	$$a Z108-COURSE-NUMBER – Z108-INSTRUCTOR-NAME
CNO-9	$$a Z108-INSTRUCTOR-NAME – Z108-COURSE-NUMBER
CNO10	$$a Z108-COURSE-NUMBER – Z108-COURSE-NAME – Z108-INSTRUCTOR-NAME
CNON1	$$a NT1
CNON2	$$a NT2
CNON3	$$a NT3
CNON4	$$a NT4

CNO-6 is the period during which the course takes place. For example: Winter, Spring, and so on. The values are stored in the CNO-6 field in the following format: $$n<period>. *n* is a number from 1 to 7. For example:$$1Spring. If the course is valid for more than one period, the field is filled with a series of values, each value in a separate subfield, with no spaces in between. For example:$$1Spring$$2Summer.

Note that if the record is connected to a common list, the expanded information will be added for each section of the course.

In addition, subfield $$8 is used in the course information, in order to group together all the information of one course.

4.3 Grouping of Expanded Fields

The Course Reading public access Web interface is based on Union View architecture, with virtual de-duplication and merged display of records that are detected as being the same in order that bibliographic records linked to more than one course will display as a single de-duplicated record. In order to retain course information grouped by course, a sequence number is added to each CNO field created by the program.

CNO-0 $8 <sequence>

$a <course number>
$b <course name>
$c <course instructor>
$d <period>
$e <unit>

CNO-1 $8 <sequence> $a <course name>

CNO-2 $8 <sequence> $a <course name> - <course instructor>

etc.

4.4 expand_doc_merge_union_crs

This expand procedure is required for the application of Union View to the Course Reading database for OPAC. This procedure adds bibliographic and course data to the Course List (connection) record. It adds all the bibliographic data (using the SID link) and the course information of ALL the courses/sections that have this same SID (i.e. are linked to the same bibliographic record). This is done using the Union View Z120 table, which groups together all documents that share a common SID. To display only active courses, add the "ACTIVE" parameter in col 3 of the tab_expand. (For more details regards the union view setup, see section 10 Union View.

5 Course Period and Logical Bases

5.1 Period

In USM30 Course periods have been set for Winter, Spring, Summer, or Not Active. Different or additional periods can be defined in the COURSE-PERIOD section of the ADM library's ./tab/pc_tab_exp_fields.<lng>.

The **Period** field is indexed in the "wpe" word file in USM30.

5.2 Logical Bases

5.2.1 Period

You can create a logical base for the course period and enable the user to search in the Web OPAC interface for a course according to a period.

In order to enable limiting a search according to the time period, you must define logical bases in the `tab_base` table based on "wpe". Note, however, that if you use common lists you cannot use "alldocuments not …" for the period. For example, "alldocuments not na".will not function as needed, and you must exclude a group by not including it in the FIND definition (e.g. wpe=(winter or spring or summer)). If your library uses the "suppressed" option for records in a Course Reading list, remember to define "suppressed" as part of the FIND command in `tab_base`.

In the following example:

- USM30 displays all records; this should be used for the Search in GUI.
- U-USM30_1 is union view display, limited to records listed for courses that have period "winter", and not including suppressed or deleted records
- U-USM30_2 is union view display, limited to records listed for courses that have period "spring", and not including suppressed or deleted records
- U-USM30_3 is union view display, limited to records listed for courses that have period "summer", and not including suppressed or deleted records

```
USM30                   Course Reading - Complete              USM30 N

U-USM30_1               Course – U.V.Winter  USM30             USM30 N
wpe=winter not wst=(suppressed or deleted)
U-USM30_2               Course – U.V.Spring  USM30             USM30 N
wpe=spring not wst=(suppressed or deleted)
U-USM30_3               Course – U.V.Summer  USM30             USM30 N
wpe=summer not wst=(suppressed or deleted)
```

6 Indexing

Indexing in the CR library (XXX30) is done in the same manner as in a BIB library. For this purpose the following library's tables should be set: tab11_word, tab11_ind, tab11_acc, tab00.lng, etc.

Note that after editing any of these tables, the XXX30 database should be re-indexed: Indexing of the Course Reading database is carried out via the Cataloging GUI, from where the Indexing services are run.

In addition, the ue_01 procedure must be consistently run in order to perform on-going indexing.

Note: To allow searching and browsing of CR records via WEB OPAC; the OPAC HTML files must quote the relevant index codes.

6.1 Browse Indexing (tab11_acc)

In addition to the bibliographic indexes, the course information can also be indexed. Following is the full list of course indexes:

Index Name	Field Code	Subfield Code	Index Code
Course Number	CNO-0	a	CNO
Course Name	CNO-0	b	COU
Department	CNO-0	e	DEP
Instructor	CNO-0	c	INS
Course Number + Instructor	CNO-2	a	COUIN
Instructor + Course Name	CNO-4	a	INSCO
Period	CNO-6		PER
Course Name + Instructor	CNO-8	a	CONIN
Instructor + Course Number	CNO-9	a	INSCN
Course Number + Course Name + Instructor	CNO10	a	CONNI

This is maximal setup. You might not want to use all these indexes in your library.

6.2 Search Indexing (tab11_word)

The Word Indexing of the Course Reading database can use bibliographic data (as words from titles, authors, subjects, etc.) as well as Course information such as words from course number, course name, department, instructor and period.

The STA (status) field is indexed in “wst”. The "wst" index is used for excluding suppressed or deleted records from the OPAC view.

6.3 Direct Indexing (tab11_ind)

Direct indexing has been set up for purposes of internal structure and staff functions, using SID, CNO and CNO-S.
The following lines must exist in your XXX30/tab11_ind:

```
SID                    SID    a
CNO                    CNO-S  a
CNO-0                  CNO    a
```

Note:
The CNO, CNO-S, SID indexes must not be renamed.

7 E-Reserves

7.1 Introduction to E-reserves

The home of e-Reserves (Electronic Reserves) in ALEPH 500 is the Course Reading library (for example, USM30). e-Reserves uses the standard Course Reading tools and workflow. The e-Reserve title is linked to a Course and cataloged either in Course Reading or in the BIB library.

E-Reserves are items which are available electronically via the library's Web OPAC. The Web OPAC is used by students and professors to perform such tasks as searching (by course, instructor name, and so on) and displaying and activating files.

The purpose of e-Reserves in ALEPH is to regulate access to items available via external links (856) based on course enrollments. Files can be in any electronic format.

Here is a specific example: e-Reserves must allow Student A to access File B which is linked to Course C - provided that Student A is enrolled in Course C. Conversely, e-Reserves must block Student A from accessing File B which is linked to Course C - provided that Student A is not enrolled in Course C.

E-Reserves functionality is based on extended services related to the subfield u of the 856 (Electronic Location and Access) field in the MARC21 standard. Subfield u defines the URL for locating an item using existing Internet protocols.

7.2 Copyright and Use Permission Form

Copyright law governs the right to prepare derivative works, to reproduce a work or portions of it, and to display or perform a work in public. The Copyright mechanism in the e-Reserves module lets a library regulate access in terms of user permissions.

The e-Reserves librarian is responsible for setting up copyright and usage permissions for each e-Reserves item. These permissions are set up in the Copyright form in the Course Reading interface or directly input in the Bibliographic record in the Cataloging module. ALEPH proprietary use of subfield $$4 in the 856 field is used to store the copyright and usage permissions defined.

7.3 Usage Permission

A patron using the ALEPH OPAC must have "Link Permission" in his profile in order to view an object. In addition, there are three levels of usage permission that can limit viewing an object:

- **Open use** - When a Web OPAC user clicks the link in the bibliographic record, he receives the object immediately without having to sign a copyright agreement.
- **Open use with copyright statement** - When the Web OPAC user activates the link in the bibliographic record, a copyright notice appears. The user must sign this copyright agreement in order to receive the object.
- **Use is allowed to patrons who are registered in the course** - Only patrons who are registered in the course to which the bibliographic record is related are given access to the object. The system recognizes patrons as being registered in the course after they have signed in.

If a patron does not meet the usage conditions set for accessing an object, e-Reserves blocks the patron from accessing the file.

For example, if a student fails to sign in when searching the Course Reading OPAC, a message displays if he attempts to view a file associated with a course.

There are three messages associated with usage permission blocks. The messages are taken from $aleph_root/error_lng/media_heading:

```
1006 0000 L Object is only permitted for course(s) $1
1007 0000 L This link is limited to students registered in
specific courses. Please view it through the Course Reading
Catalog.
```
[this message displays when the bibliographic record is displayed in the regular OPAC, and not through the Course Reading OPAC]
```
1008 0000 L Object is only permitted for courses, but no course
is linked.
```

Conversely, if all of these conditions are met, e-Reserves lets the patron access the file (if necessary, after he reads the Copyright statement and clicks **OK**).

7.3.1 Usage Permission Configuration

The following lines should exist in your XXX30/tab/ tab_buf_z403:

```
########## get_buf_z403_course          856##, ,y,*,Y,
########## get_buf_z403_course          856##, ,y,-,U,
```

7.3.2 Setting up Usage Permissions

To set up usage permissions in e-Reserves:

Usage permissions in e-Reserves are handled using subfield 4 (a special ALEPH subfield) in the 856 (Electronic Location and Access) field. The subfield 4 information can be entered in the bibliographic record in the Cataloging GUI, or through the Copyright form in the Course Reading module.

There are three options for usage permissions:

- **Open use** - $$4 = Y
  No copyright message appears.

- **Open use with copyright statement** - $$4 = blank or non-existent
  A copyright agreement appears and must be signed in order for the user to be able to access the resource.

- **Use limited to users who are registered in a given course** - $$4 = E
  Usage includes option Y above and viewing resources is allowed only for users who are registered in the course. Course registration is regulated by the Z107 (e-Reserves Course Registration) table.

  The Z107 record contains the patron's ID and the course number of the course in which the student is enrolled. If a student who is not enrolled in a specific course attempts to view a file associated with that course, a message will display.

There are additional subfields in the 856 field which contain information about the object's provider and related charges:

$$V - vendor/provider code
$$F - type of fee
$$C - fee amount

These subfields are optional and for information purposes only.

The data for the Z107 table can be loaded from a flat file. To do this, you use the Import Database Tables - Without Checks (file-04) service which is accessed from the Manage Database Tables submenu of the Services menu.

You import the files into the ADM (for example, USM50) library. In the **Table to Import** field, enter the file name `z107`. After the service has been run, the files can be found in the `xxx50/files` directory.

The flat file structure (for p_file_04) is:

Z107-ID. Positions 1-12 only. This is the user's identification key. The relevant ID number must match a valid Z303-ID).
Z107-COURSE-NUMBER. Positions 13-30. This course number must match a valid Z108-COURSE-NUMBER.

For example:

```
User ID     Course Number
1-12        13-30
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
123101      100-12
123103      100-12
123105      100-12
123106      100-13
123108      100-13
123110      100-14
123113      100-14
123115      100-14
```

8 Configuration Information

Indexing and Doc display configuration is set in the Course Reading (XXX30) library tables. All other configuration is set in the ADM (XXX50) library tables, or in ./alephe/.

8.1 tab_expand

As explained in Section 4 Reading List Records, the Reading List records do not contain any course or bibliographic information, they only contain links. The full record in the Reading List is created on the fly by expanding it with the Bibliographic and Course information. Refer to the `./usm30/tab/tab_expand` table for setup.

The `expand_doc_crs_bib` program can also be set in `tab_expand`. This program is useful in case the BIB record is expanded from another BIB record and the course record should have information from both a related BIB record and its expanded document. When setting `expand_doc_crs_bib`, define in col. 3 of tab_expand the fields that will be expanded to the course document.

Sample `tab_expand` for MAB library:

```
CRS-BIB    expand_doc_mab
CREATE-Z13 expand_doc_course
CREATE-Z13 expand_doc_crs_bib              331-2,1##-2
```

In addition to expanding Bibliographic record fields into the Reading List record, the expand_doc_course routine creates the fields from the Course record and adds them to each connection BIB. See Section 4.2 expand_doc_course for details.

8.2 Hold Request Start Date for Pick List

The "request date" of a hold request created from Course Reading is set to default to *nn* days before the starting date of the course. This *nn* number of days is set in the `course_request_days` line in the `./alephe/pc_server_defaults` file.

If there is no value defined in the course_request_days variable, the number 14 is used as a default. (The hold request date is 14 days before the Course's open date).

This enables the library to create requests for Course Reserve material before the course is actually active.

Requests placed via the Course Reading module are automatically defined as Rush Recall type Hold Requests, the Z37-CATALOGER-NAME is set to “CRSE-READ”, and the course number is registered in Z37-NOTE-1.

8.3 Submitting a List for Course Reading from OPAC

Patrons who have "Course Permission" in their profile can submit a list items that they want to put on Course Reading. This action, carried out in the Web OPAC from "My e-Shelf", sends an email message and saves the list on the server.

8.3.1 E-mail Address

You can choose to specify the e-mail address of the Course Reading administration inside the HTML file: `myshelf-course` (or myshelf-course-<base name>).

You can also use this option to build a drop-down menu of e-mail addresses where there are several course reading administrations to choose from. For example, if the following definition is present in the SEND_MAIL setting in the `myshelf-course` HTML file:

```
<input type=checkbox name="SEND_MAIL" $$0200-C"on" value="
mdewey@exlibris-usa.com ">
```

The system sends the e-mail notification message to mdewey@exlibris-usa.com:

```
If there is no e-mail address defined in the myshelf-course HTML
page, the system uses the e-mail address defined in the
www_course_email_address environment variable present in
./alephe/www_server.conf.

For example:

setenv www_course_email_address     thechoice@exlibris.co.il
```

8.3.2 Directory for Saved List

You can define the directory on the server in which the items will be saved. The directory is defined in `./alephe/aleph_start`.

In the following example from `aleph_start`, the items will be saved in a file in `./tmp/save_cr`:

```
setenv    TMPDIR            ${ALEPH_MOUNT}/a${ALEPH_VERSION
}_$ALEPH_COPY/tmp
setenv    FILE_TMPDIR       $TMPDIR
setenv    SAVE_FILE_TMPDIR  ${ALEPH_MOUNT}/a16_1/tmp/save_cr
```

If a directory is not set in `aleph_start`, the items will be saved in a file in `alephe_scratch`.

8.4 Linking

The `./xxx30/tab/tab_z103` table must be set up for Course Reading.
The following lines should exist:

```
update_z103_lkr
update_z103_course
```

8.5 Course Reading GUI Configuration

8.5.1 Display Course Information

Setup for the HTML display of Course Information in the Course Display tab in the lower pane of the List of Courses in the CIRC GUI is set in the ADM library:

```
./pc_display_eng /course-details
./pc_display_eng /course-details.trn
```

8.5.2 Quick Cataloging

The configuration for the "quick cataloging" input form is in the ADM library:

```
./tab/pc_tab_crs_fast_cat.eng
```

8.5.3 Search Form for Course Reading

The configuration for the search form (searching for a bibliographic record in order to add it to a Course Reading List) is defined in the ADM library:

```
./tab/pc_tab_find_course.eng
```

8.5.4 Display Requester IDs

On the Items tab in the lower pane of a Reading List, there is a column which displays the IDs of patron who have hold requests for the item.

The number of patron IDs that displays is defined in the ADM library's tab100 in NUM-REQUESTERS-ID.

For example:

```
NUM-REQUESTERS-ID=10
```

Variable can be between 01 and 99.

If the number of requests exceeds the value in `tab100`, the requests will still be registered, but not displayed in the Requesters column.

8.5.5 pc_tab_col.<lng>

The following sections of the ADM library's ./tab/pc_tab_col.<lng> control GUI columns display:

- PC_COURSE_LIST – List of Courses
- PC_COURSE_BIB_LIST – Course's Doc List

- PC_COURSE_FILE_LIST – List of Doc in Load file (Load button at the Course's Doc List window)

- PC_COURSE_ITEM – List of items (List of Doc, lower pane-Items)

- PC_OTHER_COURSE_LIST – Other courses to which the doc is assigned (List of Doc, lower pane-Attached to Courses)

8.5.6 pc_tab_exp_field.<lng>

The following sections of the ADM library's ./tab/pc_tab_exp_field.<lng> control dropdown lists:

- COURSE-DEPARTMENT

- COURSE-UNIT

- COURSE-PERIOD

- COURSE-COPYRIGHT

- COURSE-FEE

- COURSE-DOC-SORT-TYPE - controls sorting of the reading list. These values should match the values in xxx30/tab_sort.

8.5.7 View ADAM objects in the Course Reading OPAC

To view an ADAM object that is linked to a document from the main catalog (e.g: USM01) in the Course Reading OPAC, the following line should exist in the Course Reading library's ./tab/tab_buf_z403:

```
########## get_buf_z403_media_course      VIEW
```

In addition, www_tab_short and edit_doc_999 should contain indication about the display of the digital objects (MED, EXTL, TN).

9 GUI Services Templates

Following is a list of the Course reading batch report templates.

Template	Purpose
course-report Course Summary Report	Generated by course-06
course-summary Course Lists Report	Generated by course-01

Template	Purpose
course-pickup-list Course Pick Up List	Generated by course-02
course-inactive Course Not Active	Generated by course-05
course-reserve-item Remove items from Reserves Collection	Generated by course-03
course-reading-not-linked Report of Course Reading Records that are not Linked to a Course	Generated by course-04

10 Union View

The Union View mechanism is applied to the Course Reading database so that titles that are listed for more than one course will be virtually merged into a single record for view in the OPAC. Therefore, records that are linked to more than one course will display as a single de-duplicated record, with the course information from all the de-duped records merged in the single record display.

10.1 expand_doc_merge_union_crs

This expand procedure is required for the application of Union View to the Course Reading database for OPAC. This procedure adds bibliographic and course data to the Course List (connection) record. It adds all the bibliographic data (using the SID link) and the course information of ALL the courses/sections that have this same SID (i.e. are linked to the same bibliographic record). This is done using the Union View Z120 table, which groups together all documents that share a common SID.

For relevant tab_expand setup, see Section 10.2.5 tab_expand.

10.2 Configuration for Union View

The following configuration must be set for Union View, in the Course Reading (XXX30) library:

10.2.1 tab100

UNION-LIBRARY=1
UNION-IGNORE-MATCH=SUPPRESSED

10.2.2 tab_base

The Course Reading database must be defined as a union database in `./alephe/tab/tab_base.eng`:

U-<CRS library> Course - All USM30 <CRS library> N

In order to be identify as a Union View, the CR library code must follows the letter "U-" as in the following example of setting union base for USM30 :

U-USM30 Course - All USM30 USM30 N alldocuments not wst=(suppressed or deleted)

10.2.3 union_global_param

The CR library (XXX30) must be present in the `./alephe/tab/union_global_param` table:

Example for USM30:

USM30 B candidate_prog union_candidate_course 100
USM30 B match_prog union_match_course
USM30 B preferred_prog union_preferred_course
USM30 B normalize_prog union_normalize_cdl

10.2.4 tab_merge_union

For correct display in the OPAC Course Reading interface, which uses Union View mechanisms, the xxx30 library must have correct setup of `./tab/tab_merge_union`. The correct setup for the course union view is as follows, and should not be changed:

```
30 1 Y #####
30 1 N CNO##
30 1 N SID##
30 1 N NT###
30 2 Y CNO##
30 2 Y NT###
```

10.2.5 tab_expand

Set the relevant sections in ./xxx30/tab/tab_expand with expand_doc_merge_union routine.
For example:

WEB-FULL expand_doc_merge_union_crs
WEB-FULL-1 expand_doc_merge_union_crs
WEB-BRIEF expand_doc_merge_union_crs

The following line must exist:
PRE-MERGE expand_doc_course

Refer to the demo set up: `./usm30/tab/tab_expand` table for setup.

Note:

If you use PST for displaying the location, then the WEB-FULL expand_doc_merge_union_crs and WEB-BRIEF expand_doc_merge_union_crs should be placed **before** the expand_doc_bib_loc_n_x expands.

10.2.6 www_tab_short.<lng>

For the Brief view (in table format) for Course Reading, the following fields have received special handling in order to see all the values (and not only the first one as Brief view usually does):

CNO-2 (Z108-COURSE-NAME – Z108-INSTRUCTOR-NAME)
CNO-4 (Z108-INSTRUCTOR-NAME – Z108-COURSE-NAME)
CNO-8 (Z108-COURSE-NUMBER – Z108-INSTRUCTOR-NAME)
CNO-9 (Z108-INSTRUCTOR-NAME – Z108-COURSE-NUMBER)
CNO10 (Z108-COURSE-NUMBER – Z108-COURSE-NAME – Z108-INSTRUCTOR-NAME)

Choose one of the above CNO fields and define it in ./xxx30/tab/www_tab_short.lng for display in the "Course/ Instructor" column.

For example, to display of CNO-2, set the following line:

```
4 L Course/Instructor    02---A01---A 00 00 0100 S          ## CNO-2
```

10.2.7 edit_doc_999.<lng>

The fields displayed in Full view are defined in ./xxx30/tab/edit_doc_999.eng. Because of the Union view mechanism, one record can contain information from many courses. This happens if the bibliographic record is listed for more than one course.

To display all the fields from each course as a group, subfield $$8 has been added to define the “grouping”.

For example, edit_doc_999.lng should contain:

```
## CNO-2 8 1            D LCourse              Y Z      E        W
## CNO-6 8 1            D                      Y        E        W
## CNON1 8 1            D                      Y        E        W
## CNON2 8 1            D                      Y        E        W
## CNON3 8 1            D                      Y        E        W
## CNON4 8 1            D                      Y        E        W
## CNO-2 8 2            D LCourse              Y Z      E        W
## CNO-6 8 2            D                      Y        E        W
## CNON1 8 2            D                      Y        E        W
## CNON2 8 2            D                      Y        E        W
## CNON3 8 2            D                      Y        E        W
## CNON4 8 2            D                      Y        E        W
## CNO-2 8 3            D LCourse              Y Z      E        W
## CNO-6 8 3            D                      Y        E        W
## CNON1 8 3            D                      Y        E        W
## CNON2 8 3            D                      Y        E        W
## CNON3 8 3            D                      Y        E        W
## CNON4 8 3            D                      Y        E        W
```

10.2.8 Union view batches

The batch service union_04 (update z120/z127 records) should be running on regular basis. It is recommended that it be added to ./alephe/tab/job_list, to run every two hours. For example:

```
W2 10:00:00 Y                  USM30 p_union_04                  USM30
W2 13:00:00 Y                  USM30 p_union_04                  USM30
W2 15:00:00 Y                  USM30 p_union_04                  USM30
W2 18:00:00 Y                  USM30 p_union_04                  USM30
```

10.3 Union View Indexing

If you want to re-build an existing CR database, after all the indexes are built, run p_union_01 (build empty z120 records) and p_union_02 (creates z120 records) once. Then create the z127 records. It is necessary to run the process to create the Z127 records after running p_union_02 (populating empty equivalencies records table - z120). The process to create the z127 records is run from the command line and does not have any parameters. The syntax for running it is: csh –f $aleph_proc/create_z127.
Note that prior to running this process, it is necessary to run the Drop & Create Table and Index utility for z127 by running util/a/17/1.

Rebuild Record Equivalencies (p_union_04) should be run on a regular basis to continually update the z120/z127 records. Note that z127 triggers need to be activated.

For more information regarding the Union View mechanism, refer to *How to Set Up and Configure Union View for Your OPAC*.

System Librarian's Guide - General

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2016. All rights reserved.

Document released: November 6, 2016

Web address: http://www.exlibrisgroup.com

Table of Contents

1 CLIENT SETUP INI FILES 5

2 ALEPHCOM.INI SETTINGS 5

3 FONT DEFINITIONS (FONT.INI) 24

4 WORKING WITH ALEPH SERVICES 25

4.1 Batch Service Sub-Library and Order Unit Sensitivity 26

5 SETUP FOR EXPANDED INFORMATION DISPLAY 27

5.1 The HTML Template 29

5.2 style-css 32

5.3 Transformation 34

6 SETTING UP USER LOGONS 35

7 SETTING UP THE DATE FORMAT 36

8 SETTING UP THE TIME FORMAT 38

9 SETTING UP NAVIGATION AND OVERVIEW TREES 38

10 SETTING UP STATISTIC ANALYSIS OF SERVER USE 39

11 SETTING UP LOCK PERIODS 39

12 SETTING UP FORM INFORMATION DISPLAY 39

13 SETTING UP TIMEOUTS FOR CLIENT'S LINK TO SERVER 39

14 SETTING UP A TRANSACTIONS LOG 40

15 SYSTEM ERRORS AND MESSAGES 40

16 BATCH QUEUE 40

17 SETTING UP STAFF PRIVILEGES 41

18 SETTING THE ONLINE CHECK BOX IN SERVICES 41

19 SETTING UP A CHECK FOR UTF-8 SUPPORT FOR WEB BROWSER....42

20 SETTING UP EXTERNAL SERVICES....42

20.1 ALEPH – SFX Integration....42

21 TOMCAT SERVER....43

22 GLOBAL SYSTEM STATUS....43

23 CUSTOMIZING SHORTCUT KEY ASSIGNMENTS....45

24 SETTING UP COLUMN HEADINGS IN GUI LISTS....47

24.1 GUI Lists - Addition of Columns Missing from pc_tab_col.lng....49

25 CUSTOMIZING SHORTCUT BUTTONS....49

26 DISPLAYING THE "LINK" FIELD IN THE CALLOUT BUBBLE IN GUI LISTS....50

27 SETTING UP GUI TOOLBARS....52

27.1 Display Text....52

27.2 Editing Toolbar Tooltips....53

27.3 Identifying Icons....53

28 CASH AMOUNTS - SUPPRESS THE DISPLAY OF DECIMAL POINT AND THE DIGITS AFTERWARDS....53

28.1 Suppress Decimal in GUI and OPAC Screens – Set Up and Functionality 54

28.1.1 Setting Up - tab100: SUPPRESS-DECIMAL-DISP:....54

28.1.2 Functionality - 'Suppress Decimal' in GUI and OPAC Screen....55

28.2 Suppress Decimal in Aleph Printouts - Set Up and Functionality....58

28.2.1 Setting Up - Translate Routines for Aleph Printouts....58

28.2.2 Functionality - Suppress Decimal in Data Printouts....61

29 PRESERVATION OF THE WWW_SERVER AND PC_SERVER LOG FILES....63

30 CLEARING THE VIR01 LIBRARY....63

1 Client Setup INI Files

In ALEPH, there are a number of files with an `ini` extension that contain settings which you can modify to control the appearance and behavior of GUI clients (PC). We refer to these files as `<module>.ini` files (`alephcom.ini`, `acq.ini`, and so on). They include settings that are mostly controlled from the actual `<module>.ini` file, and not from changes within GUIs. These settings include elements such as font sizes, colors, and the like, as well as module-dependent defaults.

Ini files are located in the tab directories in your ALEPH installation on your PC's local hard drive. For example, the ini file for the Acquisitions/Serials module on your PC might be located at `C:\AL500\ACQ\TAB` (where *C:* is your local hard drive, *AL500* is the directory of your *ALEPH* installation, *ACQ* is the module's directory, and *TAB* is where modular tables are stored). Note that there are two directories under the TAB directory that are considered masters; both include original files from Ex Libris. The xxx directory includes language files (*.dat) that are used for creating new interface languages and updating the synchronization of current languages. The xxx_tab directory is for future synchronization (used for re-installing the GUI).

The `alephcom.ini` file is comprised of setting definitions that are shared by two or more modules. Other `<module>.ini` files (`acq.ini`, `cat.ini`, `circ.ini`, and so on) contain settings that relate to a specific GUI client (Acquisitions/Serial, Cataloging, and so on).

Other GUI definitions are included in a set of files called `guisys.ini` files which are present in each of the tab directories in your ALEPH installation. You cannot customize `guisys.ini` files. The definitions they include are updated automatically when changes are made from the GUI, such as the splitter's position and the last tab selected from the Main tab. The dialog box location specifications are also in the `guisys.ini` files.

The following section presents and explains various aspects of the `alephcom.ini` file. The following is a list of settings divided into those found in `alephcom.ini` and those which are found in multiple `ini` files.

2 ALEPHCOM.INI Settings

The following settings are included in this section:

ABaseWindow
ATabControl
[AutoInput]
BubbleMsg
Date_Format
Edit
ExpandField
External

FirstTab
General
ItemForm
ListBoxPrintout
Mail
Main
OverviewTree
Package
PlainTextPrintout
Print
PrintDaemon
PrintExecute
RfidMedia
SearchFind
SearchTree
Services
Splitter
TextListBox
TextOrientation
UserList
Vendor
VendorList
VersionControl
XslCreator

Click on one of the listed names to go directly to the explanation for that setting.

[Main]

```
[Main]
ActivateSplashWindow=N
ApplicationBgColor=190,240,215
Diagnostic=Y
AlwaysMaximized=N
Compression=3
Encryption=0
Version=22.01
CommunicationProtocol=0
DataBaseAutoSelect=Y
SavePassword=Y
DefaultPrintConfig=1
CtrlShiftReverseScreen=Y
ExternalILL=N
NeverClose=N
```

ActivateSplashWindow=N
The "Splash window" displays a logo of Ex Libris Ltd., the developer of the ALEPH system. The System Librarian decides whether or not the Splash window pops ups when the application is started - **Y** displays the Splash window and **N** suppresses it. The bitmap of this logo is located in `\bmp\splash .bmp` from the `\alephcom` directory.

ApplicationBgColor=190,240,215
This field sets the background color of the application.

Diagnostic=Y
This line is used for debugging purposes and should **not** be changed.

AlwaysMaximized=N
If AlwaysMaximized is set to Y, the minimize button is disabled. The default is N, that is, enable the use of the minimize button.

This switch is helpful if some of your library's PCs are ALEPH-dedicated machines.

Compression=3
Compression is used when accessing server (the server must support suppression) values: "Compression type" or "N". Do not change this parameter.

Encryption=0
This field sets the type of communication encryption between the Aleph GUI client and the pc_server.

Version=22.01
This field sets the type of the communication encryption between the Aleph GUI client and the pc_server.

Note: The **Encryption** and **Version** fields must be configured also in the server tab_version table, in $alephe_tab.

CommunicationProtocol=0
Do not change this line.

DatabaseAutoSelect=Y
Determines whether or not ALEPH clients automatically log in to the first library listed in the Connect Library window. The order of the libraries is set in the `per_lib.ini` file or `base.dat` file.

SavePassword=Y
This is the default setting for the "Save Password" check box that appears on the ALEPH password window.

If the value "X" is used, the check box appears dimmed and there is no way to save the password. Use this option when you would like to disable the possibility of saving the GUI's password on the local drive.

CtrlShiftReverseScreen=Y
If the flag is set to *Y*, then while working on input fields where multiple languages can be inserted (such as the browse input string in the Browse windows), the Ctrl + Shift hot key can be used to reverse the position of the cursor and the text (left-to-right / right-to-left). In addition, if the flag was set to *Y*, then while working with a cataloging draft, Ctrl + Shift can be used to reverse the layout of the record (left-to-

right / right-to-left). This is useful when entering script that is written from right-to-left.

ExternalILL=N
This flag indicates which ILL mode your ALEPH system uses. The values are N (NO, default value) or Y (Yes):

- N = your system uses the GUI ILL mode and the ILL is accessible and managed by the GUI.
- Y = your system uses the Web ILL mode (external ILL).This blocks the link to the ILL GUI module and disables the **Run ILL** command from the GUI menu bar/ALEPH/Switch Application.

NeverClose=Y
Setting this variable to Y disables closing of the GUI with the ALEPH\Exit menu.

[General]

```
[General]
Targets=printer01,printer02,photoslip,photowait,photofill
BibInfoFormat=999
SummaryBibInfoFormat=-1
```

Targets=printer01,printer02,photoslip,photowait,photofill
Defines the list of print IDs available in the Print Daemon in the Task Manager utility.

[VersionControl]

```
Library=XXX01
Type=
RunAtStartup=
```

[Version Control]
Can be set to access different distribution directories based on the alephcom.ini file. Additional parameters control how the application communicates with the server:

Library=XXXX
The communication protocol requires this value. The default is USM01.

Type=xxxx
This is the directory name under the `pc_exe` directory. The directory's name is limited to 10 characters. If it is not set, then the system looks for `pc_exe./version`. For example, `pc_exe.16/version`

Together with the additional parameters, **[Version Control]** allows a site to set up multiple distribution points for various GUI client configurations on the same server (for example, Circulation, Cataloging, Acquisitions/Serials, and even per person).

RunAtStartup= Y/N

Controls whether the version control check will be run at start up. Selecting the *Run Version at Startup* option in the Update Version window automatically sets this value to Y. Clearing the *Run Version at Startup* option in the Update Version window automatically sets this value to N.

[Print]

```
[Print]
SaveHistoryNumberOfDays=1
DefaultPrintConfig=1
TempDir=c:\temp\Aleph
TempFileCounter=1063
```

[Print]
Every printout is saved in the GUI's history. The history is built for every module separately. For example, for the USM50 library, the history is built in `Alephcom\files\USM50\print...`

SaveHistoryNumberOfDays=1
This setting determines how many days the print history should be stored. Cleanup of old history files takes place when the GUI is started.

DefaultPrintConfig=1
Determines the default for the options on the printer icon on the bottom right of the GUI (0=normal,1=preview, 2=view raw XML, 3=browse XML).

TempDir=c:\temp\Aleph
This is the location where temporary files reside. The directory is cleaned up every time the ALEPH GUI application is loaded.

TempFileCounter=1063
Do not change this line.

[Mail]

```
MailServer=IL-EX01
FromAddress=library@exlibris.co.il
BccAddress=
MailerName=Exlibris Library Services
AuthMethod=
BccAddress=alephadmin@exlibris.com
DefaultSendMethod=
SecureConnection=SSL
```

MailServer=IL-EX01
The `MailServer` is the machine that receives and sends e-mail messages. This must be adjusted to suit your environment. You can add the port number where a non-standard port is used, using a <host>:<port> format. For example: 10.345.24.11:45

FromAddress=library@exlibris.co.il

`FromAddress` is the e-mail address from which all library letters to patrons and vendors are sent from the mail server. This must be adjusted to suit your environment.

MailerName=Exlibris Library Services
`MailerName` appears as the name of the sender in all library letters to patrons and vendors which are sent from the mail server. In the above example, "ALEPH 500 Administrator" appears as the sender in e-mail messages instead of the actual e-mail address, "webmaster@exlibris.co.il". If the recipient clicks the Reply button, the response is sent to the actual e-mail address (in our example, "webmaster@exlibris.co.il").

AuthMethod=
Enter the required authentication method. The supported authentication methods can be LOGIN or PLAIN. If the parameter is left blank then no authentication is method is used. Note that the user name and password can only be set using the Configuration\E-Mail Settings menu option in the AlephADM module.

BccAddress=alephadmin@exlibris.com
All e-mail messages that are sent to patrons and vendors will also be sent to the address that is specified in this variable. The address will be on the Bcc list of the sent e-mail message.

DefaultSendMethod= Possible values are H (attched file), P(plain), B(both attached file and plain), and L (HTML in mail body)
Determines the default email method (attached HTML, both, plain) when there no value comes from server, such in cases when the email was generated via the Web OPAC.

SecureConnection= Possible values are NONE (communication between Aleph and the SMTP server is secured at the basic level only), SSL, and TLS. This determines the security method for communication between Aleph and the SMTP server.

[UserList]

```
[UserList]

FilterType=0
```

FilterType=0
The Users list can be sorted according to one of the following Filter types:
Name(Filter type)=0
ID Number=1
Barcode=2
The `UserList FilterType` can also be changed manually using the **Sort by** option on the Users List in the Circulation client.

[ExpandField]

```
[ExpandField]
;
```

```
; Describe the Expand button Status of special fields :
; 1 - Regular (pc_tab_exp_field_extended.eng) (default)
; 2 - Non Visible
; 3 - Special (for location=close stack)
; 4 - Special (for location=location index list)
; 5 - Special (for location=852 form)
; 6 - Special (for location=852 form, including special method for
;     calculating Author Number and Chronology Number). Valid only
;     forItemLocation1Field and ItemLocation2Field.
;
ItemLocation1Field=4
ItemLocation2Field=5
SerialLocation1Field=5
SerialLocation2Field=5
```

This section defines the window that opens when the arrow next to the Call Number and 2nd Call Number fields that appear on the Item Form in the Cataloging and Acquisitions/Serials modules is clicked.

1 – Regular (pc_tab_exp_field) (default)
2 – Non Visible
3 – Opens a secondary window which creates a 2 part location, the first part being a size identifier and the second part being a counter (from UTIL G/2) plus standard text from `pc_tab_expand_field`.
4 – Opens a secondary window, which displays the list of call numbers in the system.
5 – Opens a form for entering call number in MARC21 852 field format.
6 – Opens 852 Special Form for calculating Author number and Chronology Number based on various methods. For example, for Latin material, Cutter Sanborn and Cutter Three Figure methods and for Korean material (Hangul and Hanja), Lee Jai Chul and Elrod methods. For more details, see the Item Call Number- Author Number and Chronology Number section of the *Aleph 20.01 System librarian's Guide – Items*.

```
ItemStatisticField=3
SerialItemStatisticField=3
ItemStatisticDelimiter=/
```

This section defines the window that opens when the arrow next to the Statistic field that appears on the second tab of the Item Form in the Cataloging and Acquisitions/Serials modules is clicked.

1 - No arrow on field.
2 - Opens pc_tab_expand_field menu.
3 - Opens special four-part form, each part of which can have a `pc_tab_expand_field` menu. The parts are separated by the delimiter defined in `ItemStatisticDelimiter`.

[TextListBox]

```
[TextListBox]
ColorC01=000,000,000
ColorC02=128,000,000
.../...
CopyFromList=Y
```

This section defines the colors for the various lists in the system. The color identifier, defined as C01, C02, and so on, refers to the color defined in the window's section in `pc_tab_col.lng`. on the server. The color value (red, green, and so on) is defined here, but can be individually changed on a specific client, using the Set Display Window.

CopyFromList=Y
If set to *Y*, the user can copy and paste data from list boxes in all GUI modules.

[PrinterDeviceList]

```
[PrinterDeviceList]
Device01=Digital DEClaser 3250
Device02=Compaq Laser Printer LN16
Device03=HP LaserJet 2100
```

This is a printer names record. This table translates real names into logical names. In the `print.ini` of each GUI module, you can define the print device number for each printout. This allows the user to divide printouts for different devices without online intervention.

[Username_Format]

```
[Username_Format]
FormatStyle=0
```

The settings chosen in the `pc_server_defaults` file are saved in this section.

[Date_Format]

```
[Date_Format]
DateSequence=dmy
; s or S is space in separator
HourSeparator=:
DateSeparator=/
```

[OrderIndex]

```
RefreshFilter=N
RefreshOnStart=Y
```

The `OrderIndex` section refers to various Order lists opened in the Acquisitions module from the Search Tab and Order Index opened from Item Form - Tab2. The Order Index is opened when the user clicks on the drop-down menu in the **Order No.** field.

RefreshFilter=N
This indicates whether or not the search is performed automatically after the filter has been changed by the user without clicking the **Refresh Filter** button.

This is relevant for Order Index, OPAC Request List, Bulk Ordering List, and Bulk Arrival List.

RefreshOnStart=Y
This indicates whether or not the Index List is refreshed when it is opened. This is relevant for Order Index and OPAC Request List.

[Edit]

```
[Edit]
ExpandTextLength=100
```
Defines the minimum size of a field (as defined in the Oracle tables) that will trigger a small button next to the GUI edit field that will open a multi-line edit field.

[AutoInput]

[AutoInputBulkCheckInItemBarcode]
ILL > Lending tab > Bulk Operation node > [C] Check-in pane > Item Barcode

[AutoInputBulkReturnItemBarcode]
ILL > Borrowing tab > Bulk Operation node > Return pane > Item Barcode

[AutoInputFastCircUserId]
Circulation > Circulation menu > Fast Circulation > Patron ID

[AutoInputHoldItemUserId]
Circulation > Item tab > Items List > Hold Request button > Create Hold Request window > Patron ID

[AutoInputHoldPhotoRequestUserId]
Circulation > Requests menu > Hold Requests > Create Hold Request > Patron ID
Circulation > Requests menu > Photocopy Requests > Create photocopy request > Patron ID

[AutoInputILLItemBarcode]
ILL > Item barcode

[AutoInputInHouseItemBarcode]
Circulation > Circulation menu > In-House Use > Item Barcode

[AutoInputInventoryMarkingItemBarcode]
Cataloging > Items menu > Inventory Marking > Item Barcode

[AutoInputItemBarcode]
Cataloging > Item bar > Item Barcode
Cataloging > Items menu > Change Item Process Status > Item Barcode

Circulation > Circulation menu > Renew Loan by Barcode > Item Barcode
Circulation > Circulation menu > Fast Circulation > Item Barcode
Circulation > Circulation menu > Reserve Items (Advanced Booking) > Item Barcode
Circulation > Item bar (active in item tab, loan tab, return tab) > Item Barcode
Circulation > Items menu > Change Item Information > Item Barcode
Circulation > Items menu > Restore Item Information > Item Barcode
Circulation > Items menu > Change Item Process Status > Item Barcode
Circulation > Requests menu > Hold Requests > Create Hold Request > Item Barcode
Circulation > Requests menu > Hold Requests > Delete Single Hold Request by Barcode
Circulation > Requests menu > Hold Requests > Print Letter - Hold Request Filled > Item Barcode
Circulation > Requests menu > Hold Requests > Print Letter - Hold Request Not Filled > Item Barcode
Circulation > Requests menu > Photocopy Requests > Create photocopy request > Item Barcode
Circulation > Requests menu > Photocopy Requests > Print Photocopy Request Slips > Item Barcode

If this general AutoInputItemBarcode option is set, there is no need for a specific definition for each barcode edit field (such as [AutoInputInventoryMarkingItemBarcode])

[AutoInputNewUserBarcode]

Circulation > Global Patron Information pane > Patron ID (update)
Circulation > Patrons menu > New Patron > Patron ID
Circulation > Patrons menu > Duplicate Patron > Patron ID

[AutoInputOfflineCircItemBarcode]

Circulation > Circulation menu > Offline Circulation > Item Barcode

[AutoInputOfflineCircUserId]

Circulation > Circulation menu > Offline Circulation > Patron ID

[AutoInputPhotoItemUserId]

Circulation > Item tab > Items List > Photocopy Request button > Create Photocopy Request window > Patron ID

[AutoInputPrintLabelItemBarcode]

Cataloging > Items Menu > Print Item Labels > Item Barcode

[AutoInputReturnItemBarcode]

ILL - OutILL\Return menu

[AutoInputRoutReturnItemBarcode]

Acquisitions > Utilities menu > Return Item from Routing Group > Item Barcode

[AutoInputShipItemBarcode]

ILL > Lending tab > Bulk Operation > Ship > Ship Returnable > Item Barcode

[AutoInputShortLoanUserId]

Circulation > Circulation menu > Reserve Items (Advanced Booking) > Patron ID

[AutoInputUserId]

Circulation > Patron bar (active in patron tab, loan tab) > Patron ID

[AutoInputUserListUserBarcode]

Circulation > Patron bar > Patron List > by Patron ID

[AutoInputUserListUserId]

Circulation > Patron bar > Patron List > by Patron ID

Configuration examples:

```
AutoInputMethod=1
AutoInput1=10,9,22
```

or

```
AutoInputMethod=0
AutoInput1=10,11,22
```

If a barcode contains several types of information but the library uses only part of the barcode, automatic input methods have to be applied. The automatic input setup enables the system to extract selective information from an input source. This setup is done with the AutoInput definitions above.

There can be several methods for automatic input. The `AutoInputMethod` indicates which method is used. The current values are:

`AutoInputMethod=0` - No use

`AutoInputMethod=1` - Smart card

`AutoInputMethod=2` - Use call numbers instead of barcodes. This value is relevant only for the `[AutoInputItemBarcode]` section.

`AutoInputMethod=3` - Use the AutoInp3.dat configuration table if it exists to determine the starting point and length; otherwise, use the parameters in AutoInput1 (e.g. 0,12) as the default.

`AutoInputMethod=4` - Perform automatic left zero padding on codes read from a barcode reader, which are shorter than the fixed length.

`AutoInputMethod=5` - This input method does the following manipulation on input barcodes:

If the barcode is 12 characters length or more, no manipulation is performed.

If it is less than 12 characters, it is processed according to the following algorithm before actual retrieval:

(a) If the barcode starts with a digit, it is left-padded with zeros, e.g.
"1234" --> "000000001234".

(b) If the barcode starts with "C", "M" or "S" (see (c) below for "S" with specific prefixes) followed by digits, zeros are added between the alpha prefix and the digits, e.g. "M4567" --> "M00000004567".

(c) If the barcode starts with one of the following prefixes - "S96", "S97", "S98" or "S99", zeros are added between the prefix and the remaining digits, e.g "S981123" --> "S98000001123".

(d) Otherwise, no manipulation is performed.

The `AutoInput1` entry sets parameters with which the input method will work, if it is set to 1. The notation of the entry is as follows:

```
AutoInput1=[a],[b],[c]
```

[a] stands for the number of characters which are skipped before reading the barcode number.
[b] instructs how many characters starting after [a] should be scanned in and identified as a barcode.
[c] stands for the total number of characters which are expected by the system. In other words, after [c] characters are read, `AutoInput` is invoked.

[Services]

```
[Services]
CtlHeight=25
StaticWidth=110
```

CtlHeight=
Indicates, in pixels, the height of the fields in the dialog for GUI Services.

StaticWidth=
Indicates, in pixels, the width of the labels in the dialog for GUI Services.

[FirstTab]

```
[FirstTab]
CheckInListExpand=2
```

When a user changes the default first tab (this is only for forms which are comprised of several tabs, such as CheckInListExpand) via the Visual Properties window, the number entered is saved here. By default the first (the leftmost) tab is activated when a multi-tab window is opened. The second tab from the left is 2, the third tab from the left is 3, and so on.

This variable does not work for the SearchFind window; that is, it does not control the priority of the Search Tab tabs.

Note
The new default tab setting is in effect only after the user has closed the client and re-opened it, not during the same session (this is so for all .ini file changes).

[BubbleMsg]

```
[BubbleMsg]
;in seconds !!!
;If Delay=0 the Bubble message stays till first keyboard stroke.
ErrorMsgTimeDelay=0
```

This relates to the callout bubble invoked by right-clicking on a line in a GUI list.

ErrorMsgTimeDelay=
Indicates, in seconds, the time that elapses before an error message displays.

[XslCreator]

```
[XslCreator]
SplitColor=255,000,000
AddressColor=192,192,110 )
```

This relates to the XSL Creator utility that enables menu-driven creation of additional formats for print forms. This utility, XSL Creator, is available from all modules.

[SearchTree]

```
[SearchTree]
BkColor=255,255,255
```

This parameter defines the color of the Search tree.

[OverviewTree]

```
[OverviewTree]
BkColor=255,255,255
```

This parameter defines the color of the Search tree in Overview mode.

[SearchFind]

```
AutoShow=5
FindAdjacency=N
FindTypes=S,C,F
```

This section relates to the setup of the Find tab in the Search function.

AutoShow=

The system automatically displays records in full view, if the number of records in the set does not exceed or equal the number set here.

FindAdjacency=
This is the default setting of the Words Adjacent option.

FindTypes=
This is the order, left-to-right, in which the various find tabs appear. S=Advanced Search, C=Multi-base CCL Search, F=Multi-field Search.

[PrintDaemon]

```
[PrintDaemon]
Position=Center
Iconize=Y
SleepTime=30
Libraries=USM12,USM50
Targets=printer01
LogFile=N
PrintdAutoActivate=N
PrintdAutoDeactivate=N
```

Position=Center
In this example, the Print Daemon application will be displayed in the center of the screen (not the center of the Task Manager application). Alternately, you can enter a pair of numbers, such as `Position=0,75` to define the offset (in pixels) of the Print Daemon application from the top left corner of the Task Manager application.

Iconize=Y
Determines if the Print Daemon application will be opened to its regular size or iconized when the Print Daemon is started. If you want the window to be iconized, enter Y. If you want the window to be opened to its regular size, enter N.

SleepTime=30
Defines the Look Up Interval which tells the system how often to check for files that need to be printed by the Print Daemon. Enter a number between 1 and 300 (seconds). For example, if you enter 120, the system checks for files every 120 seconds.

Libraries=USM12,USM50
Defines the libraries selected by default in the Print Daemon.

Targets=printer01
Determines the print IDs whose files will be directed to the PC's default printer.

LogFile=N
This variable controls creation and updating of the print daemon log file. Setting the variable to N prevents the log file from being created or updated.

PrintdAutoActivate=N
Determines whether or not the print daemon will be activated automatically when the GUI is activated. To launch the print daemon automatically whenever the GUI is activated, set this value to Y.

PrintdAutoDeactivate=N
Determines whether or not the print daemon will be deactivated automatically when the GUI is closed down. To deactivate the print daemon automatically whenever the GUI is closed, set this value to Y.

[External]

```
[External]
; PopupRunQuery -
;   is used to determine whether a dialog box should be displayed,
allowing the
;   user to change the execution parameters
;
; other lines are in the form
;
; <code>=<format> <import> <exe file>
;
; <code>   - is up to 3 characters and is server-dependent.
; <format> - should be 'A' for ASCII or 'B' for binary (for ftp).
: <import> - This flag is obsolete
;
; <exe file> - the file which should be executed
;              the first %s will be replaced by the file name
received from
;              the server
;
PopupRunQuery=N

;htm=A X C:\Program Files\Internet Explorer\IEXPLORE.EXE %s
htm=A X C:\Program Files\Internet Explorer\IEXPLORE.EXE %s
;htm=A X c:\Program Files\Netscape\Communicator\Program\netscape.exe
%s
;htm=A X C:\Program Files\Netscape\Netscape 6\netscp6.exe %s
xv=B X c:\lview\lview31.exe %s
ig=B X c:\lview\lview31.exe %s
txt=A X c:\windows\notepad.exe %s
doc=B X C:\Program Files\Microsoft Office\Office\WINWORD.EXE %s
tln=A X telnet.exe %s
pdf=B X C:\Program Files\Adobe\Acrobat 6.0\Reader\AcroRd32.exe %s
```

The External button on the Full+Link display of a record is available when the user highlights the 856 (External) field. In order for the system to activate the external program, you must first define various parameters.

PopupRunQuery
Enables you to determine whether or not a dialog box should be displayed to the user, allowing him to change the execution parameters.

Other lines are in the following form:

```
; <code>=<format> <import> <exe file>
```

Each element is explained below.

<code>

Enter the code for the type of file. Examples are xv, ig, txt, htm, doc, and tln. The code that is entered here should be one that is used in the Catalog record.

<format>
This indicates whether the file contains text or images. Enter A for ASCII (for text) or B for binary (for images).

<import>
You must always enter the letter X, which stands for a code that is not currently in use

<exe file>
This is the path of the program that should be run. An example is `c:\windows\notepad.exe`

%s
This is a code that the system will replace with the actual filename.

[FirstTab]

```
[FirstTab]
SubscrListExpand=2
ItemExpand=2
```

This is an option which allows you to set the default open tab, that is, the tab that first appears when a multi-tab window is selected. This is defined as <window name>=n, for example, SubscrListExpand=2.

[Package]

```
[Package]
AlwaysImportFiles=Y
```

In the Cataloging module, when you connect to a Home Library, the system checks that the module is using the most recent version of the database tables. If the `AlwaysImportFiles` flag is set to *Y*, the system automatically loads the updated tables without a load prompt being displayed. If the `AlwaysImportFiles` flag is set to *N*, the user is asked if she/he wants to load the library tables.

This flag also determines whether or not to load automatically the most recent printout templates from the server to the GUI client. If the `AlwaysImportFiles` flag is set to *Y*, the system loads the updated templates automatically. If the `AlwaysImportFiles` flag is set to *N*, the user will be asked if she/he wants to load the templates.

[PrintExecute]

```
ERROR_EDITOR=notepad.exe
EDIT=notepad.exe $1
BROWSER=C:\Program Files\Plus!\Microsoft Internet\Iexplore.exe $1
EXE_COMMAND=C:\Program Files\Plus!\Microsoft Internet\Iexplore.exe $1
WORD=C:\Program Files\Plus!\Microsoft Internet\Iexplore.exe $1
SaxonVersion=saxon8.jar
```

This section defines which external program will be used to open an XML print file. This section works according to the setup of column 3 of `XXX50/tab/form_print_method`.

SaxonVersion
This variable defines which utility will be used for printing. The available options are Saxon7.jar and Saxon8.jar. Note that both options require an installed Sun Java Virtual Machine. If no parameter is entered, the default is Saxon7.jar.

[ATabControl]

```
[ATabControl]
FgColor=000,000,128
```

This setting defines the color of the text in the tab controls.

[ABaseWindow]

```
[ABaseWindow]
ButtonHeightFactor=1.5
MaxButtonText=13
```

`ButtonHeightFactor=1.5`
The height of the buttons in the Base windows.

`MaxButtonText=13`
The maximum length of the text on the buttons in the Base windows. The length of the text can be from 5 to 40 characters. The default length is 12.

[PlainTextPrintout]

```
[PlainTextPrintout]
CharsetL=iso-8859-1
CharsetU=utf-8
FontL01=16Courier New
FontL02=18Courier New
FontL03=20Courier New
FontL04=22Courier New
FontL05=24Courier New
FontU01=16Courier New
FontU02=18Courier New
FontU03=20Courier New
FontU04=22Courier New
FontU05=24Courier New
```

These are the font definitions for printing of lists and for PLAIN printouts.

[ListBoxPrintout]

```
[ListBoxPrintout]
Font=16Bitstream Cyberbit
```

This line defines the size and the font used for printing lists from List Box windows.

[ItemForm]

```
[ItemForm]
DupCallNo=N
DupCallNo2=N
```

The lines `DupCallNo` and `DupCallNo2` determine whether Call Number and Call Number 2 are duplicated when duplicating an item. If set to "Y", the call number is duplicated. Note: If your call number is assigned by the system counter, set to "N".

[Vendor]

```
[Vendor]
DuplicateAddress=N
```

DuplicateAddress
Determines whether or not the Vendor Address (Z72) is duplicated when duplicating a Vendor/Supplier(Z70) using the Duplicate button on the Vendor List. The default value is Y. Note that in the ACQ and ILL ini files, the `DuplicateAddress` parameter does not exist and therefore the Vendor Address will always be duplicated (the same as value Y).

[VendorList]

```
[VendorList]
SaveVendorListStartPoint=Y
VendorListStartPoint=SWETS
VendorListSortOption=C
```

SaveVendorListStartPoint=Y
Determine whether or not to save the 'Vendor List' starting point. The value Y activates the saving option in the Vendor List. The value N deactivates the saving option in the Vendor.

VendorListStartPoint=SWETS
When the Vendor List window is opened and `SaveVendorListStartPoint` is set to *Y*, the starting point is stored in `VendorListStartPoint`.

VendorListSortOption=C
When the Vendor List window is opened and `SaveVendorListStartPoint` is set to *Y*, the "Sort by..." parameter (Name or Code) is stored in `VendorListSortOption`. Possible values are: C (code) or N (name).

[Splitter]

```
[Splitter]
HighlightFocusedPane=Y
FOCUS_HILIGHT_COLOR_LTO=180,075,025
FOCUS_HILIGHT_COLOR_RBO=245,005,025
FOCUS_HILIGHT_COLOR_LTI=145,095,075
FOCUS_HILIGHT_COLOR_RBI=220,065,040
FocusedPaneBrokenLine=Y
```

```
FOCUS_HILIGHT_COLOR_BROKEN=255,0,0
```

This refers to the mechanism for highlighting the current work area (pane). You can move from pane to pane in the GUI by clicking in a pane, or by pressing SHIFT + TAB.

Note
Do not adjust the color settings.

HighlightFocusedPane=Y
If set to Y, a colored border appears around a selected frame.

If set to N, borders do not appear around selected frames.

FocusedPaneBrokenLine=Y
If set to Y, the colored border that appears around a selected frame will be displayed as a broken line. This variable has no effect if the HighlightFocusedPane variable is set to N.

FOCUS_HILIGHT_COLOR_BROKEN=255,0,0
This variable sets the color of the broken line that is displayed around a selected frame if the FocusedPaneBrokenLine and HighlightFocusedPane variables are set to Y.

[TextOrientation]

```
[TextOrientation]
AlwaysLeft=N
```

This line controls the text direction of the lines in the full view of a record in the Show node of the Search tab.

If N is present, the text will be aligned according to the language direction. If Y is present, the direction of the text will always be left-aligned.

[RfidMedia]

```
[RfidMedia]
RfidReaderAddress=10.1.235.101
Vendor=Bibliotheca
```

The [RfidMedia] section is required when library items are RFID-tagged. For more information on using RFID equipment, please refer to the How-to document in the Ex Libris Documentation Center that is relevant to your RFID vendor (for example, for Bibliotheca, you would refer to *How to set up a BiblioChip® interface in ALEPH® 500 - 18.01*).

RfidReaderAddress=10.1.235.101
If the library uses NEDAP equipment for RFID, this variable is used to setup the IP address of the RFID Reader that is used by the client station.

Vendor=Bibliotheca
This variable is used to determine the RFID vendor whose equipment is applied at the library. Supported values are Bibliotheca , which is the default value, and Nedap.

3 Font Definitions (font.ini)

The font.ini file contains font definitions for Find, Browse and cataloging forms' edit fields (WindowControls); for list boxes and captions (`ListBox` and `ListBoxCaption`); for the Cataloging draft (`EditorTag`, `EditorField` and `EditorDescription`); and for the Cataloging floating keyboard (`AlephKeyboard`).

Note
You can define different fonts for different Unicode ranges (columns 2 and 3 of the file).

Following is a sample of the `font.ini` file:

```
!       1              2   3                   4               5 6 7
8    9
!-------------------+----+----+-------------------------------+-+-+-+-
-+--------------------
UnicodeEdit           0000 00FF Courier                        N N N
18 DEFAULT_CHARSET
UnicodeEdit           0000 FFFF Bitstream Cyberbit             N N N
18 DEFAULT_CHARSET
```

Key to the font.ini file:

Column 1 – Context

Instance for which the fonts are being defined. Values are:
`ListBox`, `ListBoxCaption` (list boxes and list captions),
`WindowControls (Find, Browse and cataloging forms edit fields)`
`EditorTag`, `EditorField` and `EditorDescription` (cataloging draft).
`AlephKeyboard` (cataloging floating keyboard)
Window Controls (buttons, edit fields, static fields and so on).
`TabNormal` (used for tab)
`TabSelect` (used for the current tab selected (Underline+bold))
Viewer
`DefaultPrintout`, `PlainTextPrintout`, `ParagraphPrintout`, `TablePrintout`, `TextLbPrintout` (printout definitions)

Column 2 - 'From' Unicode value
'From' Unicode value (hexadecimal). Use to define the starting point for the range definition.

Column 3 - 'To' Unicode value
'To' Unicode value (hexadecimal). Use to define the end of the range definition.

Column 4 - Facename of Font
Font (for example, Courier).

Column 5 - Bold
Values are Y and N. Determines if bold formatting should be applied.

Column 6 - Italic
Values are Y and N. Determines if italic formatting should be applied.

Column 7 - Underline
Values are Y and N. Determines if underline formatting should be applied.

Column 8 - Font size
Determines the size of the font.

Column 9 - Character Set
Defines the character set that is used. Possible character sets are:

```
ANSI_CHARSET
DEFAULT_CHARSET
SYMBOL_CHARSET
SHIFTJIS_CHARSET
HANGEUL_CHARSET
GB2312_CHARSET
CHINESEBIG5_CHARSET
OEM_CHARSET
JOHAB_CHARSET
HEBREW_CHARSET
ARABIC_CHARSET
GREEK_CHARSET
TURKISH_CHARSET
THAI_CHARSET
EASTEUROPE_CHARSET
RUSSIAN_CHARSET
MAC_CHARSET
BALTIC_CHARSET
```

4 Working with ALEPH Services

In ALEPH, you can run services (batch jobs) to generate reports and to carry out file maintenance. All services are managed from the GUI modules. Each module has a menu named **Services** on the menu bar.

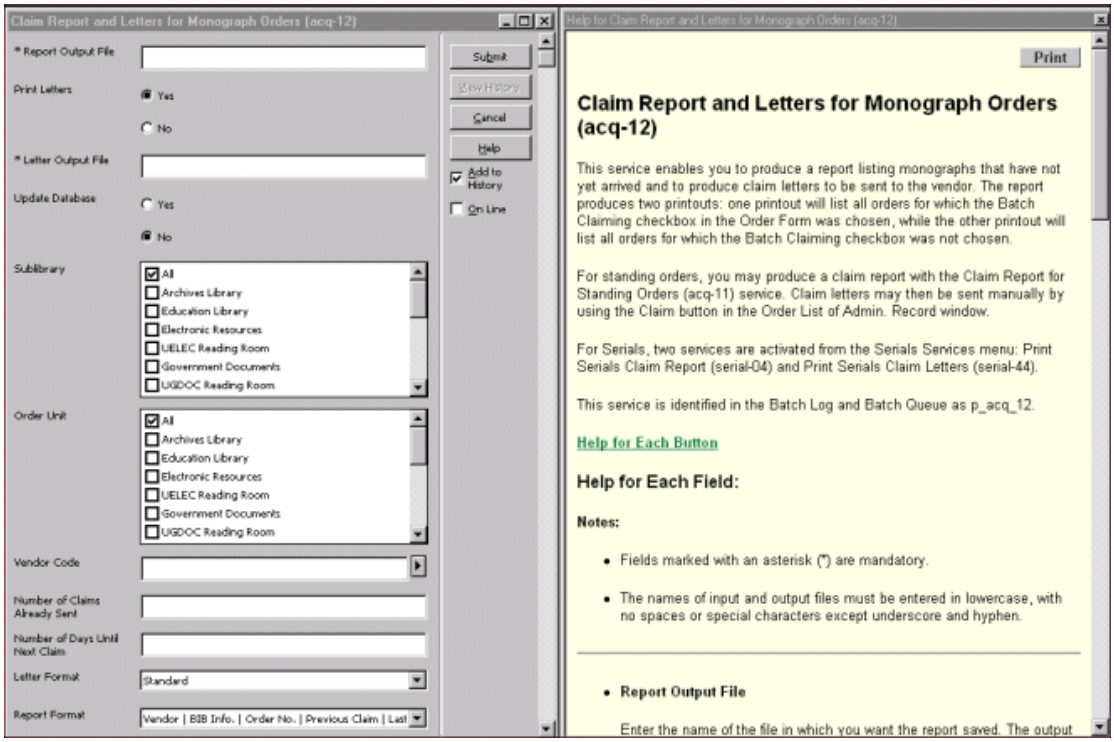
If a staff user does not have permission for `./alephe/tab/user_function.lng` WWW-B/MAIN, the Services menu appears dimmed in all GUI modules.

The services screens are in XML format and are stored in the `./alephe/pc_b_lng` directory, which you can access in vi by using the shortcut *pcb*. The Help screens are in HTML format and are stored in the `./aleph/pc_b_lng` directory. They should not be edited, as they will be replaced by version upgrade. The HTML help files for

custom services must be placed in the `./alephe/pc_b_lng` directory, where they will be retained.

The Services menu and submenus are built by menu files in the `./alephe/pc_b_lng` directory.

Here is an example of a service which is accessed from the Acquisitions GUI, Claim Reports and Letters for Monograph Orders (acq-12):



4.1 Batch Service Sub-Library and Order Unit Sensitivity

Some of Aleph batch services can be run for all sub-libraries or specific sub-libraries, for all order units or specific order units. This is determined by the staff users, who choose the sub-libraries and order units for which they want the service to run.

To display list of sub-libraries on the service form, **SUB-LIBRARY** source type should be used in ./alephe/pc_b_lng/<service name>.xml. For example:

```
<control>
        <expand_checkbox>
                <label>Sublibrary</label>
                <source>SUB-LIBRARY</source>
                <argname>F05</argname>
        </expand_checkbox>
</control>
```

In order to display only sub-libraries that the staff user is allowed in, **SUB-LIBRARY-AUT** use source type instead of **SUB-LIBRARY.**

To display list of order units on the service form, use **ORDER-UNIT** source type in ./alephe/pc_b_lng/<service name>.xml. For example:

```
<control>
        <expand_checkbox>
                <label>Order Unit </label>
                <source>ORDER-UNIT</source>
                <argname>F05</argname>
        </expand_checkbox>
</control>
```

To display only order units that the staff user is allowed in, use **ORDER-UNIT-AUT** source type instead of **ORDER-UNIT**.

Matching permissions should be given to staff users (sub-library permissions are assigned for each service).

The "All" option includes also sub-libraries / order units that the staff user is not allowed in. To avoid the "All" option, the "<empty_line>NO_USE</empty_line>" should be defined in ./alephe/pc_b_lng/<service name>.xml. For example:

```
<control>
        <expand_checkbox>
                <empty_line>NO_USE</empty_line>
                <label>Sublibrary</label>
                <source>SUB-LIBRARY-AUT</source>
        </expand_checkbox>
</control>
```

5 Setup for Expanded Information Display

Expand displays show complete information on a line in the GUI lists. The expand display is formatted in HTML templates. The files that control the display are stored in `XXX50/pc_display`. These files do not have an extension. One file is used for each expand display. The information from one file can be distributed among several tabs.

In addition to the HTML files, transformation files are used to change the data from database data to meaningful formatted data. These files have the `.trn` extension.

The files are stored in one subdirectory per language `pc_display_<lng>`. For a second language, the complete set of files must to be present in this language's subdirectory. For example, `XXX50/pc_display_por` for the setup of Portuguese language display.

The following templates are used:

HTML Template	Module	Where	Tabs
budget-balance	Acquisitions/Serials		
ill-in-request	ILL		
ill-in-return	ILL		
ill-patron	ILL		

HTML Template	Module	Where	Tabs
ill-requests	ILL		
ill-return	ILL		
ill-supplier	ILL		
item-current-summary	Circulation, Cataloging	Item Tab / Circ Summary / 1. Current Summary	Item, Loan
item-expand	Circulation, Cataloging	Item Tab / Item	Item, Bib Info, Loan
item-history-expand	Circulation, Cataloging	Item Tab / History / 3. Item Changes / -> View button / Item Information	Item, Bib Info, Change Info
item-loan	Circulation	Item Tab / Loan	Loan, Item
item-return	Circulation	Return Tab / Session	Item, Bib Info
Item-summary	Circulation, Cataloging	Item Tab / Circ Summary / 2. History Summary	History
order-info	Acquisitions/Serials	Order Search Tab / Index List	Order Info, Bib Info, Invoice, Expenditure, Encumbrance
Record-bibliography	Cataloging		
Subscription-info			
user-cash	Circulation	Patron Tab / Cash	Cash Transaction, Bib Info, Item, Loan
user-details	Circulation	Patron Tab / Session / Patron Information / Patron Details	
user-details	Circulation	Loan Tab / Session / Patron Information	
user-loan	Circulation, Cataloging	Patron Tab / Loans / 1. Loan Expand	Loan, Bib Info, Item

To view the entire range of data that the system creates for a particular record:

1. On the server, from the command prompt, enter:

   ```
   >> pc_display_off
   ```

2. Re-start `pc_server`. The "expanded display" displays in XML format, giving you all possible data. It is possible that there is more XML data than is mapped to the HTML.

In order to set the system back to HTML display:

1. On the server, from the command prompt, enter:

```
>> pc_display_on
```

2. Re-start pc_server.

5.1 The HTML Template

SECTION

The `SECTION` tag defines different parts of the HTML template. These parts are predetermined by Ex Libris and correspond to the sections used in the data files.

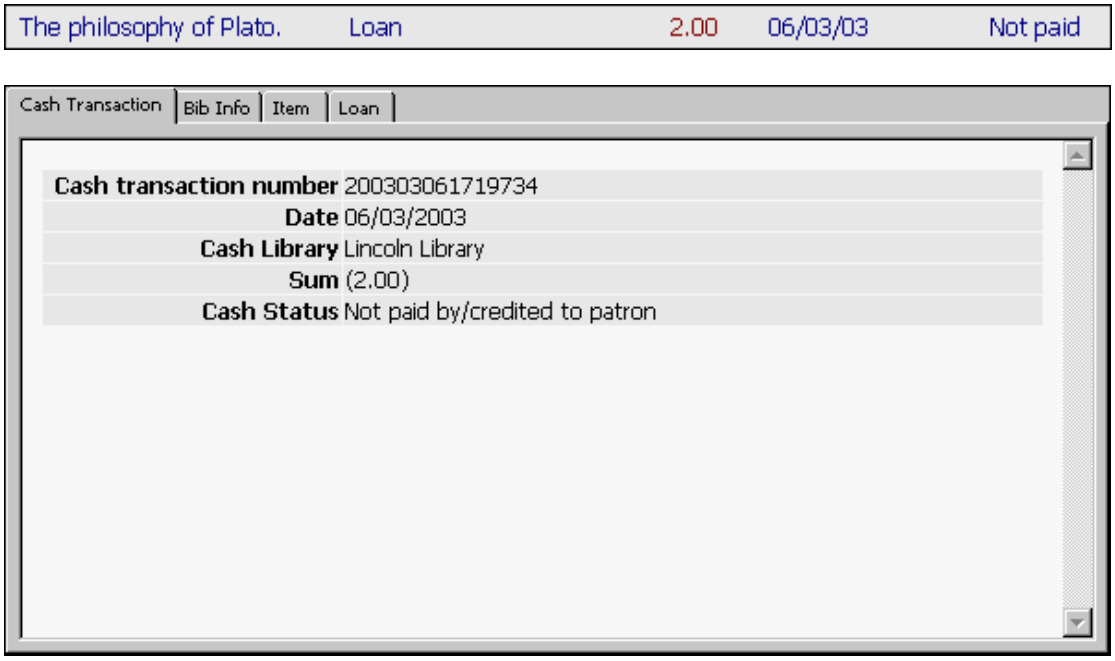
Each template is made up of at least one section. A section is defined by the opening and closing tags:

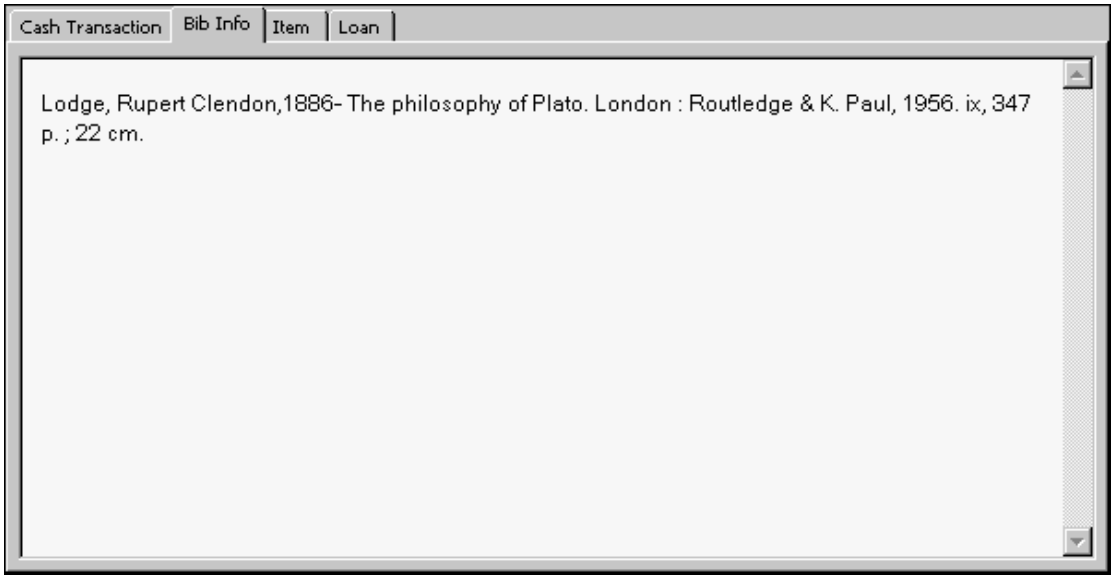
SECTION-01

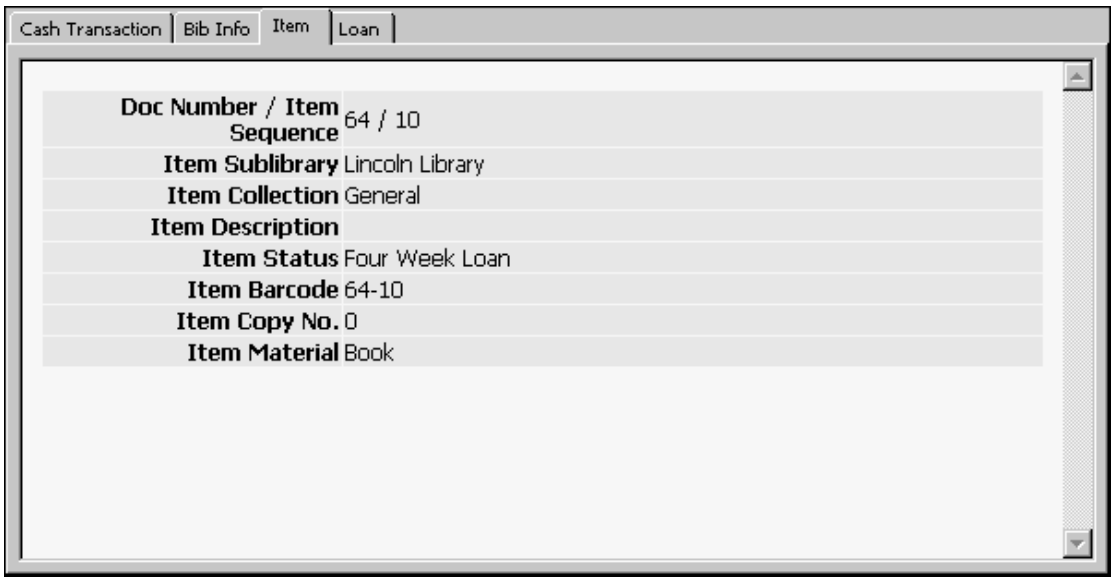
END-SECTION-01

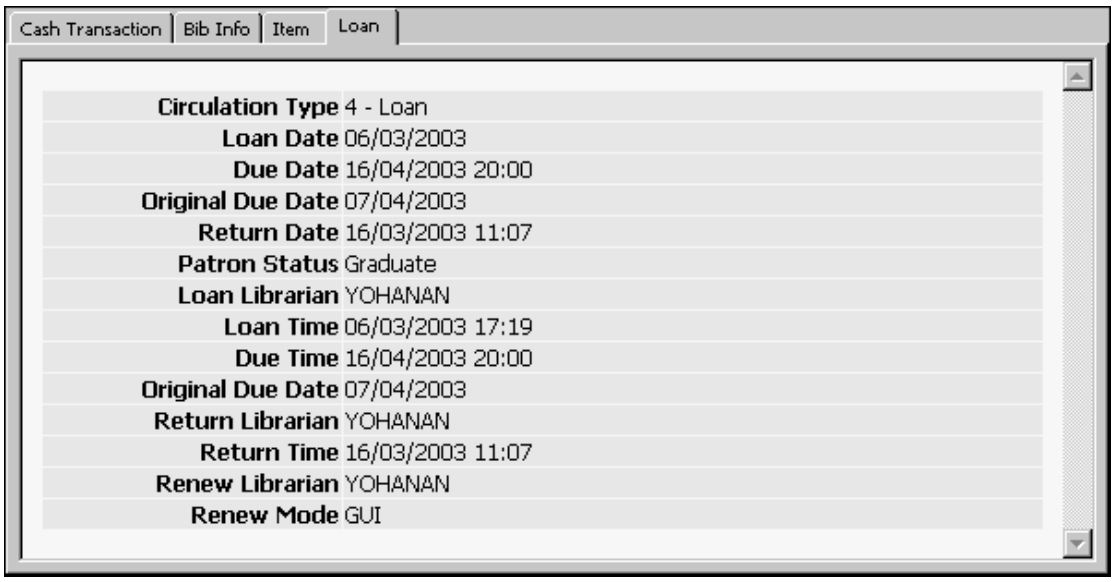
Sometimes `SECTION` tags (such as `SECTION-01` / `END-SECTION-01`) will appear a few times in the template. This is because the `SECTION` / `END-SECTION` tags must be defined at the beginning and the end of each tab. The values are still from the same section.

For example, in the user-cash template, the `SECTION-01` / `END-SECTION-01` tags appear a few times because there are four tabs.




Cash Transaction
Bib Info
Item
Loan
Lodge, Rupert Clendon,1886- The philosophy of Plato. London : Routledge & K. Paul, 1956. ix, 347 p. ; 22 cm.


Cash Transaction
Bib Info
Item
Loan
Doc Number / Item Sequence 64 / 10
Item Sublibrary Lincoln Library
Item Collection General
Item Description
Item Status Four Week Loan
Item Barcode 64-10
Item Copy No. 0
Item Material Book


Cash Transaction
Bib Info
Item
Loan
Circulation Type 4 - Loan
Loan Date 06/03/2003
Due Date 16/04/2003 20:00
Original Due Date 07/04/2003
Return Date 16/03/2003 11:07
Patron Status Graduate
Loan Librarian YOHANAN
Loan Time 06/03/2003 17:19
Due Time 16/04/2003 20:00
Original Due Date 07/04/2003
Return Librarian YOHANAN
Return Time 16/03/2003 11:07
Renew Librarian YOHANAN
Renew Mode GUI

LABEL

The LABEL tag creates and defines a tab. The name in the LABEL definition is displayed as the name of the tab.

In the example above the tab definitions are:

```
#LABEL Cash Transaction#
#LABEL Item#
#LABEL Loan#
```

The LABEL tag does not have a closing tag. The tab begins after the LABEL tag and ends before the next LABEL tag or at the end of the file.

Note
A tab will not be displayed if all fields in the tab are empty.

The CONDITION tag can be used together with the LABEL tag to define a tab that will only be displayed under certain conditions.

INCLUDE

The INCLUDE tag is used to insert a predefined modular part to the display. This is useful when the same element is used by several files. The include files are in the `pc_display_<lng>` directory. The include files are:

```
html-head
html-tail
open-table-1
open-table-2
style-css
bib-info
```

html-head

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>60;title>
<include>style-css
<head>
<body>
```

Each tab display is opened with `html-head`. The INCLUDE must appear directly after the LABEL tag. For example:

```
##LABEL Item##
<include>html-head
```

html-tail

```
<body>
<html>
```

Each tab display is closed by `html-tail`.

open-table-1 and open-table-2

In most screens the display is a two-column table. `open-table-1` and `open-table-2` are used to open and define table definitions. Note that table definitions are closed with the `</table` tag.

5.2 style-css

bib-info

The information displayed in the Bib Info tab uses a special definition. In order to include the Bib Info tab in an HTML template insert the line:

```
<include>bib-info
```

The Bib Info tab will be displayed if there is bibliographic information. (For example the cash transaction Local Patron Registration does not involve an item, and therefore does not have any bibliographic information.)

The following explanation is only relevant if you are going to modify the standard Bib Info display as provided by Ex Libris.

The bib-info Include File

The bib-info include file uses the LABEL, CONDITION and SECTION labels, and the html-head and html-tail include files.

```
!
! Bib information
```

```
! section-01 - dependent on existence of bib-info
!

SECTION-01

CONDITION bib-info

##LABEL Bib Info##

<include>html-head


<div class="table-bib-info">
  ./bib-info
<div>
<table>

<include>html-tail

END-CONDITION

END-SECTION-01
```

The actual bibliographic information is displayed in the `bib-info` field and is dependent on the setup of the `global.trn` and `xxxx.trn` tables in the `XXX50/form_<lng>` directory.

CONDITION

The CONDITION tag is used to suppress the display of one or several defined fields under one of the following conditions:

- The field is not in the raw XML file that was generated.
- The field is empty.
- The field's value is zero or zeroes.

Example:

```
CONDITION z36-recall-date
  <tr>
     <th class="table-1" id="bold" align="right">
        Recall Due Date
     <th>
     <td class=table-1>
        ./z36-recall-due-date
     <td>
  <tr>
  <tr>
     <th class="table-1" id="bold" align="right">
        Recall Date
     <th>
     <td class=table-1>
        ./z36-recall-date
     <td>
  <tr>
END-CONDITION
```

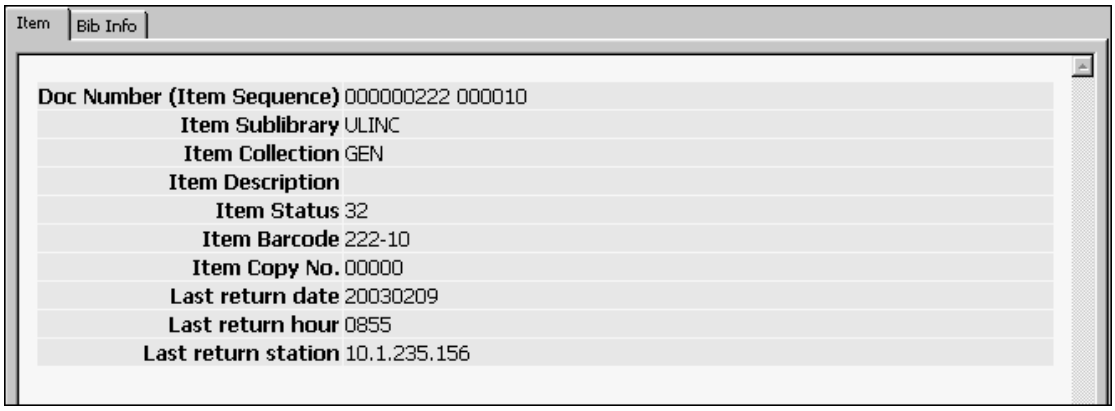
In this example, if z36-recall-date is empty, it means that the item was not recalled. Therefore, there is no Recall Date and no Recall Due Date to display.

CONDITION <field>=<value>

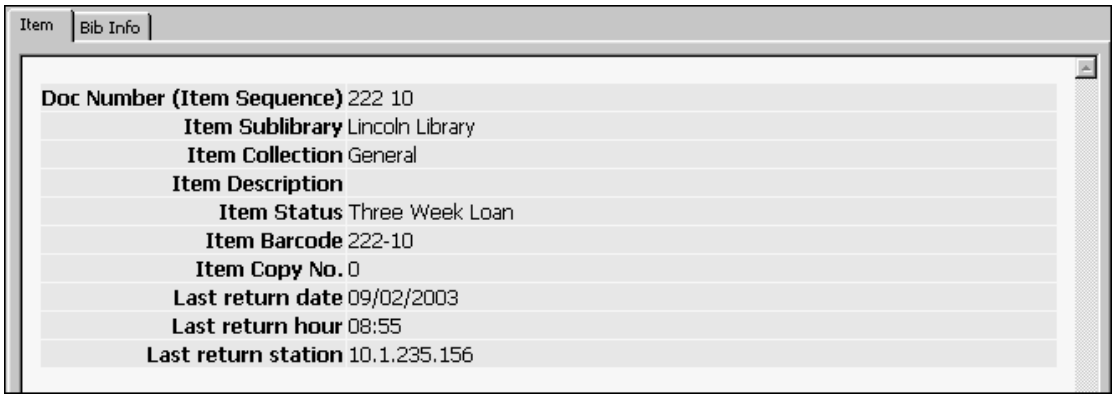
The `CONDITION <field>=<value>` tag is used to display of one or several defined fields when the condition is met.

5.3 Transformation

The ALEPH Transformation Mechanism changes the displayed data from database data:



to meaningful formatted data:



The transformation files are stored in `XXX50/pc_display` and if not found there - they are searched in `XXX50/form_<lng>`.
Note that a path_convert can be used to redirect from XXX50/form_<lng> to XXX01/form_<lng>.
There are two types of transformation files: `global.trn` and `xxxx.trn`, where xxxx = the HTML print templates. These are ALEPH tables which contain data relating to the ALEPH transformation rules. Here is an extract from the `global.trn` ALEPH table:

```
!                                                M
!                                                /
!Tag                                  Type        S Parameters
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!!->
bib-info                              SUPPRESS    S
```

```
z303-delinq-1-update-date           DATE       S
z303-delinq-2-update-date           DATE       S
z303-delinq-3-update-date           DATE       S
z305-delinq-1-update-date           DATE       S
z305-delinq-2-update-date           DATE       S
z305-delinq-3-update-date           DATE       S
```

Column 1 : field name - XML tag
Column 2 : data type - ALEPH transformation method
Column 3 : single / multiple - use either S or M
Column 4 : (additional) parameters

M is used in column 3 when data is composed of several codes delimited by spaces, and each is less than 100 characters.

6 Setting up User Logons

When a user opens the first GUI module on your PC, if he is not already logged on, the software prompts him for a user name and password. It checks all the servers listed in the ..\ALEPHCOM\DAT\library.ini file. If this name/password is not valid on any of the servers it can connect to, he gets the message "Password not verified on connectable hosts". This indicates that:

- This name/password is not valid, or
- The address specified in the ..\ALEPHCOM\DAT\library.ini is wrong, or
- The pc_server specified in the library.ini entry is down.

Column 3 of the library.ini file is limited to 30 characters. Thus, when the host name is long and, together with the port number, will be longer than 30 characters, use the numeric IP address. For example, instead of:

```
my_very_long_domain_name_1:6991
```

use

```
010.001.235.019:6991
```

Assuming that the password is verified, you then connect to a database (click **ALEPH** in the menu bar, then **Connect**). When you click Connect, the system displays the libraries listed in the module's per_lib.ini file (for example, `..\CIRC\TAB\per_lib.ini`).

When you click on a particular library (or base), the client software goes to the `..\ALEPHCOM\DAT\library.ini` file and tries to connect to the IP address specified for this library.

Note
Since the ALEPHADM module does not require a connection to a particular library as with other modules, the `..\ALEPHCOM\DAT\library.ini` file must contain a PCVER type of line with IP and port definitions for the ALEPHADM version compatibility. Following is a sample of the line:

```
PCVER                          PCVER 10.1.235.19:6997
```

For customers who use a single IP address for all PC workstations, it is possible to use the Windows logon name to identify a PC workstation instead of the IP address.

To implement this functionality:

1. Run `$alephe_root/aleph_shutdown` and make sure that no Aleph processes are running.
2. Run the following command:

   ```
   csh -f $aleph_proc/clear_vir01 VIR01
   ```

3. Add the following line to the file `$alephe_root/aleph_start.private`:

   ```
   setenv use_login_for_lock Y
   ```

4. Run the command `source $alephe_root/aleph_start`" or log on again to the server.
5. Run $alephe_root/aleph_startup.

7 Setting Up the Date Format

In ALEPH GUIs and in the Web OPAC, you can set up the date format both for screen display and for forms. The default settings of the date format are set in `$alephe_root/aleph_start` and/or in `$alephe_root/pc_server_defaults`. If defined in both files, `pc_server_defaults` takes precedence.

You can choose whether ALEPH services use North American date style (mm/dd/yy) or European date style (dd/mm/yy). The line `aleph_date_time` defines the date style display (1 - US, 2 - European). For example:

```
setenv    aleph_date_time        2
```

The line `setenv date_style_s nn` (s=screen) or `setenv date_style_f nn` (f=form) defines the date display.

Example:

```
setenv date_style_s 12
```

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

with separator

first digit

0	separator '-'
1	separator '/'
2	separator ' '

second digit

1	Dmy
2	Dmy with century
3	Dmy with century, long month
4	Dmy with century, short month
5	Mdy
6	Mdy with century
7	Mdy with century, long month
8	Mdy with century, short month

<u>without separator or with separator '/'</u>

first digit

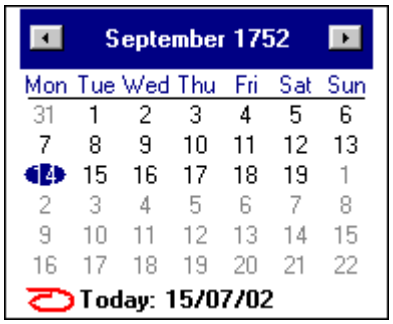
3	no separator
4	separator '/'

second digit

1	ddmmyy
2	mmddyy
3	yymmdd
4	ddmmyyyy
5	mmddyyyy
6	yyyymmdd

Date Range

The earliest possible date is 14/09/1752 (this is the date that the Gregorian calendar was introduced in the United Kingdom):



The latest possible date is 31/12/9999:

8 Setting Up the Time Format

In ALEPH GUIs and in the Web OPAC, you can set up the time format both for screen display and for forms. The default settings of the time format are set in `$alephe_root/aleph_start` and/or in `$alephe_root/pc_server_defaults`. If defined in both files, `pc_server_defaults` takes precedence. The line `setenv time_style_s nn` (s=screen) or `setenv time_style_f nn` (f=form) defines the time display.

Example:

```
setenv time_style_s 04
```

The numbers denote the format. The default is *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

9 Setting Up Navigation and Overview Trees

You can modify the following lines in `pc_server_defaults` in order to adjust the default display of the ITEM-HOL connection in the Cataloging navigation tree and overview trees:

```
setenv item_hol_tree_style              3
```

1 = Do not display the connection between the item and the HOL record.
2 = Display items under both the ADM record and under the linked HOL record.
3 = Display items only under the linked HOL record, and display HOL libraries before ADM libraries

```
setenv expand_tree_style            2
```

This line defines if nodes display open or closed in the Overview tree and in the Cataloging navigation tree:

1 = Do not open, only the selected record node will be visible.
2 = Open only the record nodes (ADM, BIB and HOL).
3 = Open all tree nodes (ADM, BIB, HOL, items, order, subscriptions, and so on).

```
setenv pc_tree_view_max_branch         10
setenv pc_filter_tree_view_max_branch  99
```

These lines define the number of leaf nodes (records) that display in the GUI Navigation Tree. Trees are limited to a maximum of 800 lines.

10 Setting Up Statistic Analysis of Server Use

Z34 records can be used for statistical analysis of server use. You can modify the following line in `pc_server_defaults` in order to control whether or not this option is available:

```
setenv create_statistics              Y
```

If set to *Y*, the `last-z34-sequence` counter must be defined in the Z52 table (UTIL G/2).

11 Setting Up Lock Periods

You can modify the following line in `pc_server_defaults` in order to define the period that Acquisitions, ILL, and Circulation records are locked for (that is, the length of time that other staff users are prevented from updating and deleting records):

```
setenv default_lock_period
```

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:

12 Setting Up Form Information Display

You can modify the following line in `pc_server_defaults` in order to determine whether or not information about forms is displayed in the `pc_server_log` (`$TMPDIR`):

```
setenv form_load_display        Y
```

13 Setting Up Timeouts for Client's Link to Server

The following line in `pc_server_defaults` determines the maximum response time for getting a service from the server - after which time, *failed to read reply* or *failed to connect to host* messages are displayed:

```
setenv PC_SERVER_TIMEOUT  60
```

The defined time is in seconds.

The maximum value that can be defined is 600.

14 Setting up a Transactions Log

The following line in `pc_server_defaults` determines whether or not PC transactions are written in a log file:

```
setenv pc_transactions_log
```

If set to *Y*, a `pc_ser_<port>` file is created in `$LOGDIR` and transactions are written to this file.

If set to *N*, a pc_ser_<port> file is not created in $LOGDIR.

It is recommended to set this parameter to *N* in order to improve performance.

15 System Errors and Messages

The error and messages files in ALEPH are located under the `$aleph_root/error_lng` directory. Localization and translation can be performed under the `$alephe_root/error_lng` directory. The system will first search in `$alephe_root/error_lng` and if the error does not exist, it will then take the standard from `$alephe_root/error_lng`. The extension `lng` is the language code, e.g. `ger`, `eng` or `fre`.

16 Batch Queue

The library's batch queue is managed through the `que_batch` file in the library's `data_files` directory. All batch job requests are written to this file.

To start the library batch queue, use UTIL C/2. This invokes the `<lib>_batch` process (for example, `USM01_batch`) which checks the `que_batch` file and sends the earliest waiting job for execution. After UTIL C/2 has been initiated, the process remains running until system shutdown, or until UTIL C/3 is invoked.

If jobs are submitted to `batch_que` when the queue is not running, the jobs are held in a buffer, and will be executed in chronological order as soon as the queue is started. A log file of the `lib_batch` process, called `run_b.<nnnn>`, is written in the library's `data_scratch` directory. The number appended to the name indicates the process number, and a new log file is opened each time that the `lib_batch` process is started.

You must invoke UTIL C/2 when:

- The system is activated. This can be set automatically in the system's startup procedures.
- The queue has been stopped using UTIL C/3.

You can run a single batch queue for a group of libraries. You can do this by using `tab_library_group` which defines a group of libraries to share the same batch. For example, according to the following line, the hosting library, in which the batch queue

will be running, is USM10. The batch queue will handle USM11 and USM12's batches too.

```
!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!
USM10 USM11 USM12
```

The batch queue can be started and closed only from the library in the first column.

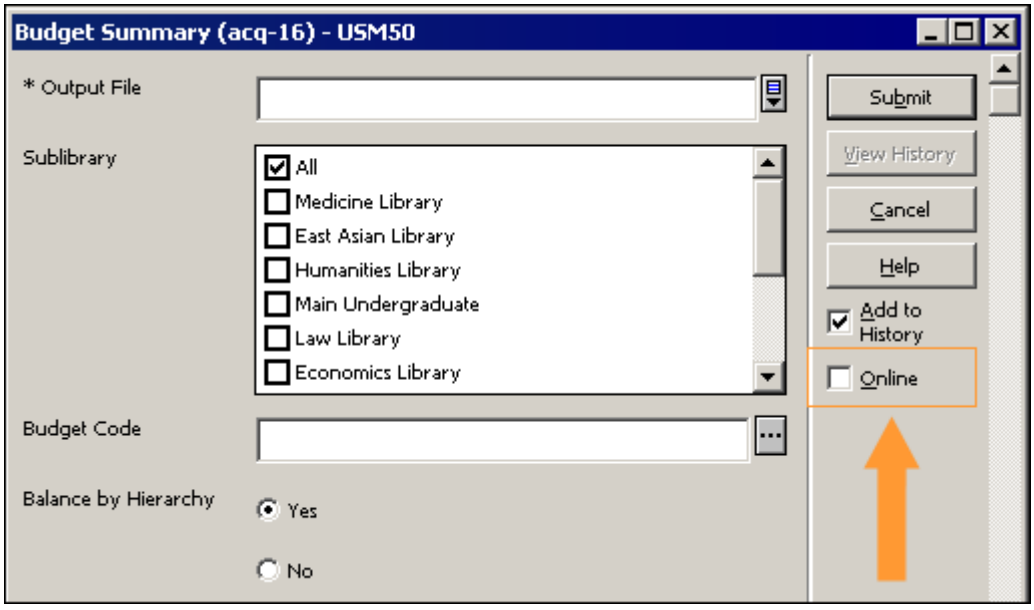
Note
Staff users running a GUI service can select the Online option to run the service immediately without submitting it to the batch queue.

17 Setting Up Staff Privileges

Refer to the Staff Privileges module of the ALEPH User Guide for a complete description of the process of assigning privileges to staff users.

18 Setting the Online Check Box in Services

There is an option to add a check box named **Online** to services:



When this check box is selected, the job will be run immediately from the GUI in online mode, thereby circumventing the batch queue.

To add the Online check box to a service, enter the

```
<allow_online></allow_online>
```

tag in the service's XML file. For example, the opening lines of the p-acq-16 service might appear as follows:

```
<?xml version = "1.0"?>

<!DOCTYPE dialog SYSTEM "dialog.dtd">
<dialog>
<title>Budget Summary (acq-16)</title>
<proc>p_acq_16</proc>
<allow_online>
</allow_online>
<control>
        <edit>
                <argname>F01</argname>
                <label>* Output File</label>
                <size>100</size>
        </edit>
```

Do not add this option to services which lock the library (such as indexing).

19 Setting Up a Check for UTF-8 Support for Web Browser

Two tables, **no_utf_pc** (for PCs) and **no_utf_mac** (for Macintosh computers), are used in ALEPH to check whether or not the Web browser in use supports UTF-8 for the PC. They reside under the `$alephe_tab` directory. These tables contain a list of browsers/versions (user agents) that do not fully support UTF-8. For example:

```
Netscape 4
MSIE 4
```

The main server checks the user agent of each request against the values in this table. If the current user agent is found in the table, then the browser is considered as not supporting UTF-8. In this event, the browser uses the character set defined in the following line in `www_server.conf`:

```
setenv server_default_charset        "iso-8859-1"
```

You can configure the list of browsers to be considered as "not supporting UTF-8" by adding browsers and versions to this table.

20 Setting up External Services

The www_base variable must be defined in aleph_start and not in www_server.conf if the Catalog GUI's External Services are used.

20.1 ALEPH – SFX Integration

Since ALEPH supports two versions of OpenURL for connecting to SFX, a special parameter in the `aleph_start` file defines which version of OpenURL is currently used by the system.

```
setenv sfx_openurl_version    ver-0.1
```

stands for OpenURL version 0.1, and

```
setenv sfx_openurl_version    ver-1.0
```

stands for OpenURL version 1.0

Depending on the OpenURL version defined in the `aleph_start` file, the system loads Web files from `/alephe/www_f_eng/` with the corresponding suffix: `sfx-record-ver-1.0` or `sfx-record-ver-0.1`. These files control the building of the OpenURL according to the standards of the version it refers to. If the file with the corresponding suffix does not exist, or the parameter is not present in `aleph_start`, the system loads the `sfx-record` Web page.

The `tab_sfx_denied` configuration table defines when the SFX button is not displayed in the Web OPAC. The same table under `usm30/tab` controls the suppression of the SFX button in the Course Reading module.

The `ext_srv.ini` file under the `\Catalog\tab` or `\ACQ\tab` directories contains a list of URLs with their captions to appear in the dynamic menu of external services in general and in SFX in particular. The menu is only available when there is an active document / order in the module.

The system provides a link to the SFX Menu.

The URL contains placeholders for the active library, active doc number and cataloger and order-sequence (relevant only for ACQ module).

The field from which the ISSN component of the SFX link will be constructed is defined in the aleph_start sfx_issn_routine parameter. This parameter has the following configuration options:

00 – Extract the last 022 $$a

01 – Extract the last 022 $$y if the default routine does not find any ISSN.

21 Tomcat Server

The Apache Tomcat Application Server is supported in ALEPH.

22 Global System Status

You can enable or disable the various elements of the system including the GUI modules, the batch queue and the system daemons. This can be useful if there is a need to stop all or part of the system, for example for maintenance purposes.

This is controlled via the UTIL W/5 utility "View/Update Global System status". When invoked, it displays a list such as the following:

```
1. SYSTEM               UP
2. WWW-SERVER           UP
3. ACQ                  UP
4. CIRC                 UP
```

```
5. CAT                  UP
6. ILL                  UP
7. JOBD                 UP
8. QUEUE-BATCH          UP
9. ALL-UE               UP
10. UE-01               UP
11. UE-03               UP
12. UE-06               UP
13. UE-08               UP
14. UE-11               UP
15. UE-13               UP
```

"UP" indicates that the system or the specified part of it is active; "DOWN" indicates that the system or the specified part of it is inactive.

The SYSTEM and the ALL-UE options are general options that control a few functions. The ALL-UE option disables or enables all of the UE daemons. If the ALL-UE function is DOWN, then all UE daemons, such as UE-06, will be disabled. It is still possible to disable only a specific UE by leaving the ALL-UE option on UP and toggling the specific UE daemon, such as UE-06, to DOWN. You can use the SYSTEM option to disable and enable all of the system's servers and daemons. WWW-SERVER and JOBD will enable or disable the www_server and the job_daemon. If set to "DOWN", the ACQ, CIRC, CAT and ILL options will block any operations originated by the Acquisitions, Circulation, Cataloging or ILL GUIs.

Note that the list that displays when UTIL W/5 is invoked is a global list and affects all the libraries that are defined in the system. It is possible to control some of the system elements (the UE procedures and batch queue) at the level of a specific library. This can be done by entering an L to "View Library System status". The list of the active library will be displayed. You will then see a list such as:

```
USM50 System Status
Process Name                        Status
---------------------------         ----------------------------
1. QUEUE-BATCH                      UP
2. ALL-UE                           UP
3. UE-01                            UP
4. UE-03                            UP
5. UE-06                            UP
6. UE-08                            UP
7. UE-11                            UP
8. UE-13                            UP
```

On this list you toggle only USM50 elements. For example, disabling UE-06 will disable only UE-06 of USM50.

The system status is saved in the `$alephe_tab/tab_system_status` table. This table should not be edited manually.

23 Customizing Shortcut Key Assignments

You can customize shortcut keys in ALEPH by assigning shortcut keys to commands that do not already have them or by removing shortcut keys that you do not want. If you do not like the changes you make, you can return to the default shortcut key settings at any time in the present session. However, once you close a session, any changes you have made in shortcut key assignments will be saved in `accel.dat`.

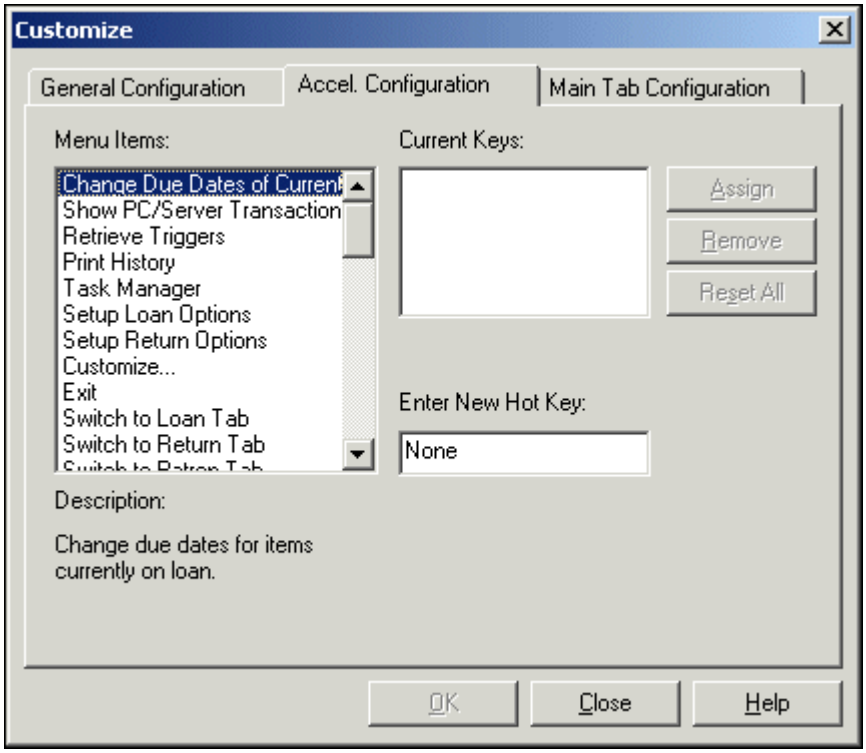
The following is true for all ALEPH GUIs:

Do not use the following shortcut keys: Ctrl+H, Ctrl+I, Ctrl+J. These shortcut keys are reserved for Windows functions and cannot be reassigned.

The Shift key can only be used together with the virtual keys F1-F12.

To customize the shortcuts available in the GUI:

From the ALEPH menu, point to Options, click **Customize**, and select the Accel. Configuration tab:



- To see which shortcut keys are currently assigned, highlight an item in the **Menu Items** list by clicking on it. The shortcut key currently assigned to the selected item is displayed in the **Current Keys** field.

- To assign a new shortcut key, after selecting a menu item, press a combination of keys (for example, Ctrl+F3) and click **Assign**.

- To remove a shortcut key, select the shortcut key from the **Current Keys** field and click **Remove**.
- To reset all of the shortcut key assignments to the settings present at the beginning of the current session, click **Reset All**.

Any changes you make via the Accel. Configuration tab are automatically saved in a text file, **accel.dat**, located in the module's tab/<lng> directory on the client, and vice versa. All menu options that are listed in the menu.dat file can be listed in the accel.dat file along with the keys (or key combinations) to be used to activate the menu option.

The key combinations must follow these rules:

- If the Ctrl, Alt or Shift key is to be used, type it first. (You can also type any combination of these keys, for example, **Ctrl+Alt**.)

- To indicate that these keys must be held down while pressing another key, type a plus sign (+), then type the ASCII character (in square brackets []) or the virtual key that must be pressed. A virtual key is of the type VK_* (for example, VK_F3).

Examples:

Ctrl+VK_F3 (this is displayed in the GUI menu as **Ctrl+F3**)
Alt+[V] (this is displayed in the GUI menu as **Alt+V**)

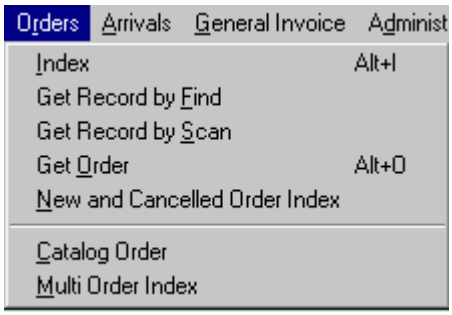
The following examples show key definitions in accel.dat in the Circulation module, together with the name of the function as assigned by the programmer.

```
!                        Accel Texts
!
!                ID                                  Data
!                40                                   255
!-------------------------------------!-----------------------------
------------>
SWITCH_APPLICATION                     Ctrl+[A]
NEW_USER                               Ctrl+VK_F7
USER_LIST                              Alt+VK_F7
LOCATE                                 Alt+VK_F8
```

The `menu.dat` file contains the name of the function as assigned by the programmer and the text that appears in the menu. For example:

```
!                        Menu Texts
!
!                ID                                  Data
!                40                                   255
!-------------------------------------!-----------------------------
------------>
ALEPH                                  &ALEPH
DATE                                   &Dates
LOAN_RENEW                             Renew &Loan by Barcode
LOAN                                   &Circulation
.../...
```

In the application, the system displays the menu text as defined in `menu.dat` together with the shortcut key definition defined in `accel.dat`:



24 Setting up Column Headings in GUI Lists

`pc_tab_col.lng` defines the columns of information that are displayed in list windows in the GUI clients.

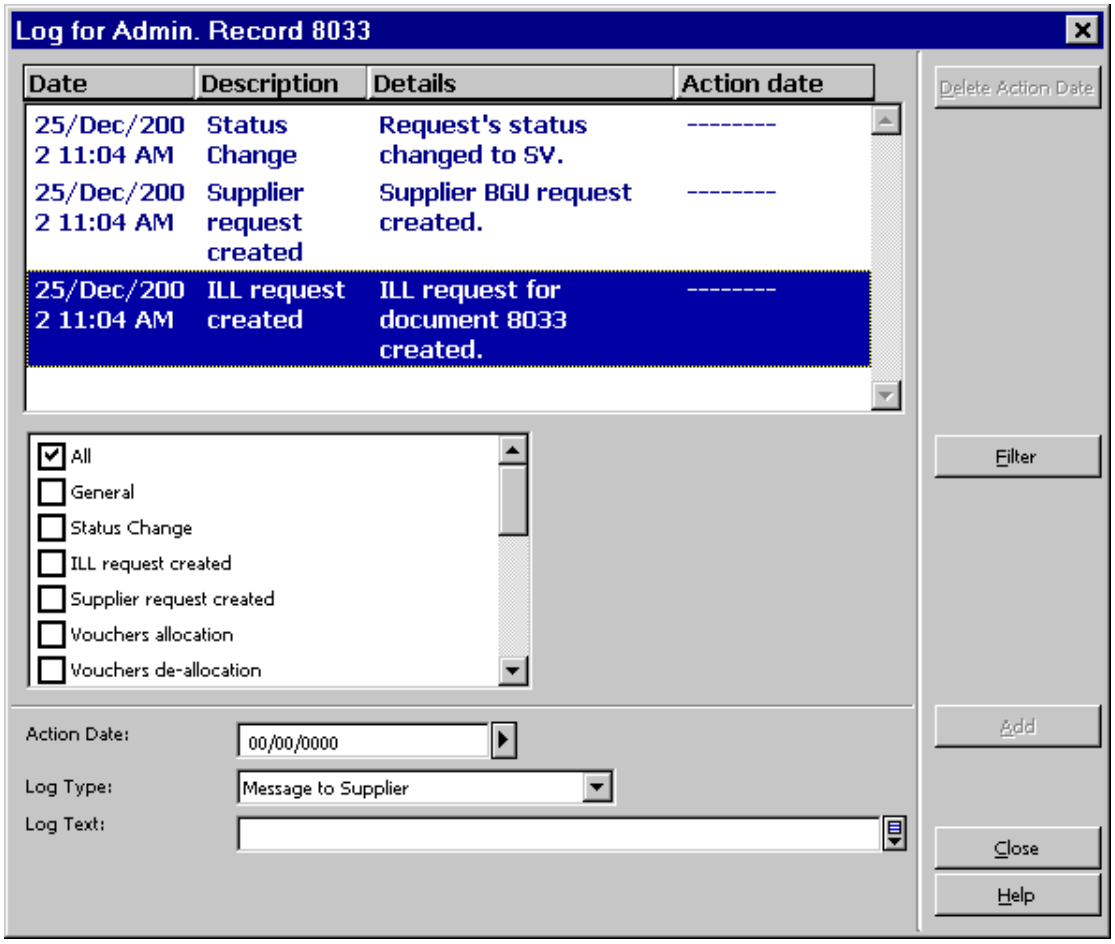
Each list is assigned a unique identifier (column 1), and each column is assigned a column number (column 4). These two elements (column 1 and column 4) must NOT be changed. The display order of the columns is set by the order of the lines in this table.

The default width of the column is defined as a percentage of the line length (column 5). The percentages of the displayed columns should not exceed 100. A column is set for non-display by setting its percentage to 000.

Note
Some list window columns are not controlled by this table. They are controlled by the \alephcom\tab\lng\tab_col.lng table on the GUI client.

Following is an example set of column definitions in pc_tab_col.lng for the Log for Admin Record List in ILL, and the corresponding GUI display:

```
1        2   3             4   5  6   7    8   9     10

PC_ILL_LOGGER  L Date           01 025 01 C04           Date+time
PC_ILL_LOGGER  L Description    02 025 01 C04           Z45-TEXT
PC_ILL_LOGGER  L Details        03 050 01 C04           Z45-DATA
```



Column 1 is the unique code by which the system identifies the set of column headings.
For a list of each module's identifiers and corresponding windows, see the "Column Headings (pc_tab_col.lng)" section of the module's System Librarian chapter.

Column 2 is the code for the character set of the column heading. The standard is L, which stands for Latin.

Column 3 is the text of the column heading that is displayed in the GUI.

Column 4 is the column number (from left to right).

Column 5 is the percentage of the page width that you want the column to take up. The number must be 3 digits. For numbers less than 100, enter leading zeros. For example, *025*. You may remove a column heading from the table by entering *000*. Do not delete any line from the table.

Column 6 is the 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.lng` setup, by using `ListBox##`. When a bitmap is used (for example, a checkmark) the characters *it* should be entered here.

Column 7 is the color code. The color value is defined in the client's

`Alephcom\tab\alephcom.ini` file, in the `[TextListBox]` section. When a bitmap is used (for example, a checkmark) the characters *map* should be entered here.

Column 8 is font 2 for font differentiation. This is only relevant in instances where the line has a system differentiation, for example, arrived/non-arrived issue.

Column 9 is color 2 for color differentiation. This is only relevant in instances where the line has a system differentiation, for example, arrived/non-arrived issue.

Column 10 is a note that is NOT displayed on the screen.

24.1 GUI Lists - Addition of Columns Missing from pc_tab_col.lng

When a GUI list contains more columns than those listed under the appropriate section in `pc_tab_col.lng` (including those with 000 percentage in column 5 of `pc_tab_col.lng`), the missing columns are all headed "Not in Table"; they can be activated by right-clicking on the list header and selecting their check boxes. In any event, these columns are always included in the callout bubble displayed by right-clicking on a line in the list.

This solution is useful for cases in which the program responsible for creating a GUI list generates columns which are not listed in `pc_tab_col.lng` for some reason. Thus, the "Not in Table" columns, created on the fly, provide instant indication of a mismatch between the program and `pc_tab_col.lng`, which can be resolved forthwith.

25 Customizing Shortcut Buttons

Each toolbar has a left side and a right side. The left side contains buttons that represent the toolbar's operational context (for example, Items, Patron, Loan, Return). The right side contains buttons that can be used as shortcuts instead of menu commands. You can customize this bar to:

- Add icons - add a definition to `tbarbmp.dat`
- Delete icons - delete a definition from `tbarbmp.dat`
- Modify icons - enter a different bmp file name, or update an existing BMP file.

These icons are Bitmap files that are stored under `alephcom/tbarbmp` The file that describes each bar is alephcom/tab/tbarbmp.dat

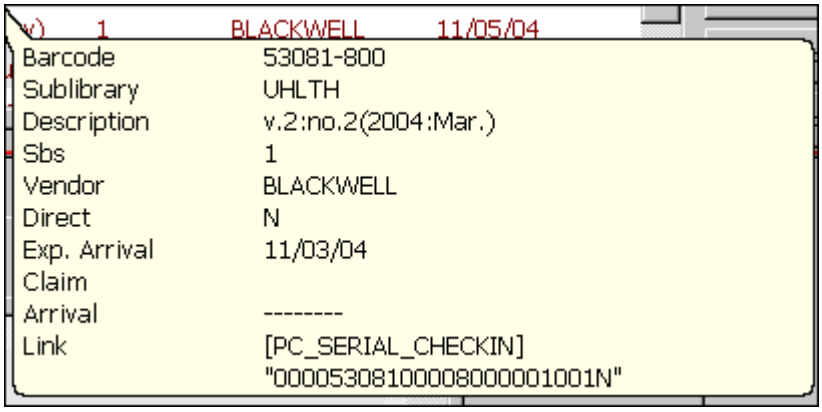
The toolbar context strings bars are:

```
Acquisitions
------------
"OrderBar"
"InvoiceBar"
"SerialBar"
```

```
"OrderBarCommands"
"InvoiceBarCommands"
"SerialBarCommands"

Circulation
-----------
"UserBar"
"ItemBar"
"UserBarCommands"
"CircItemBarCommands"

Cataloging
----------
"RecordBar"
"RecordBarCommands"
"CatalogItemBar"
"CatalogItemBarCommands"
```

26 Displaying the "Link" Field in the Callout Bubble in GUI Lists

You can display the "Link" field at the bottom of the callout bubble invoked by right-clicking on a line in a GUI list.



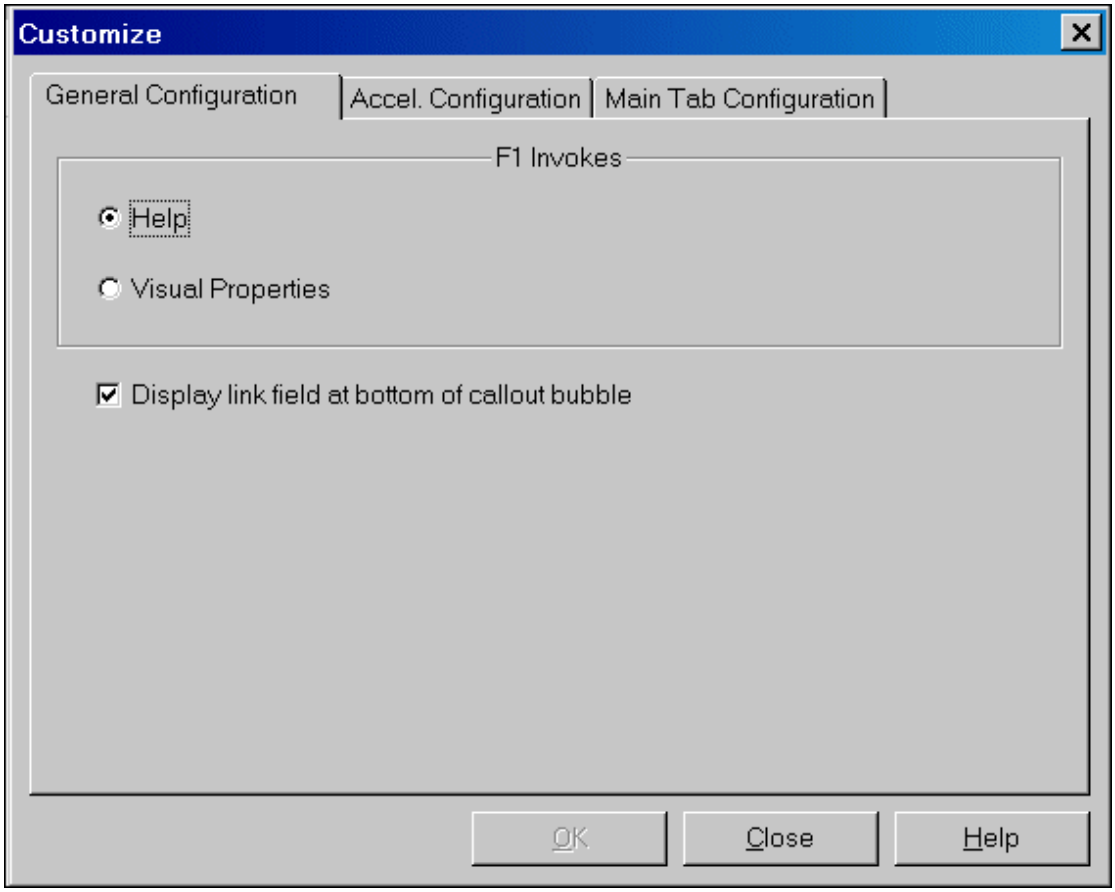
In the majority of GUI lists, the "Link" field is always attached to the end of a line.

The "Link" field contains the name of the section in `./usm01/tab/pc_tab_col.lng` 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...)



Here is an example of a "Link" definition in `.\Alephcom\Tab\Eng\window.dat`:

```
GeneralConfigDlg.Check.ShowLinkInBubble  Display link field at bottom
of callout bubble
```

The following is an example of the text that might appear in a "Link" field:

```
Link              [PC_ACQ_LIST] "000001650000015OMRI
010                  NNNM00005"
```

In the example above, the section in `pc_tab_col.lng` 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:                   OMRI
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
```

Displaying the "Link" field enables librarians to identify the list in `pc_tab_col.lng`; it is also useful to programmers for debugging purposes.

27 Setting up GUI Toolbars

Toolbar is the standard term for the various rows in the GUI below the menu bar that contain fields and icons for commonly performed tasks.

27.1 Display Text

You can determine the content and formatting of the text that displays in the toolbars in the client by setting up lines in the following files:

- `$alephe_root/error_lng/pc_acq_c0506` - used for the Order Bar in the Acquisitions/Serials client
- `$alephe_root/error_lng/pc_serial_c0339` - used for the Serials Bar in the Acquisitions/Serials client
- `$alephe_root/error_lng/pc_cir_c0461` - used for the Items Bar in the Circulation client
- `$alephe_root/error_lng/pc_com_c0174` - used for the Items in the Cataloging client
- `$alephe_root/error_lng/pc_cir_c0463` - used for the Patron Bar in the Circulation client

The first four toolbars above share the same convention: $1=title, $2=author, $3=BIB number, $4=ADM number, $5=Year.

For example, let us assume that the following setup appears in `$alephe_root/error_lng/pc_acq_c0506`:

```
2001 0000 L BIB=$3;ADM=$4 - $1 ($2). Year: $5
```

This configuration produces the following display in the Order bar in the Acquisitions/Serials client:

The Patron Bar in the Circulation client follows a different convention: $1=name, $2=title, $3=ID and $4=barcode

For example, let us assume that the following setup appears in `$alephe_root/error_lng/pc_cir_c0463`

```
2001 0000 L $1 ($3/$4)
```

This configuration produces the following display in the Patron Bar in the Circulation client:

The maximum number of characters that can be displayed for each placeholder is 40.

Note that the color of the text is set under the ToolBar name in *.ini files and is

always called `FgColorDescript`. For example, in `circ.ini`, the following line defines the Patron Information that displays in the Patron Bar:

```
[UserBar]
FgColorDescript=000,000,255
```

27.2 Editing Toolbar Tooltips

In the Acquisitions/Serials, Cataloging and Circulation modules, a ToolTip appears when you point to the icon on the left of a toolbar:

You can edit the ToolTip's text in `.\<module>\Tab\ENG\window.dat`:

```
<BarName>.ToolTip.Info
```

For example, in `C:\ALEPH 500\Catalog\Tab\Eng\window.dat`, the line:

```
RecordBar.ToolTip.Info                    Record Bar
```

supports the following ToolTip:

The ToolTip that appears when you point to the button in the above toolbar is determined by:

```
RecordBar.ToolTip.Go                      Load Record from Server
```

27.3 Identifying Icons

There are non-clickable icons on the far left of each toolbar. These are used to indicate the toolbars functionality (Serials, Items, and so on). These icons are not customizable.

28 Cash Amounts - Suppress the Display of Decimal Point and the Digits Afterwards

When setting up system variables, Aleph enables the reporting and displaying of currency in any other pre-defined currency without a decimal point and the digits afterwards.

Two variables determine the format in which Aleph displays or prints amounts from the GUI, the OPAC, and printed forms.

This feature influences only the way the user interface displays the amount of the currency, the actual amount of currency in the Aleph database contains the decimal point and the digits afterwards.

The suppress decimal point feature affects most of the times that currency amounts are reported in the GUI and OPAC, printed in online and batch reports, slips, and letters, and displayed in exported Aleph data X-services and DLF-API.

It is not available in GUI forms (in which the user enters an amount and the system calculates and displays a result). See **Example 1: GUI – Acquisitions - General Invoice – List of Invoiced Line Items** on page 55.

In addition to the GUI forms, there are other examples in which the suppress decimal point feature is not available.

Aleph System Variables for Suppress Decimal Point

In order to activate this feature; configure the following two variables:

- SUPPRESS-DECIMAL-DISP variable in `./alephe/tab/tab100`
  This tab100 flag defines the currencies for which the suppress decimal display feature is activated.

- Translate routine to be set in *,trn files: SP-DECIMAL.
  This variable works in conjunction with tab100 SUPPRESS-DECIMAL-DISP. The SP-DECIMAL translate routine determines the amount displayed in:

  - Reports, slips, and letters (online and batch)
  - GUI read-only information screens (HTML view via GUI).
  - Exported Aleph data (X-services, DLF-API)

28.1 Suppress Decimal in GUI and OPAC Screens – Set Up and Functionality

The tab100 variable, SUPPRESS-DECIMAL-DISP, defines the currencies for which the suppress decimal point feature is activated in the GUI and OPAC.

This variable is also consulted when SP-DECIMAL translate routine is activated.

28.1.1 Setting Up - tab100: SUPPRESS-DECIMAL-DISP:

SUPPRESS-DECIMAL-DISP flag is set in tab100.

Set this variable for all currencies (separated by a comma) for which you want the decimal point and the two digits after it to be dropped when displaying their amounts.

For example:

```
./alephe/tab/tab100:
SUPPRESS-DECIMAL-DISP=JPY,KRW,VND
```

The above example supports the suppress decimal point feature for currency in JPY (Japanese Yen), KRW (Korean Won), and VND (Vietnamese Dong).

Amounts that represent other currencies that are not defined in SUPPRESS-DECIMAL-DISP (for example, USD or EUR) are displayed with a decimal point and the two digits after it.

Up to 10 different currencies can be set in SUPPRESS-DECIMAL-DISP, delimited by a single space. The currency codes must match the codes defined in the Aleph Currency table that are set via GUI-ACQ-Administration tab-Currency.

In case SUPPRESS-DECIMAL-DISP is not set (or no currencies are listed), the suppress decimal point feature is not activated.

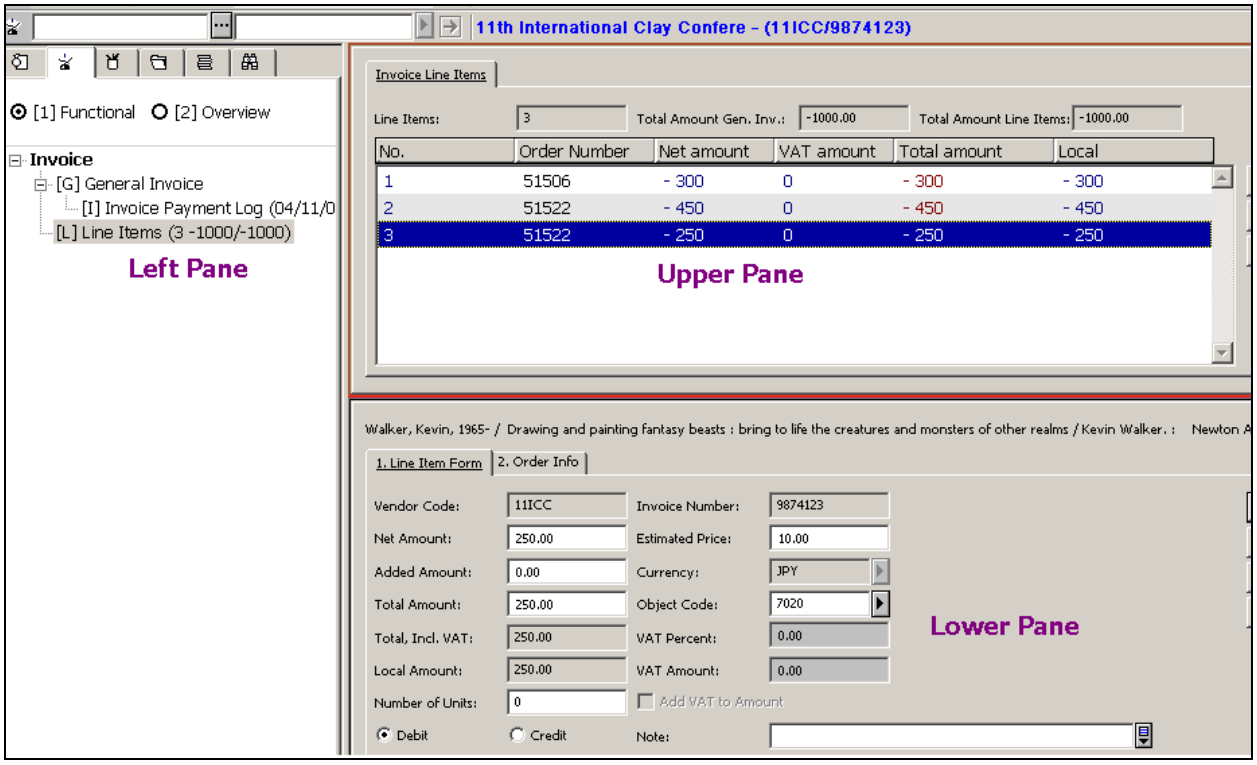
28.1.2 Functionality - 'Suppress Decimal' in GUI and OPAC Screen

When activating tab100 SUPPRESS-DECIMAL-DISP the amount displayed for the defined currencies in various GUI and OPAC pages are without a decimal point and the digits afterwards.

GUI forms (in which user enters an amount and the system calculates and displays a result) are not included in this feature. The amounts are always displayed with a decimal point and the digits afterwards.

The following are examples of an Aleph UI amount display with SUPPRESS-DECIMAL-DISP activated.

Example 1: GUI – Acquisitions - General Invoice – List of Invoiced Line Items



The figure above displays a line item list of a general invoice whose currency is in JPY - Japanese Yen.

Left-Pane – The left pane displays a summary of three invoice line item sum and the total general invoice sum. The amount display format is determined by SUPPRESS-DECIMAL-DISP so that the sum is displayed as 1000 rather than 1000.25 or 1000.00.

For a USD (US Dollar) invoice (or any other currency not listed in SUPPRESS-DECIMAL-DISP), the amounts are displayed with a decimal point and two digits afterwards.

Upper pane – The upper pane lists each of the debited line items. The amount display format is determined by SUPPRESS-DECIMAL-DISP so that the sum in the first line is displayed as 300 and not 300.00 or 300.15.

The suppress decimal point feature does not handle the sums displayed at the upper bar of the upper pane (the Total amount Gen. Inv and Total Amount Line Items read-only fields).

For a USD (US Dollar) invoice (or any other currency that is not defined in SUPPRESS-DECIMAL-DISP), the amounts, except for that of the Local Amount, are displayed with a decimal point and the two digits afterwards.

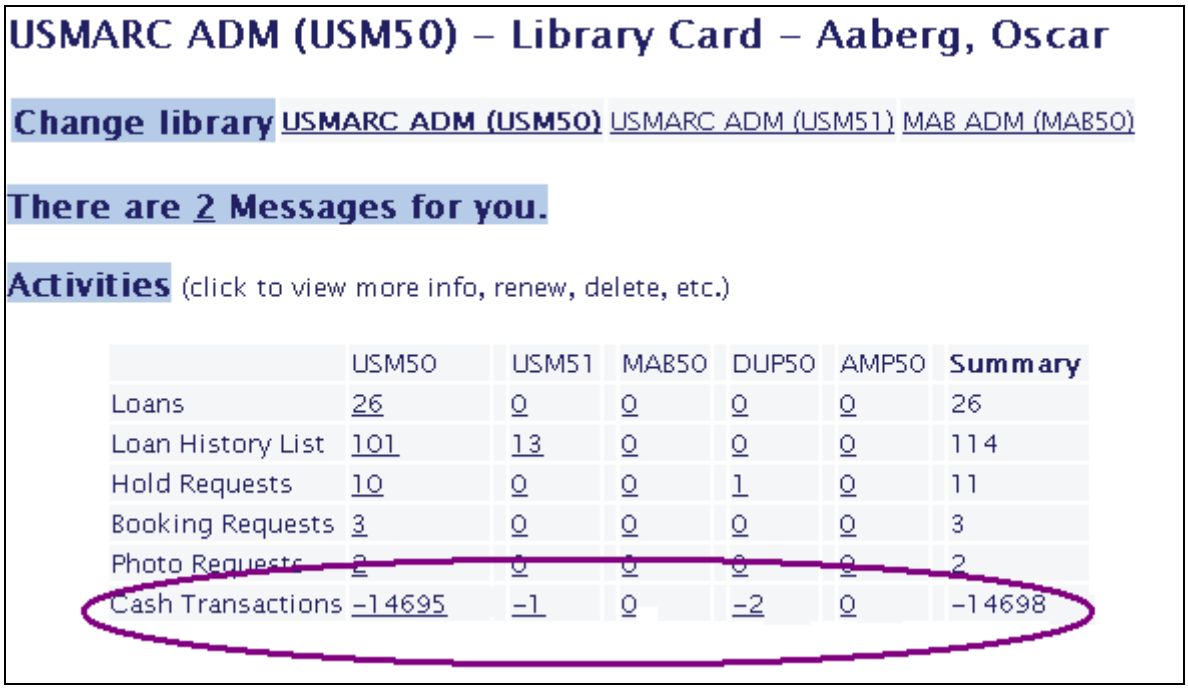
The Local Amount display consults the local currency code (`local_currency` variable of `aleph_start`). When the local currency is JPY (or any other currency listed in SUPPRESS-DECIMAL-DISP), the amount is displayed without a decimal point and the two digits afterwards (for USD Invoice as well as for JPY Invoice).

Lower-pane – The lower-pane shows the GUI Line Item Invoice form (details of the third line that is focused on the upper pane). The amounts data in this form are editable and are not handled by SUPPRESS-DECIMAL-DISP. All amounts are displayed with a decimal point and the two digits afterwards.

The same principles for displaying amounts are applied in the GUI module, this includes:

- GUI-Circulation – Patron Cash List
- GUI – Acquisitions – Order List
- GUI-Acquisitions – Order Encumber window
- GUI Acquisitions – Budgets transaction List
- GUI- Order/Subscription/Circulation/ILL Log

Example 2: OPAC – Library Card – Summary of Patron Activities and Cash Transaction List



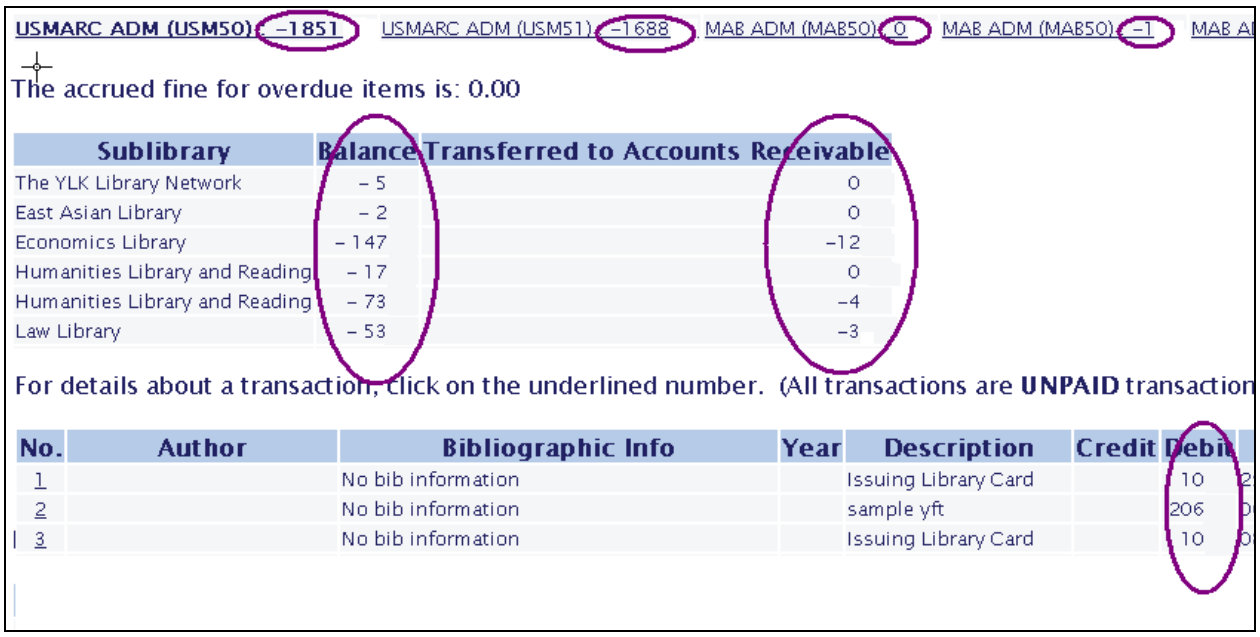
USMARC ADM (USM50) – Library Card – Aaberg, Oscar

Change library USMARC ADM (USM50) USMARC ADM (USM51) MAB ADM (MAB50)

There are 2 Messages for you.

Activities (click to view more info, renew, delete, etc.)

	USM50	USM51	MAB50	DUP50	AMP50	Summary
Loans	26	0	0	0	0	26
Loan History List	101	13	0	0	0	114
Hold Requests	10	0	0	1	0	11
Booking Requests	3	0	0	0	0	3
Photo Requests	2	0	0	0	0	2
Cash Transactions	-14695	-1	0	-2	0	-14698



USMARC ADM (USM50) -1851 USMARC ADM (USM51) -1688 MAB ADM (MAB50) 0 MAB ADM (MAB50) -1 MAB AI

The accrued fine for overdue items is: 0.00

Sublibrary	Balance	Transferred to Accounts Receivable
The YLK Library Network	- 5	0
East Asian Library	- 2	0
Economics Library	- 147	-12
Humanities Library and Reading	- 17	0
Humanities Library and Reading	- 73	-4
Law Library	- 53	-3

For details about a transaction, click on the underlined number. (All transactions are **UNPAID** transaction

No.	Author	Bibliographic Info	Year	Description	Credit	Debit
1		No bib information		Issuing Library Card		10
2		No bib information		sample yft		206
3		No bib information		Issuing Library Card		10

Taking into account the local currency code and SUPPRESS-DECIMAL-DISP; the amounts are displayed without a decimal point and the two digits afterwards.

The amount is displayed as -14695 and not -14695.00.or -14695.35.

28.2 Suppress Decimal in Aleph Printouts - Set Up and Functionality

This section includes a description of the setup and functionality for the suppress decimal point feature in Aleph printouts. In this section, the term Aleph Printouts refers to:

- Reports, slips, and letter (online and batch)
- GUI read-only information screens (HTML view via GUI)
- Exported Aleph data (X-services, DLF-API)

The Aleph translate mechanism allows the editing of Aleph data for display purposes. This includes the data shown in the above examples.

Libraries may set the required translate routines in the Aleph configuration tables.

To support the suppress decimal point feature in the above Aleph printouts, the SP-DECIMAL routine must be set in the `.trn` files.

The SP-DECIMAL routine works in conjunction with the tab100 SUPPRESS-DECIMAL-DISP variable.

Upon setting up the currencies in SUPPRESS-DECIMAL-DISP and defining the SP-DECIMAL translate routine, the printed amount of the predefined currencies is displayed without a decimal point and the two digits afterwards.

All other translate mechanisms that can be set for numeric data (for example, INTEGER, ADD-SIGN, and SIGN1) stays as is and may be activated as well (for more details see `./usm01/form_eng/ translate_readme.txt`).

28.2.1 Setting Up - Translate Routines for Aleph Printouts

The following is a brief explanation of the translate routine mechanism. The SP-DECIMAL is applied similarly for all the translate routines:

Translate Routines for Online and Batch Reports/Slips/Letters

The translate routines for online and batch reports, slips, and letters (XSL files) are set in the library's `./form_<lng>` directory.

Libraries may set translate routines for a specific XSL printout file (for example, `cash-user-letter.trn` for `cash-user-letter.xsl`) or for all XSL printouts in the `global.trn` file. Translate routines that are set in `global.trn` file activate the translate routines for all printouts.

Translate Routines for GUI HTML

The translate routines for GUI HTML screens are set in the library's `./pc_display_<lng>` directory.

Libraries may set translate routines for a specific HTML file (for example, `user-cash.trn` for user-cash) or for all GUI HTML screens in the `global.trn` file. Translate routines that are set in the `global.trn` file activate the translate routines for all GUI HTML view screens.

If no translate routines are set in `library's ./pc_display_<lng>`, the system looks for translate routines in library's `./form_<lng>` directory.

Translate Routines for X-Services

The translate routines for X-Services data are set in the directory `./alephe/www_x_<lng>`. Libraries may set translate routines for a specific X-services or define translate routines in the `global.trn` file.

Translate Routines for DLF Services

The translate routines for DLF services are set in the directory `./alephe/dlf_<lng>`. Libraries may set translate routines for a specific DLF service or define translate routines in the `dlf-global.trn` file.

Implementing SP-DECIMAL in trn Files

Set the SP-DECIMAL translates routine in all relevant `.trn` files, as described below:

For the relevant tag (column 1 of the `.trn` file) libraries must define DECIMAL-SP translate type (column 2) and a parameter (column 4) that indicate the data element of the currency code.

For example:

```
./usm01/form_eng/global.trn
Col 1. - tag
Col.2 - translate type
Col.3 - S/M (Single or Multi occurrence)
Col. 4 - Parameter
       1                                  2      3        4
!!!
```

```
z411-e-local-price                  SP-DECIMAL S LOCAL
Z411-s-price                        SP-DECIMAL S z411-s-currency
z411-s-local-price                  SP-DECIMAL S LOCAL
```

Column 2 is set with SP-DECIMAL for tags (column 1) that represents amounts of:

- Acquisition Order (Z68)
- Budget Transaction (z601)
- Patron's Cash Transaction sums (Z31)
- General Invoice (Z77)
- Invoice Line items (Z75)
- ILL Borrowing and lending requests (Z411 and Z416)

Column 4 is set with the data tags that point to the currency code. This parameter is essential in order to ensure that the suppress decimal point feature is activated only for currencies defined in tab100 SUPPRESS-DECIMAL-DISP.

In this way, USD amounts are displayed with a decimal point and the two digits afterwards and JPY amounts are displayed without a decimal point and the two digits afterwards. Therefore, for currencies that are not defined in SUPPRESS-DECIMAL-DISP, the SP-DECIMAL works as a regular DECIMAL translate routine (amounts are displayed with a decimal point and the two digits afterwards).

Column 4 must be set for all for DECIMAL-SP tags.

For Z31 tags – use 'LOCAL' in col.4. The system always consults the local currency code (the l`ocal_currency` variable of `aleph_start`). If the local currency is JPY (or any other currency listed in SUPPRESS-DECIMAL-DISP), the amount are displayed without a decimal point and the two digits afterwards. Otherwise, the SP-DECIMAL works as regular DECIMAL translate routine (amounts are displayed with a decimal point and the two digits afterwards).

Set LOCAL in column 4 for other tags that represent local amount (for example, z601-local-sum, z416-local-price, and z411-e-local-price).

For Z68, Z77, and other entities that contain the currency element, use z68-e-currency, z77-i-currency, etc.

For the line item invoice tag, Z75 (which does not contain currency within the Z75 entity) set the line item currency in col. 4. (This is a special tag that generates the line item currency from the related general invoice Z77 entity).

If a user mistakenly leaves column 4 empty, the suppress decimal point feature works as a regular DECIMAL translate routine (amounts are displayed with a decimal point and the two digits afterwards).

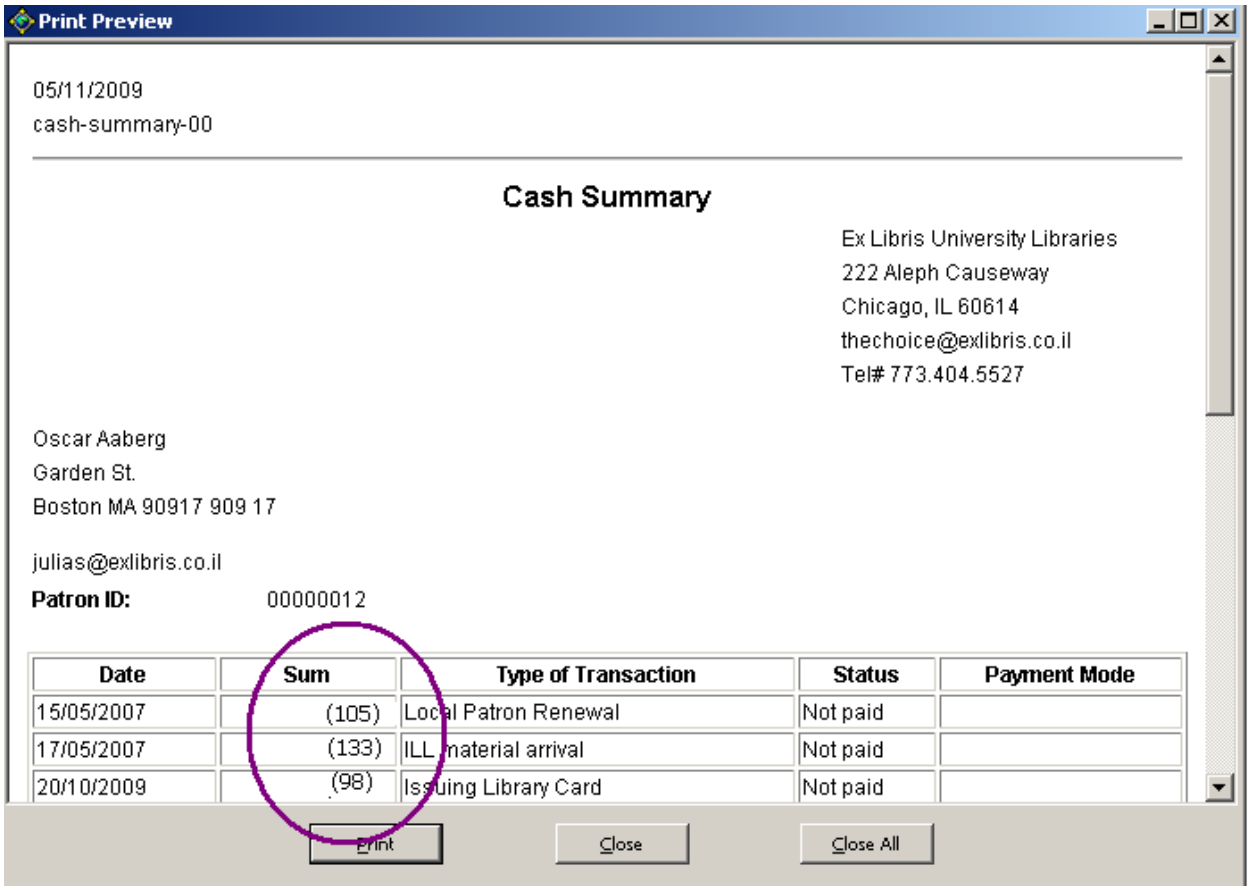
Do not define DECIMAL and SP-DECIMAL for the same tag. If a user mistakenly sets both routines to the same tag, the system applies the first routine that comes up.

If SP-DECIMAL is set in the `.trn` files but tab100 SUPPRESS-DECIMAL-DISP is left empty (no currencies are defined for the suppress decimal point feature), SP-DECIMAL works like a regular DECIMAL translate routine (amounts are displayed with a decimal point and the two digits afterwards).

28.2.2 Functionality - Suppress Decimal in Data Printouts

Upon setting up `.trn` files with SP-DECIMAL, the amounts reported in data printouts for the currencies listed in tab100: SUPPRESS-DECIMAL-DISP are displayed without a decimal point and the two digits afterwards.

Example 1: GUI Online Printout – Cash Summary

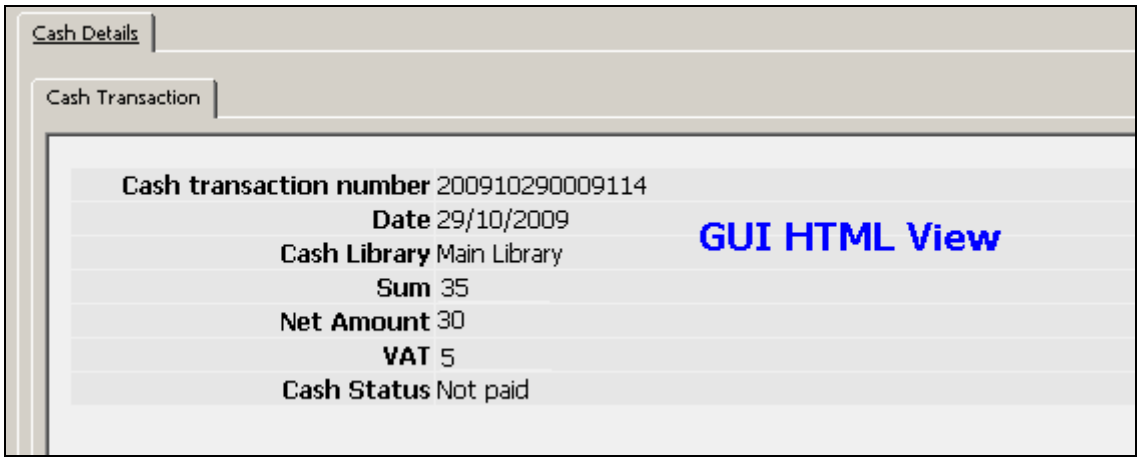


The sum amounts are displayed without a decimal point and the digits afterwards.

The same format of the amount display is applied to other printouts, this includes:

- Online reports produced by the user via the GUI
- Slips and letters produced by the user via the GUI (can be sent by e-mail or print)
- Batch services reports, slips, and letters

Example 2: GUI –Circulation - HTML View – Cash Transaction Details

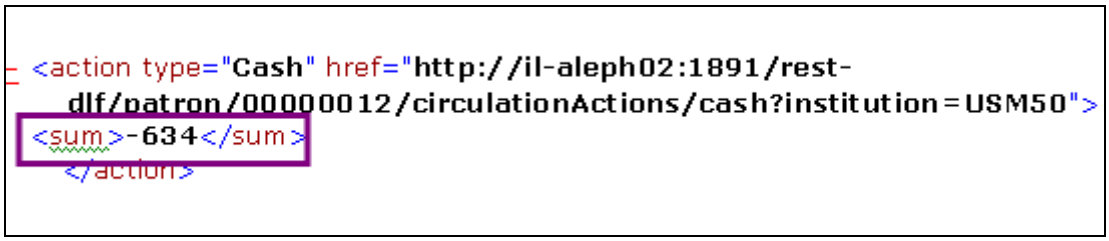


The sum amounts are displayed without a decimal point and the digits afterwards.

The same format of the amount display is applied in other GUI HTML views, this includes:

- GUI-Acquisitions-Order Details
- GUI-Acquisitions-Budget Balance
- GUI-Acquisitions-Order's Invoice and Encumber details

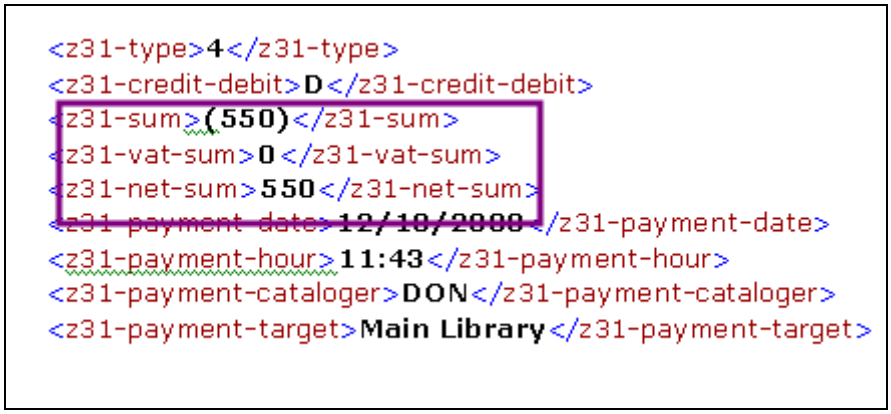
Example 3: Exporting Aleph Data - DLF – Patron's Information - Cash sum



The sum amounts are displayed without a decimal point and the digits afterwards.

The same format of the amount display is applied for other DLF services that export amount data.

Example 4: Exporting Aleph Data – X-Services – Patron's information - Cash Sum



The sum amounts are displayed without a decimal point and the digits afterwards.

The same format of the amount display is applied for other X-Services that export amount data.

29 Preservation of the www_server and pc_server Log Files

The $KEEP_LOGDIR value in aleph_startup is set to automatically remove www_server and pc_server log files that are more than 7 days old. This is necessary because the www_server and pc_server log files can increase in size quickly and take up all of the available disk space.

Most of the information you need for OPAC statistics is recorded in the Z69 OPAC Events table, which is kept indefinitely. However, some information is not saved in the z69 table, for example, the number of times a 856 field is clicked in the Web OPAC leading to an external resource. Therefore, if you have room on your server, you might want to keep the the www_server and pc_server log files for more than 7 days. To do so, change the value of the $KEEP_LOGDIR variable in the Cleanup TMPDIR/LOGDIR section of $alephe_root/aleph_startup. Another solution is to tar or gzip the www_server and pc_server log files rather than deleting them, so that they are available to you in the future.

30 Clearing the VIR01 Library

The VIR01 library (included as part of the standard ALEPH setup) is a library that contains temporary records in temporary Oracle tables. These records are required for ongoing work of the Web OPAC and PC server.

Clean this library periodically by using the $aleph_proc/clear_vir01 script.

The clear_vir01 process cleans the data in the database tables of the VIR01 library in Cold or Hot mode.

The tables that are involved in the clear_vir01 process in Hot mode are: Z05, Z60, Z63, Z64, and Z65.

In Cold mode, all tables and sequences in the VIR01 library (Z05, Z60, Z63, Z64, Z65, Z52, Z00, Z01, Z02, Z11, Z13, Z13u, Z51, Z56, Z1xx, and Z200) are recreated and the Z52 table is initialized.

The process is invoked from the utility menu (util X 8 - Clean Virtual Library), from the prompt in the VIR01 environment ($aleph_proc/clear_vir01), or by entering a line, such as the following, in the $alephe_tab/job_list:

00 03:00:00 N clear_vir01.log csh -f /exlibris/aleph/a20_3/aleph/proc/clear_vir01 VIR01,hot,2,1000

The mode of the run must be specified. If no mode is specified, the default is Cold mode. (Online is considered Cold mode).

The parameters that the process accepts are the following:

VIR01[run mode, [no. days to retain],[count]]

The following is an explanation of the parameters:

- VIR01 – The library in which the process runs
- Mode – Hot or Cold (online)
- No. days to retain (Only applicable in Hot mode) –data that was set in the VIR01 tables earlier than the number of days to retain is deleted.
- Count (Only applicable in Hot mode) – the number of records to be deleted

Hot mode enables the clear vir01 script to run without shutting down the ALEPH process. In this mode, the script deletes data that is set in the VIR01 tables earlier than the number of days to retain as indicated in the third parameter and deletes the number of records as indicated in the fourth parameter.

Cold mode deletes all VIR01 tables without reference to any parameters and then recreates them and resets the VIR01 counters (Z52) to 0 or to predefined counters located in the $data_tab/z52.init file.

In Cold mode, all ALEPH processes need to be shutdown; otherwise, some tables might be locked by the running processes and VIR01 might not be synchronized.

Before running clear_vir01 in Cold mode, run aleph_shutdown. After it finishes, run aleph_startup.

Sometimes, when VIR01 is not synchronized, a run of clear_vir01 in Cold mode can reset the values in the VIR01 tables and synchronize the Web OPAC activity.

Some signs that VIR01 is not synchronized are unavailable Web OPAC pages (Page cannot be displayed error) or getting unpredictable results to queries sent from the browser.

It is recommend to run clear_vir01 in Hot mode every day while retaining records from the previous two days and in Cold mode once a week when the system is down (along with a cold backup or other maintenance procedures that requires you to shut down ALEPH).

If the run of clear_vir01 in Hot mode is done regularly, the run of clear_vir01 in Cold mode need only be done periodically in order to synchronize records that are not synchronized.

System Librarian's Guide - Indexing

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: December 2016

Web address: http://www.exlibrisgroup.com

Table of Contents

1 INDEXING OVERVIEW 6

2 INDEXING PROCESS 7

2.1 Defining Indexes in ALEPH 7
2.1.1 Defining a New Index 7
2.1.2 Assigning Fields to an Index 7
2.1.3 Adding an Index to pc_tab_sear.lng 8
2.1.4 Add an Index to Web Include files 8

2.2 Ongoing Indexing 8
2.2.1 Prioritizing the Ongoing Indexing 9

2.3 Building Indexes 10

3 HEADINGS INDEX 11

3.1 Filing of Headings 12
3.1.1 Normalization 13
3.1.2 Database Tables Involved 14

3.2 Main Headings Services 14
3.2.1 Update Headings Index (manage-02) 14
3.2.2 Pre-Enrich Bibliographic Headings Based on the Authority Database (manage-102) 14
3.2.3 Subject Subdivisions 14
3.2.4 When to Run the Headings Index 15

3.3 Other Headings Services 15
3.3.1 Alphabetize Headings (p_manage_16) 15
3.3.2 Alphabetize Long Headings (p_manage-17) 16
3.3.3 Build Counters for Logical Bases (manage-32) 16
3.3.4 Update Brief Records (manage-35) 17
3.3.5 Update Short Bibliographic Records (manage-07) 17
3.3.6 Create Sort Keys (manage-27) 18
3.3.7 Delete Unlinked Records (manage-15) 18

3.4 Linking Process 18

4 WORD INDEX 18

4.1 Defining Words 20

4.2 Database Tables Involved 21

4.3 When to Rebuild the Word Index 21

5 DIRECT INDEX 22

5.1 Filing Direct Indexes 23

5.2 When to Update the Direct Index 23

5.3 Database Tables Involved 24

6 APAC INDEXING 24

6.1 Headings Indexing 24

6.2 Words Indexing 26
6.2.1 Normalizing the Indexed Text 26
6.2.2 Defining Segmentation Routines for Indexing 27
6.2.3 Defining Segmentation Routines for Searching 27
6.2.4 Suggested APAC Indexing and Searching Segmentation Routines Setup 29

7 MAIN TABLES SUPPORTING INDEXING 32

8 SORTING AND WORD BREAKING 36

8.1 Sorting Headings and Indexes 36

8.2 Sorting Item Lists 41
8.2.1 Sort Options 42

8.3 Word Breaking 44

9 EXPAND ROUTINES, TABLES AND INDEXING EXPANDED FIELDS 46

9.1 Expand Record 46

9.2 Expand Routines 49

9.3 Expand-Related Tables 93
9.3.1 Configuration tables (expand_doc_type) 93
9.3.2 tab_expand_split 96
9.3.3 tab_abbrev 97
9.3.4 tab_expand_duplicate_field 99
9.3.5 tab_expand_external 100
9.3.6 expand_doc_bib_z30 100
9.3.7 expand_doc_bib_z403 101

9.4 Indexing Expand Fields (Virtual Fields) 101
9.4.1 tab_expand_extract 102
9.4.2 tab_expand_join 102
9.4.3 tab_expand_join_simple 103

10 OTHER INDEXES 104

10.1 Update Short Bibliographic Records (manage-07) 105

10.2 Update Sort Index (manage-27) 105

10.3 Update Indexes for Selected Records (manage-40) 105

11 PREPARATION FOR INDEX JOBS 105

11.1 Clean temp/scratch Directories 106

11.2 Check Oracle Space 106

11.3 Cancel Jobs Which Might Interfere 106

12 PARALLEL INDEXING 106

13 INDEXING SERVICES 112

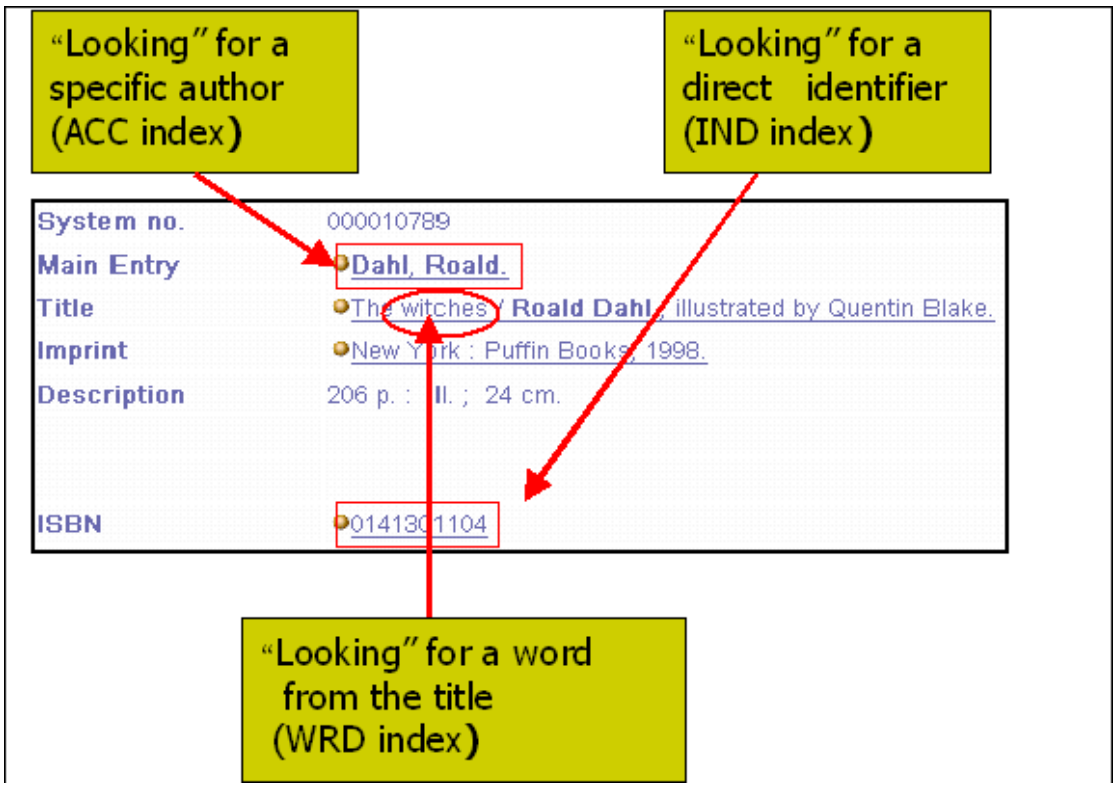
14 FURTHER READING 114

1 Indexing Overview

ALEPH allows various definitions of access paths (indexes) to document records. These definitions can be tailored to suit the needs of each application.

Indexes can be based on specific fields, subfields, on a combination of fields, or on individual words from specific document fields.

There are three major ways of accessing the bibliographic database, via three types of indexes: Headings Index, Word Index and Direct Index. Each of these types of indexes has specific subindexes attached. For example, among the subindexes attached to the Headings (ACC) index are the TIT and AUT subindexes.



Indexes are used by end users for OPAC searches and by librarians for internal needs, such as catalogers' lists and record retrieval to produce various reports.

This chapter is intended mainly for system librarians. It provides an overview of data access methods. It includes index definitions and types, index specifications, indexing processes, expand programs.

Note that the various indexing services can be accessed via the Cataloging GUI by opening the Services menu and selecting an option from the Build Indexes to Catalog submenu.

2 Indexing Process

2.1 Defining Indexes in ALEPH

There are three basic steps for defining indexes in ALEPH:

- Defining codes and names of the indexes in `tab00.lng`.
- Assigning field tags to the various index codes to define the connection between the record fields and the indexes.
- Placing the index in `tab/pc_tab_sear.lng` and in `alephe/www_f_lng/` files to enable searching of the index via the Search function in the GUIs and Web OPAC.

2.1.1 Defining a New Index

To define a new index:

1. Decide which type of index you are creating: Direct, Word or Headings.
2. Add the Index to `xxx01/tab/tab00.lng`. Look at `tab00.lng` and decide on a code for the new index (make sure the code is not already in use).

Notes

- Although the code can be up to five characters, by convention it is usually only three characters.
- If the index is a Direct index, specify "IND" for the index type.
- If the index is a Headings index, specify "ACC" for the index type.
- If the index is a Words index, find the next "W-nnn" value and use that for the index type. The -nnn numbers must be unique and consecutive.
- The Words index type should always start with the letter W.
- **The system always provides SCAN and FIND access by system number and FIND access by barcode. Therefore, although they do not need to be defined in the indexing table (tab11), they must be defined here in order to define the index name in column 11.**

2.1.2 Assigning Fields to an Index

To assign fields to an index:

1. Add the Index to `tab11`.
2. Include the code you specified in `tab00.lng` in the entry for each field in `xxx01/tab/tab11` which you want to have sent to this index.
3. If you specified "IND" in `tab00.lng`, put the entry in `tab11_ind`.
4. If you specified "ACC", put the entry in `tab11_acc`.
5. If you specified "W-nnn", put the entry in `tab11_word`.

2.1.3 Adding an Index to pc_tab_sear.lng

If you want an index to be searchable through the Search functionality in the GUIs, then add it to `xxx01/tab/pc_tab_sear.lng`.

If you want to access the index in a browse mode, include it in the "SC" (Scan) section. The index terms are displayed in alphabetical order, and you can scroll forward. Headings indexes are particularly well-suited for this mode.

If you want to be able to retrieve a set of records that match the index terms, include it in the "FI" (Find) section. WORD, SYS and BAR indexes are particularly well-suited for this mode.

2.1.4 Add an Index to Web Include files

If you want the index to be searchable in the Web OPAC in order to display an alphabetically-arranged scan list, add the index to the `scan-include-2` file in the `alephe/www_f_lng/` directory.

In order to perform the retrieval of records which are presented in a set, add the index either to the `scan-include-2` file or to the `find-code-include` file in the `alephe/www_f_lng/` directory.

2.2 Ongoing Indexing

The UE_01 background process updates the Headings Index, Word Index and the Logical Bases Counter (Z0102); the Direct Index is updated immediately.

When a cataloging record is added or updated, its system number is placed in the Z07 table. UE_01 checks the Z07 table to detect new/updated records waiting for indexing.

Once the procedure has updated the heading indexes, the Z07 record is deleted and a Z07a record is created. Once the logical bases counter is updated and the word index is created, the Z07a record is deleted. A smoothly running system should not have many records in the Z07 and Z07a tables.

The ue_01 process includes three separate sub-processes:

- ue_01_a: This process is responsible for the update of the headings (in addition it updates other and indexes and tables such as the Z13, Z00R, Z101, and so on).
- ue_01_word_parallel: This process is responsible for the word index (for performance considerations). It builds the words in parallel to other indexes.
- ue_01_z0102_index: This process is responsible for the Z0102 records (logical bases counter).

The logical bases counter (Z0102 records) was updated by the ue_08 process. Note that the performance of the ue_08 process has been greatly improved by the introduction of the ue_01_z0102_index sub-process.

The UE_01 process can be initiated in two ways:

- Invoking UTIL E/1 when the system is started up, or
- Setting the process for automatic startup in the `aleph_startup` file of the `$alephe_root` directory. If the UE_01 daemon is not running, you cannot browse or search for recently added records. However, you can look them up using their system number, which is indexed automatically in the IND index.

For big libraries experiencing backload in ongoing indexing, an improvement has been added to ue_01. The word indexing part is split into 10 processes (thus ue_01 includes 13 separate sub-processes).

In order to implement this functionality, execute the following steps:

1. Stop ue_01: `util e/2`.
2. Build 10 new temporary tables used internally by the processes:
   a. Add in `usm01/file_list` the following line:
      ```
      TAB z98t                2K              0K              TS1D
      ```
   b. Run `util a/17/1` with table name: z98t (it builds all 10 tables).
3. Build PL/SQL procedures. For instance, to run `ue_01` in USM01, run the following:
   ```
   cd $alephm_root/sql_tab/
   s+ usm01
   SQL> @z98t_proc
   SQL> exit
   ```
4. Set the flag activating the mechanism in [`bib_library`]`/prof_library`:
   ```
   setenv word_queue Y
   ```

Note:

For all other libraries (AUT, ADM, ILL and HOL libraries) that do not use this mechanism, deactivate it by setting the "word_queue" flag to "N" in `prof_library`:

```
setenv word_queue N
```

5. Execute the following command: `dlib USM01`.
6. Restart ue_01: `util e/1`.

2.2.1 Prioritizing the Ongoing Indexing

The ongoing indexing that the UE_01 background process manages, uses a set of priority rules so that some type of record updates may be indexed before others. Some of these priority rules are hardcoded, and some may be configured.

As a rule, every change in the catalog triggers a Z07 record creation. The Z07 record is the trigger for the UE_01 background process management. The lower the sequence that Z07 is assigned, the higher the record's priority is for UE_01 management. The following rules apply to setting the Z07 sequence:

- Newly cataloged records are assigned the sequence 1990, regardless of whether they are created online or offline. All new records are indexed on a first created-first indexed basis. This is a hardcoded mechanism.
- Updates in existing catalog records, regardless of whether the records are updated online or by the OCLC server, get assigned the sequence 1998. This will make new records receive a higher priority over changes in existing records (as 1990 is less than 1998). This is a hardcoded mechanism.
- When running the 'Load ALEPH Sequential MARC Records (manage-18)' batch job, the priority of the cataloged records may be set using the job's 'Override Indexing Priority' parameter. For example, if the parameter is set to 1985 then the records loaded by the batch will receive a higher priority than newly online cataloged records (as 1985 is less than 1990).
- All loading batch jobs may be assigned a priority by setting an environment variable in the $alephe_root/aleph_start.private, which sets the indexing priority for records that are created by the batch. For example:

  setenv z07_p_manage_40 2000
  setenv z07_p_manage_18 2001
  setenv z07_p_manage_180 2001
  setenv z07_p_file_90 2002
  setenv z07_p_file_93 2002
  setenv z07_p_file_95 2003
  setenv z07_p_file_96 2004
  setenv z07_p_file_97 2005
  setenv z07_p_file_98 2005
  setenv z07_p_file_99 2005

- In addition, the OCLC server and the UE_08 daemon may be set with priorities, for example:

  setenv z07_ue_03 2006
  setenv z07_oclc_server 2007
  setenv z07_ue_08 2008

Notes:
The priority that is set in the 'Override Indexing Priority' parameter of the manage-18 batch jobs overrides a value that is set in the z07_p_manage_18 variable.

The priority that is set in these environment variables overrides the hardcoded definitions listed in the previous points.

2.3 Building Indexes

Indexes are built by three batch procedures:

- p_manage_02 (Headings index)

- p_manage_01 (Word index)
- p_manage_05 (Direct index)

The batch-building procedures must be run either after conversion and/or after modifying the indexing tables (for example, `tab11`, `tab00.lng`, and so on) to update the index records.

3 Headings Index

This chapter includes the following sections:

Filing of Headings

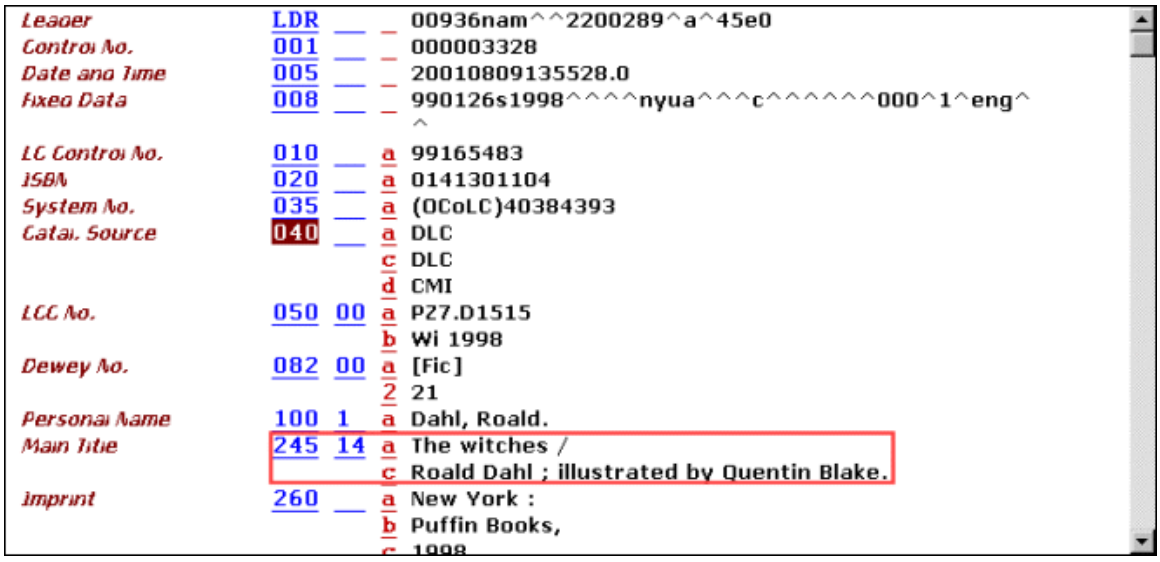
Main Headings Services

Other Headings Services

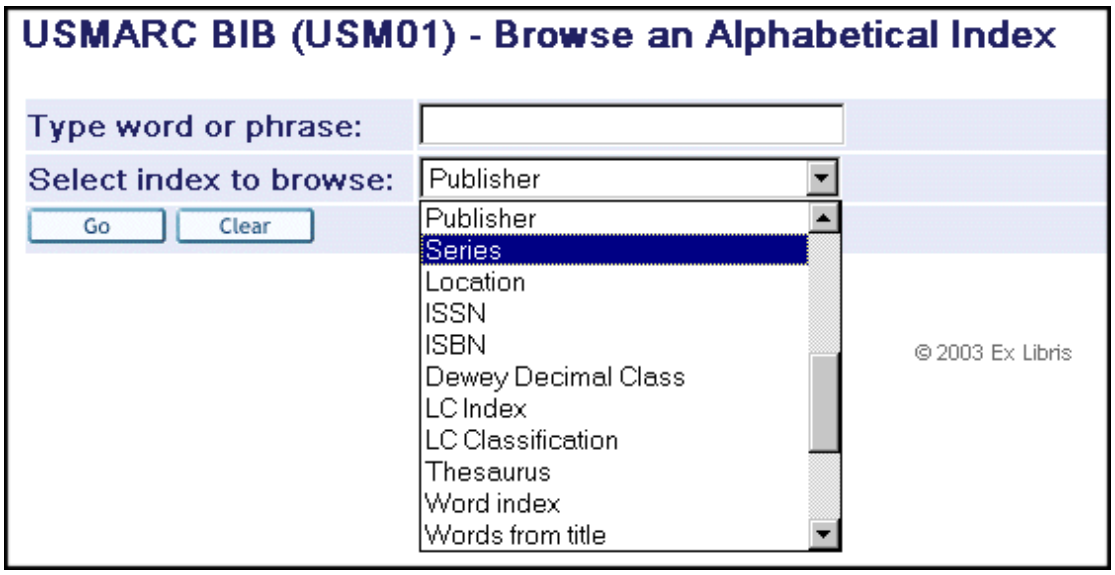
Linking Process

The Headings Index creates a list of entries that can be browsed by the patron or by a librarian in the Web OPAC. The Headings Index is also sometimes called the Browse List or ACC List.

Headings indexes are phrases from a record field such as author, title, subject, publishers, and so on. A Headings Index term can be either an entire field or one or more specific subfields.



Each library decides which fields of the bibliographic record form the basis for the Browse function in the Web OPAC. For example, you can decide to provide the ability to browse by authors, titles, publishers, and so on.



The Headings or Browse index can be accessed through the Browse function in the Web OPAC or via the Search functionality in the GUIs.

Each field or subfield is assigned to a specific headings group. For example, all types of titles can be assigned to the title headings group. Subjects can be assigned to a different "sub-index" for subjects.

3.1 Filing of Headings

Headings are filed (organized) in the browse list according to the "filing text" of the heading.

The filing text is built in three steps, based on the field text:

Display text (usually data taken from the bibliographic record without the final punctuation).

Normalization of the display text.

Normalized text to filing text.

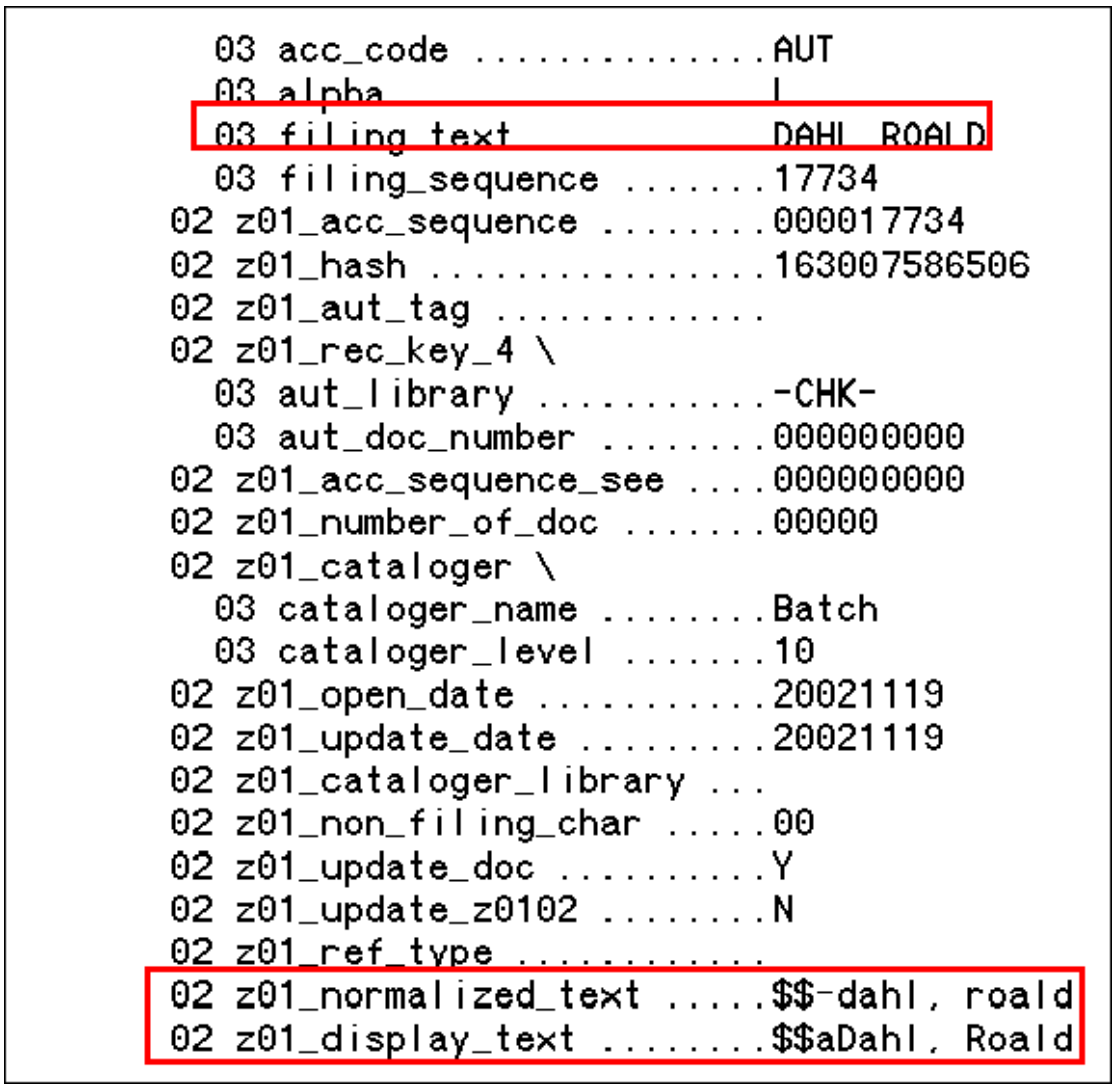
Conversion takes place according to the procedures defined in the tab_filing table of the library's tab directory and in the tab_character_conversion_line table of the `alephe/unicode` directory.

```
z01_display_text .........$$aDahl, Roald

z01_normalized_text .....$$-dahl, roald

z01 filing text ...........DAHL ROALD
```

3.1.1 Normalization

Normalization refers to the process whereby diacritics, most punctuation marks, special characters, and case differences are stripped from access fields (headings) for the purpose of determining the uniqueness of headings.

The normalized form of the headings is built according to the rules defined in the library's `tab_filing` table:



To view the whole list of filing routines and an explanation of normalization routines refer to Sorting and Word Breaking on page 36.

The aims of normalization are:

To treat the same headings alike.

To ensure that each unique heading is stored only once in the headings index.

To distinguish headings that are different by means of unique identifiers.

3.1.2 Database Tables Involved

Doc

The document table that is being indexed (in the case of a bibliographic library, this is the bibliographic record).

Z01

The Z01 table is a list of headings (entries) derived from information in the bibliographic record through which the user can browse in the Web OPAC and via the Search functionality in GUIs. The Headings Index is also sometimes called the Browse List or the ACC List. Users can browse records by Author, Title, Subject or any other category defined by the library.

Z01 contains the headings according to the indexing defined in the tab00.lng and tab11_acc tables of the library's tab directory.

Z02

Z02 holds links between the Z01 - Access Headings - and the bibliographic records.

Z07

When a cataloging record (bibliographic, authorities, holdings, and so on) is created or updated, its system number is placed in the Z07 table. The Z07 is used by the system for controlling the updating of index files. Indexes are updated by the background job, initiated by UTIL E/1 (Update Doc Indexes - UE_01).

3.2 Main Headings Services

The Headings index is created by p-manage-02, enriched by UE_08, and updated by UE_01.

3.2.1 Update Headings Index (manage-02)

This service updates the Headings index of the database.

3.2.2 Pre-Enrich Bibliographic Headings Based on the Authority Database (manage-102)

p_manage_102 copies the headings from authority records into the bibliographic file Z01. This makes the long, complete run of UE_08, which matches headings with authority records, unnecessary.

For further information about this process, refer to the Authorities - Batch Jobs for Authority Enrichment and Correction of Bibliographic Libraries section of the Authorities chapter in the ALEPH System Librarian’s Guide.

3.2.3 Subject Subdivisions

For information about subject subdivisions, refer to the Authorities - Control section of the Authorities chapter in the ALEPH User Guide.

3.2.4 When to Run the Headings Index

The "Update - Headings Index" (manage-02) service must be run after:

A new code has been added to an already existing index,

A change has been made to the `tab00.lng` table or `tab11_acc` table that affects the Headings Index.

A change has been made in `tab_filing` that affects the display or normalized form of the heading.

The "rebuild" option in the "Procedure to Run" field of the "Update Headings Index" service must be run after making changes that affect already existing index entries.

We recommend rebuilding the Headings Index periodically, using the "Update Headings Index" service.

Note

After you run this service using the "Rebuild" option, always run the Alphabetize Long Headings (manage-17) service.

If the customer implemented the "Keyword Search in Heading List" functionality, he must run the `p_fill_z01x` script in the BIB library after the "Update Headings Index" service.

The following is an example of how to run the script in the USM01 library:

```
> cd $aleph_dev/aleph/proc/
> csh -f p_fill_z01x USM01
```

For more information, refer to the **Setting Up Keyword Search in Heading List Functionality** chapter of the *Aleph System Librarian's Guide – Web OPAC*.

3.3 Other Headings Services

3.3.1 Alphabetize Headings (p_manage_16)

This service alphabetizes the headings according to the rules for alphabetization that are stored in the `tab00.lng` table and in the `tab_filing` table.

These rules create a "filing text" by which the heading is alphabetized. The headings are then alphabetized according to the first 69 characters of the filing text of each entry.

When to Run this Service

This service must be run any time the rules for alphabetization have been changed in the `tab00.lng` table or in the tab filing table (including character conversion tables). For example, you may decide that you now want to alphabetize under "ue" instead of

"u". If the change in rules affects display or normalized texts, p-manage-02 must be run instead.

Note

After you run this service, always run the Alphabetize Long Headings (manage-17) service.

3.3.2 Alphabetize Long Headings (p_manage-17)

This service alphabetizes those headings whose filing texts are longer than 69 characters.

When to Run this Service

Run this service after conversions or if you suspect that there is a problem with the filing of long headings. In addition, if at any time, the rules for alphabetization are changed in the `tab00.lng` table or in the `tab_filing` table (including character conversion definitions), all the headings should be re-alphabetized to reflect the new rules. First run the Alphabetize Headings - Setup (manage-16) service, then run the Alphabetize Long Headings service.

3.3.3 Build Counters for Logical Bases (manage-32)

This service builds the counters for logical bases.

Counters for logical bases (Z0102) can be used to make browsing from the Web OPAC more efficient when scanning (browsing) logical bases which are less than 10% of the total database. "Y" in col. 8 of `./alephe/tab_base.lng` determines that the logical base must have Z0102 records built. If the library does not use logical bases, or if `tab_base.lng` does not include "Y" in col. 8, this section is irrelevant.

When a logical base is being browsed, the system uses these counters to determine whether or not to display the heading without having to retrieve the documents attached to the heading, read them and then determine how to proceed.

When to Run this Service

This service must always be run after building the headings index (manage-02). The database tables involved are:

Z01

Z02

Z0102

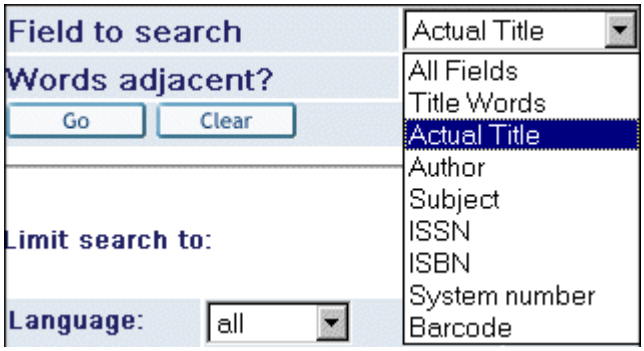
Z0102

Entries in the Z0102 table are built for each heading and for each logical base. The table includes the filing text, the acc-sequence identifier of the heading and a counter of the relevant documents attached to the heading. The pointer to the documents is still stored in the Z02 (ACCDOC) table.

3.3.4 Update Brief Records (manage-35)

This service updates and creates brief records. The Brief record display format is used to sub-arrange the records that are attached to a particular heading. These records are built according to the record's format, the headings index and the field of origin.

The structure of the brief record is defined in the `tab_z0101` and the `tab_z0101_text` tables of the library's `tab` directory.



The database tables involved are:

Z0101

Note that libraries that work with the Z0101 format must add the last-z0101-sequence counter to UTIL G/2.

3.3.5 Update Short Bibliographic Records (manage-07)

This service updates the Short Bibliographic Records of the database. The purpose of the Short Bibliographic Record is to provide bibliographic information in an efficient and timely manner, particularly for instances where bibliographic information is an adjunct to administrative information.

The Short Bibliographic Record is built by the system, according to the setup of the tab22 table (in the library's tab directory), when records are uploaded into the database (when the indexing parameter is set to 'Full'), or when records are added or updated through the Cataloging module.

A Short Bibliographic Record is an abbreviated version of the bibliographic record in standard Oracle table format. It can contain up to seven fixed (system-defined) fields (year, call number, call number key, author, title, imprint and ISBN/ISSN) each limited to 100 characters (except for the call number key field that is up to 80 characters). Additionally, it contains up to fifteen user-defined fields, each limited to 500 characters.

When to Run this Service

Run this service after making a change in the tab22 table that affects the Short Bibliographic Records.

The database tables involved are:

Z13

3.3.6 Create Sort Keys (manage-27)

Sort keys are used in the Web OPAC for sorting a set of records. The keys are data extracted from a field in the bibliographic record (for example, title, year). They are built in accordance with the rules set in `tab_filing` and `tab_character_conversion_line`.

3.3.7 Delete Unlinked Records (manage-15)

This service deletes headings that are not linked to records. Each modification of cataloging records creates new headings in the system. The old heading records are kept but the link to the record is deleted. This function deletes all such extra, unlinked headings.

Note that this process does not delete heading records which have an authority link. This is in order to keep the cross-references which are not linked to the document directly.

3.4 Linking Process

The process that triggers the building of the bibliographic heading - authority record connection is the UE_08 daemon. This process is initiated by UTIL E/8.

The UE_08 procedure checks new headings in the bibliographic library against the authority library and adds cross-references and/or multi-lingual equivalents to the bibliographic headings table.

A heading becomes "authorized" when a direct match is made between a heading [Z01] from the bibliographic database and a record from the authority database. The authority record is found through the authority library's headings index.

For further information about this process, refer to The Authority Database as Search Aid section of the Authorities chapter in the ALEPH User Guide.

4 Word Index

This chapter includes the following sections:

Defining Words

Database Tables Involved

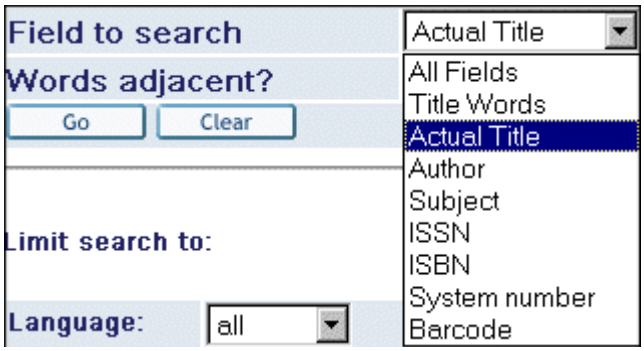
When to Update the Word Index

The Word Index contains a list of words that appear in specific fields of the bibliographic records in the database.

When a patron or librarian uses the Search function on a Word index in the Web OPAC, the system retrieves all documents containing the keyword(s) entered by the user.

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 word group.

Web OPAC searches can apply to the general word index or to any specified sub-index:

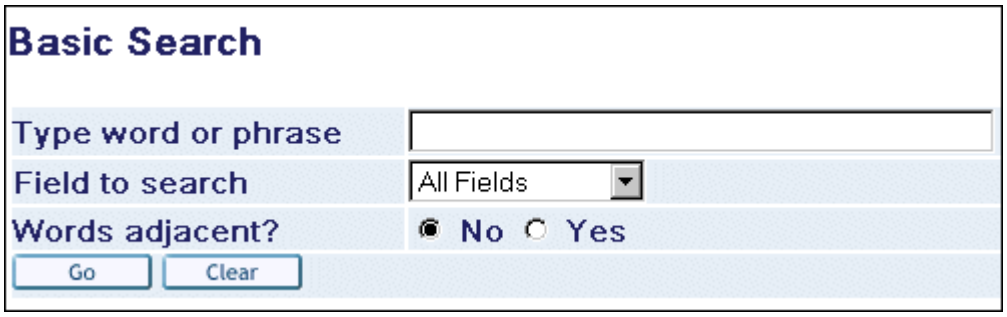


The system extracts each unique word from the specific fields of the record, stores it in the Word file, and maintains pointers to the document:

The extracted words are stored in the Z97 table that contains the word dictionary. The word dictionary is a list of all the searchable words derived from information in the document record.

When the user performs a Find/Search request from the Web OPAC, the system checks the Word Index to retrieve all documents containing the keyword(s) entered by the user.

The Word Index is usually used for the Search function in the Web OPAC and in the GUIs:



4.1 Defining Words

The default definitions of a word are:

A character string from blank to blank, or

From the beginning of a line to the first blank, or

From the last blank to the end of a line.

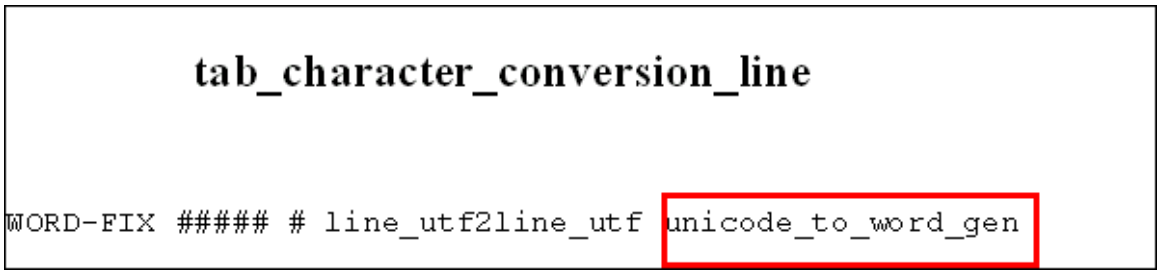
Word Breaking

Word-breaking routines are used to define what will be considered a "word" for the system in special cases (for example, I.B.M). Word-breaking routines are listed in Sorting and Word Breaking on page 36.

Word-breaking routines are specified for each word index group through column 6 of the `tab11_word` table of the library's tab directory.

Character Conversion

In addition to the word-building procedures, after text has been broken into words, a character conversion table is used to define equivalencies for characters. The system uses the character conversion table that is listed in column 5 of the `tab_character_conversion_line` table of the `$alephe_unicode` directory under the WORD-FIX entry:



4.2 Database Tables Involved

Z95

The records of this table contain the list of words in a document. The UE-01 online indexing process writes the Z95 records.

Z97

The Z97 table contains the word dictionary. It contains a list of all the words derived from the document record.

Each library can decide which fields of the document record form the basis for the Find/Search function in the Web OPAC. For example, you might decide to provide the ability to search by words from titles. The system extracts each unique word from the Title fields of the records and creates a unique entry for the word in the Z97 table. The Z97 table contains all the searchable words according to the indexing defined in the library's `tab/tab00.lng` and `tab11_word` tables.

Adjacency Searching

If the environment variable, "setenv ADJACENCY 2" is specified in ../alephe/aleph_start, then word indexing automatically builds word pairs in the Z97-Word Dictionary, for adjacency searching.

Note that the creation of paired words for adjacency searching requires a lot of disk space (four times more per record). On the other hand, it improves the performance of adjacency searching.

Z98

This table contains word document relations. Z98 contains a bitmap-compressed map of word occurrences in documents. The bitmap maintains pointers from the words registered in Z97 to the documents. The WORD3 utility (UTIL/F/4/word3) assists in reading the bitmap. This utility reads the bitmap in order to find the documents that contain word X stored in index Y.

4.3 When to Rebuild the Word Index

The Rebuild Word Index (manage-01) Target service must be run after making changes in the tab00.lng table or in the tab11_word table of the library's tab directory that affects word indexing.

5 Direct Index

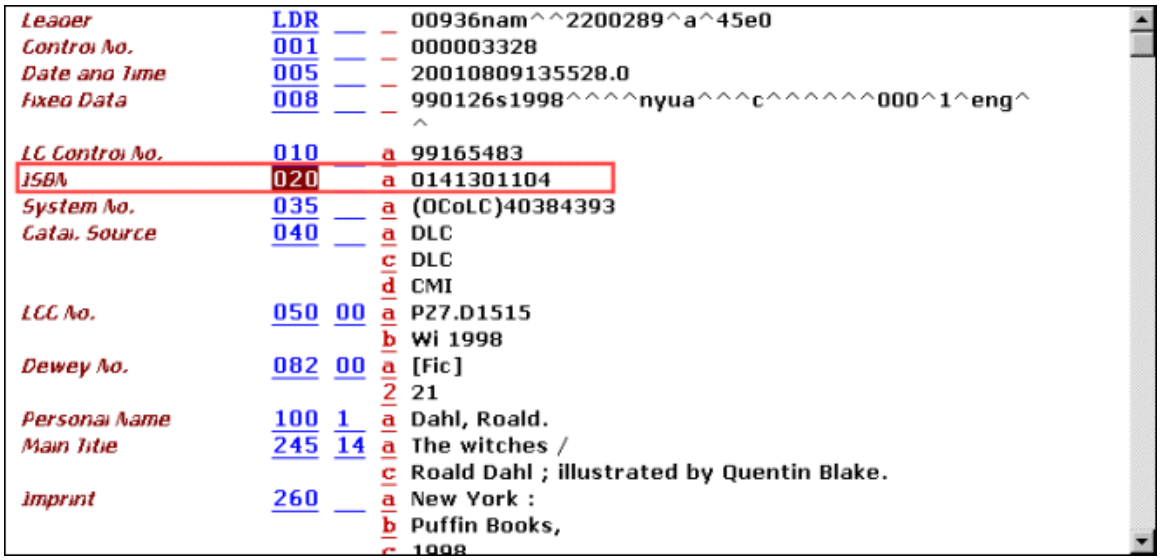
This chapter includes the following sections:

Filing Direct Indexes

When to Update the Direct Index

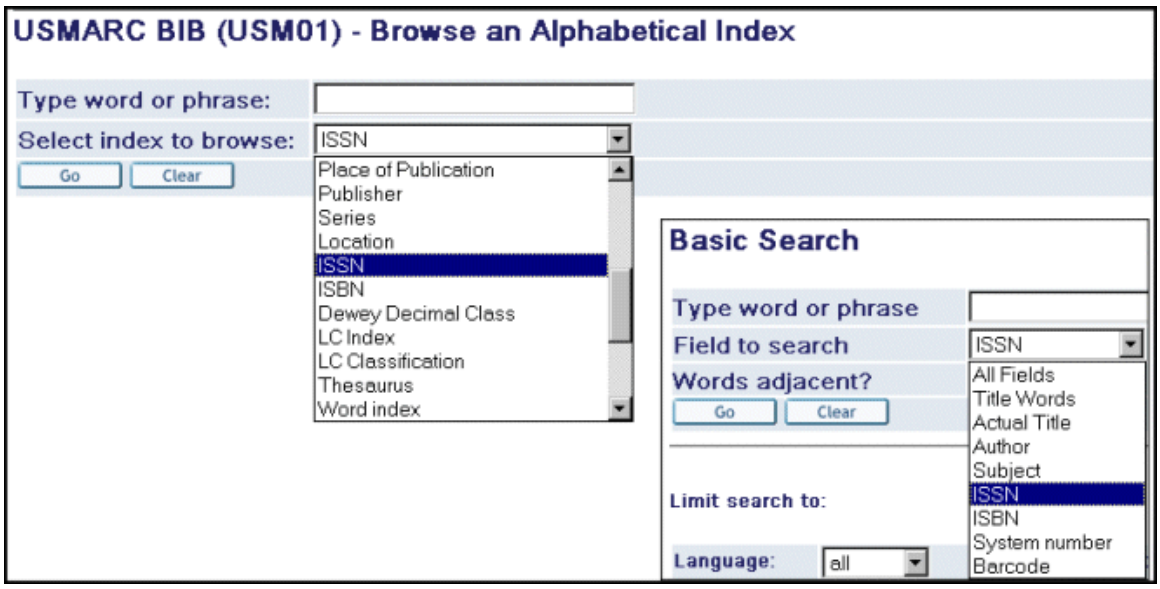
Database Tables Involved

Direct indexes enable the user to retrieve a specific record. A direct index is suited to unique or almost unique identifiers (such as the ISBN, the ISSN and the shelving location) of the record and provides quick access to a record:



Each library can decide which fields of the record are suited for this type of index and search capability.

Direct access functions are available from the menus of the Browse and Search options in the Web OPAC. Browsing a direct index displays the index term and the record title. When, for example, a patron or librarian uses the Browse function in the Web OPAC to locate the ISBN field of the bibliographic record, the system checks the Direct Index to retrieve the exact record containing the search string, or the next closest record. When the Search function is used, the system retrieves the matching record(s) in a set.



5.1 Filing Direct Indexes

Direct indexes are filed according to the filing text of the direct index, which is built according to the rules defined in the `tab_filing` table of the library's `tab` directory.

Filing routines are specified for each direct index through column 5 of the `tab00.lng` table of the library's `tab` directory.

5.2 When to Update the Direct Index

The Update Direct Index (manage-05) service must be run after making a change in the `tab00.lng` table or `tab11_ind` table of the library's tab directory that affects the direct index.

Rebuild
If changes that affect already existing index entries have been made, the Rebuild option in the procedure to run the drop-down list box must be run.

Update
If a new code has been added to an already existing index, or if a new index has been added, use the Update option in the procedure to run the drop-down list box.

Note

The Direct Index must be updated if a large number of records has been uploaded into the database in the "partial" mode and if the indexing has not been performed automatically for those records.

5.3 Database Tables Involved

Z11

The Z11 table contains the direct indexes defined in the `tab11` and in the `tab00.lng` tables of the library's `tab` directory.

Note

Z11 indexing is independent of the ue_01 process; it happens automatically when a bibliographic record is added or updated.

6 APAC Indexing

6.1 Headings Indexing

There are special features for filing Chinese, Japanese, and Korean headings.

Four filing routines are available in order to define whether CJK headings are sorted by Pinyin or by strokes. If these routines are not used, the headings are sorted by Unicode value and no special indexing setup is required. The following are the available CJK filing routines:

- **cjk_pinyin**: This routine adds an exclamation point (!) before each CJK character, translates the characters to pinyin using the Z114 table, and adds the Unicode value in decimal notation. The exclamation point (!) causes the pinyin filing-text to be sequenced separately from regular Latin characters.
- **cjk_stroke**: This routine is similar to the cjk_pinyin routine, except that each character is translated to the stroke value, using the Z114 table.
- **chi_pinyin**: This routine translates each character to pinyin using the Z114 table and adds the Unicode value (in decimal notation) for each character. The Unicode value is added in order to differentiate between different characters that have the same pinyin value. Because the pinyin filing text is 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 (for example, AUTC).
- **chi_stroke**: This routine is similar to the chi_pinyin routine, except that each character is translated to the stroke value, using the Z114 table.

The Aleph default setup is adjusted to sort by Pinyin. Therefore, when multiple CJK languages are used, it is required to create a separate CJK index for each CJK language that is indexed in the catalog and browsed by the user. The following is a description of how this setup is configured.

- **tab_expand**

```
!   1                    2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!>
```

ACC expand_doc_bib_lng_cjk

These lines set the headings and words indexing process to run the `expand_doc_bib_lng_cjk` routine. The routine adds the $$9 subfield with the language code from 008/34-36 to each field which contains Chinese, Japanese, or Korean characters. $$9 is then used in `tab11_acc` and `tab11_word` to put the CJK entries into different indexes.

- **tab11_acc**

```
! 1    2    3     4        5            6           7 8
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!-!-!
245##       9 chi        TITC  -e468
245##       9 jpn        TITJ  -e468
245##       9 kor        TITK  -e468
245##       9 -          TIT   -e468
```

In this example, each CJK language is set in a separate heading.

- **tab00.lng**

```
!   2     3   4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H TITC  ACC     51 00       00       Chinese Titles
H TITJ  ACC     52 00       00       Japanese Titles
H TITK  ACC     53 00       00       Korean Titles
H TIT   ACC     11 00       00       Titles
```

In this example, each CJK heading is assigned a separate filing routine.

- **tab_filing**

```
!1 2        3              4
!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
51 F chi_pinyin
52 F jpn
53 F kor
```

This example is coordinated with `tab00.lng` which has "51" as the filing procedure for TITC (Chinese Tiles) and so on.

When creating the filing key, "chi_pinyin" uses Z114 to translate characters to Pinyin and adds the Unicode value of the character after each Pinyin character, in order to differentiate between multiple Chinese characters that share the same Pinyin value.

There is no parallel dictionary available for Japanese and Korean. When creating the filing key for these languages using "jpn" and "kor", the system uses the Unicode value of the characters.

Note

If `tab100` is set to sort headings by Pinyin, but `tab_expand` is not set to add $$9 with the language code, then headings with Chinese characters will interfile with headings in Latin characters, because the headings will be written on the general AUT/TIT lists. However, in this case, filing procedure "cjk _pinyin " can be used to separate the Pinyin words from the regular sequence. It adds an exclamation point (!) before each

CJK character, translates the characters to Pinyin using the Z114 table, and adds the Unicode value in decimal notation. This causes the Pinyin filing-text to be sequenced separately from regular Latin characters within the general browse lists.

6.2 Words Indexing

Searching of APAC words is based on

- Normalizing the indexed text
- Defining Segmentation Routines for Indexing
- Defining Segmentation Routines for Searching

6.2.1 Normalizing the Indexed Text

Several normalizing routines may be required prior to indexing the APAC text. For example:

Punctuation Removal
Replace all CJK and Latin punctuation by space. This may be done using the existing "to_blank" routine.

Characters Transliteration
Replace characters by equivalent characters using a transliteration table. This may be done using the existing "char_conv" routine.

Normalizing the indexed text is done by setting tab11_word and tab_word_breaking, for example:

- tab11_word of the BIB library:

```
!1    2     3 4          5                     6  7 8 9     10
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!-!-!-!!!!!-!!!!!!!!!!!!!!
245##                    w                     01     WTI
```

  The WTI Index will contain words from 245 $$w using "01" word-breaking routines.

- tab_word_breaking of the BIB library:

```
!1 2        3                                    4
!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
01 # to_blank             !@#$%^()_={}[]:";<>,.?|\
01 # char_conv            CJK_TO_NORMAL
```

  The "01" word breaking routine removes punctuation using the to_blank routine, and transliterates the characters based on the char_conv routine.

The tab_character_conversion_line table in $alephe_root/unicode must define the char_conv table to be used. For example:

```
!!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!
CJK_TO_NORMAL       ##### # line_utf2line_utf             cjk_to_normal
```

The `cjk_to_normal` table must be defined in the $alephe_root/unicode directory.

6.2.2 Defining Segmentation Routines for Indexing

The tab_word_breaking routine that will be used in the indexing process is defined in column 6 of the relevant tab11_word line. For example:

```
1    2     3 4          5                     6  7 8 9     10
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!-!-!-!!!!!-
!!!!!!!!!!!!!
245##                    w                      01     WTI
```

The tab_word_breaking routine will, in addition to the above described normalization routines, also define the required segmentation routine. For example:

```
!1 2        3                                    4
!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
01 # to_blank             !@#$%^()_={}[]:";<>,.?|\
01 # char_conv            JAPANESE_TO_NORMALIZED
01 # morpheme_index
```

See the Suggested APAC Indexing and Searching Segmentation Routines Setup on page 29 for examples of segmentation routines.

6.2.3 Defining Segmentation Routines for Searching

The tab_word_breaking table is used to define the policy for segmentation of a given string during the searching process.

The following tab_word_breaking sections are reserved by the system for processing the search string when a FIND action is invoked:

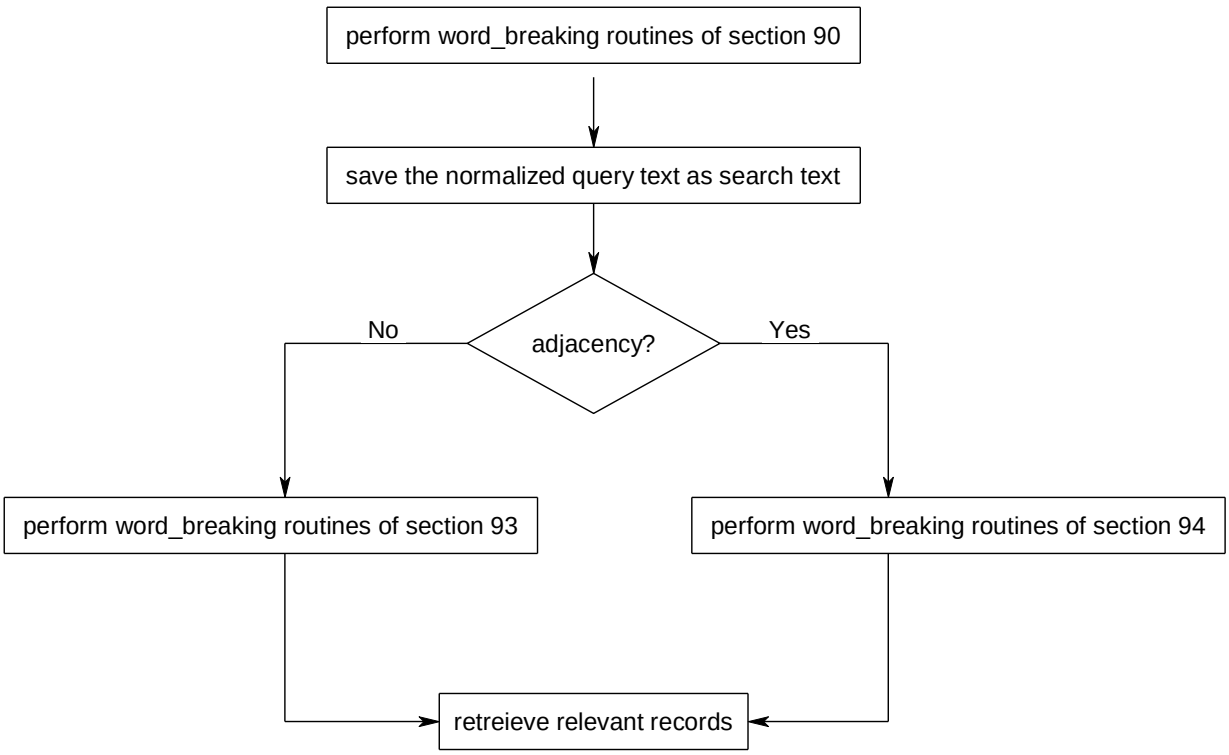
- Section 90 - the word_breaking routines set in this section will be performed for parsing a FIND query.

  The saved search text is the find query text AFTER performing the routines set in section 90.

- Section 93 – the word_breaking routines set in this section will be performed on the find query text after word_breaking routines of section 90 have been performed. The routines set in this section will be performed only when searching without adjacency.
- Section 94 – the word_breaking routines set in this section will be performed on the find query text after word_breaking routines of section 90 have been performed. The routines set in this section will be performed only when searching *with adjacency*.

Note that routines of section 90 are always performed on the FIND string, while sections 93/94 are performed only if the FIND string contains CJK text.

The following is a summary of the flow when the system segments the FIND query:



The following is an example of a tab_word_breaking setup for CJK search:

```
!1 2        3                                    4
!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

90 # to_blank             !@#$%^()_={}[]:";<>,.?|\

93 # char_conv            CJK_TO_NORMALIZED
93 # cjk_2gram_all

94 # char_conv            CJK_TO_NORMALIZED
94 # morpheme_search
```

Generally, it is possible to configure one word breaking routine to be used if the adjacency flag is N (section 93 of tab_word_breaking) and a different word breaking routine if the adjacency flag is Y (section 94 of tab_word_breaking).

To configure Aleph to use two different routines with different adjacency definitions:

1. Add the following to alephcom.ini:

```
[SearchFind]
FindAccurate=Y
```

This adds the Force checkbox next to the Adjacency checkbox in the GUI search tab.

The following table describes the behavior of Aleph when the Force checkbox is configured:

Adjacency Flag	Force Flag	Value to Server	Behavior
Not checked	Not checked	N	93 section + N adjacency
checked	Not checked	Y	94 section + Y adjacency
Not checked	checked	M	94 section + N adjacency
checked	checked	B	93 section + Y adjacency

2. Add M and B values to the search HTML page for OPAC.

6.2.4 Suggested APAC Indexing and Searching Segmentation Routines Setup

The following is a summary of the existing segmentation routines that handle APAC text:

- **split_cjk** – The segmentation is performed according to the Z113 table (Chinese dictionary) from left to right. In addition, the text is split, character by character.

- **cjk_to_word** – The text is divided into words from right to left according to the longest word principle by using the Z113 table (Chinese dictionary).
- **cjk_split1** – Each CJK character is considered as a word when defining segmentation routines that are performed on the search string.
- **cjk_split3** – Words are determined according to a pre-defined dictionary (z113).
- **cjk_simplified** – Characters are translated to a simplified form.
- **cjk_input_adj** – Each CJK character is considered as a word when searching with adjacency.
- **cjk_input** – Text is divided into words from right to left according to the longest word principle by using the Z113 table (Chinese dictionary). The next word starts after last character of the previous word.
- **cjk_2gram_lng** – This routine is used with a parameter. Allowed values are CONC and NO-CONC.
  - cjk_2gram_lng with parameter: CONC
    Words are:
    - All Bi-Gram segments of the concatenation of consecutive CJK substrings of the same writing system (Chinese, Hangul or Kana).
    - All space delimited non CJK substrings (“Latin Words”)
    - All space delimited single CJK characters
    - All Chinese characters
    - All Hangul characters that are a one character Korean word (defined in the /alephe/unicode/tab_cjk_single_char_word table).
    - All normalized Katakana characters that are the normalized form of one Kana character Japanese word (defined in the /alephe/unicode/tab_cjk_single_char_word table).

  - cjk_2gram_lng with parameter: NO-CONC

Words are all Bi-Gram segmentation of the CJK space delimited substrings. In other words, no concatenation of the CJK substrings is done before the Bi-Gram segmentation. In addition, all "Latin words" are considered words.

- **cjk_2gram_all** – Words are all Bi-Gram segments of the concatenation of the whole text (including non CJK text).
- **cjk_add_single** – Every CJK character is added as a word in the index.
- **cjk_add_space –** Insert space between characters of different writing systems.

- **morpheme_index –** To be used only for segmentation during the indexing process. Creates all possible substrings that are concatenation of successive words in the given field.

  The number of morphemes that will be concatenated is limited to 30. As result, if the patron will enter a search term that consists of more than 30 morphemes with no spaces between them, the search will not match any record.

  When this routine is used for segmentation during the indexing process, the cjk_morpheme_search routine must be used during the segmentation of the FIND string.
  Note that when indexing according to the Bi-Gram algorithm, Japanese characters and Chinese ideograms are handled in the same manner.

- **morpheme_search** – To be used when morphem_index is used in the indexing process.
  Note that when indexing according to the Bi-Gram algorithm, Japanese characters and Chinese ideograms are handled in the same manner.

  Note that the Morpheme index algorithm does not produce word pairs; therefore, use Morpheme search only for searching without adjacency (section 93 of tab_wrd_breaking).
- **morpheme2_index –** To be used only for segmentation during the indexing process. Creates all possible substrings that are a concatenation of successive CJK words that are less than the number of characters (configurable) in the given field.

  When this routine is used for segmentation during the indexing process, the morpheme2_search routine must be used during the segmentation of the FIND string.
- **Morpheme2_search** – To be used when morphem2_index and cjk_2gram_lng / cjk_2gram_all are both used in parallel in the indexing process. If the searched word is longer than 8 characters, this routine performs the cjk_2gram_lng algorithm. Otherwise, words are found based on the words created by the morpheme2 algorithm.

- **thai_index –** To be used only for segmentation during the indexing process.
  The text is divided into words from right to left according to the longest word principle by using the Z117 table (Thai dictionary).

  When this routine is used for segmentation during the indexing process, the thai_search routine must be used during the segmentation of the FIND string.

- **thai_search** – To be used when thai_index is used in the indexing process.
  The text is divided into words from right to left according to the longest word principle by using the Z79 table (Aleph dictionary). The next word starts after last character of the previous word.

The following options may be used for defining the indexing (in tab11_word) and searching ("93" and "94") segmentation routines:

- For indexing use *split_cjk*
  For searching use:
  - *cjk split_1*

  Or
  - *cjk split_3*

  Or
  - *cjk_char_to_simlified*
- For indexing use *cjk_to_word*

  For searching use:
  - *cjk_input_adj* (for adjacncy search)
  - *cjk_input* (for non adjacency search)
- For indexing use:
  - *cjk_2gram_lng with parameter "CONC"*
  - *cjk_2gram_all*

  For searching use:
  - *cjk_2gram_lng with parameter "NO-CONC"* (for non adjacency search)
  - *cjk_2gram_all* (for adjacncy search)
- For indexing use:
  - morpheme_index

  For searching without adjacency use:
  - morpheme_search

- For indexing use:
  - thai_index

  For searching use:
  - thai_search

- For indexing use:
  - morpheme2_index
  - cjk_2gram_lng / cjk_2gram_all

  For searching use:
  - morpheme2_search

In the following example, if the indexing is done using the "01" and "02" segmentation routines then the following tab_word_breaking setup is recommended.

```
!1 2         3                                   4
```

```
!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

01 # to_blank             !@#$%^()_={}[]:";<>,.?|\
01 # char_conv            HANJA_TO_HANGUL
01 # char_conv            KANA_TO_NORMALIZED
01 # cjk_add_space
01 # cjk_2gram_lng        CONC

02 # to_blank             !@#$%^()_={}[]:";<>,.?|\
02 # char_conv            HANJA_TO_HANGUL
02 # char_conv            KANA_TO_NORMALIZED
02 # cjk_add_space
02 # cjk_2gram_all


93 # to_blank             !@#$%^()_={}[]:";<>,.?|\
93 # char_conv            HANJA_TO_HANGUL
93 # char_conv            KANA_TO_NORMALIZED
93 # cjk_add_space
93 # cjk_2gram_lng        NO-CONC


94 # to_blank             !@#$%^()_={}[]:";<>,.?|\
94 # char_conv            HANJA_TO_HANGUL
94 # char_conv            KANA_TO_NORMALIZED
94 # cjk_2gram_all
```

Note that the following is required to supplement this setup:

The tab_character_conversion_line table in $alephe_root/unicode must contain the following lines:

```
!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!
HANJA_TO_HANGUL      ##### # line_utf2line_utf             hanja_to_hangul
KANA_TO_NORMALIZED         #####       #       line_utf2line_utf
kana_to_normalized
```

The hanja_to_hangul and kana_to_normalized tables must be defined in the $alephe_root/unicode directory.

Note: If you want to index single CJK characters as words in addition to regular words, set `cjk_add_single` in `tab_word_breaking`. Note that this routine should be added AFTER the routines `cjk_to_word` and `split_cjk`, if they are in use.

7 Main Tables Supporting Indexing

The following index-related tables are explained below:

tab00.lng

tab11_acc

tab11_word

tab11_ind

tab_word_breaking

tab_filing

tab_expand

tab_character_conversion_line

tab20

tab22

tab_sort

tab_aut

tab_base

tab00.lng

This table is used to define the system access codes and names. It is divided into three sections for each type of index:

ACC file (Headings)

WRD file (Words)

IND file (Direct)

tab00.eng

```
!   2     3    4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H WRD   W-001      00       01       Words
H WTI   W-002      00       02       W-titles
H WAU   W-003      00       02       W-authors
H WSU   W-004      00       03       W-subjects
H WUT   W-005      00       02       W-Unif .Titles
```

```
!   2     3    4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H TIT   ACC     11 00       00       Titles
H AUT   ACC     01 00       00       Authors
H SUB   ACC     01 00       00       General Subjects
H PLA   ACC     11 00       00       Place
H PUB   ACC     01 00       00       Publisher
```

```
!   2     3    4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H SYS   IND     21 00       00       System Number
H BAR   IND     21 00       00       Barcode
H 010   IND     22 00       00       LCCN
H ISBN  IND     21 00       00       ISBN
H ISSN  IND     21 00       00       ISSN
```

tab11_acc

The table is used to assign record fields to headings indexes. This table is limited to 1500 lines.

tab11_word

The table defines the connection between the record fields and the word indexes (one field can be indexed in one or more word groups). A field can be listed several times in tab11_word, in order to index it a number of times, with different word breaking routines each time. This table is limited to 10,000 lines.

You can define word group codes which are up to five characters in length. Note that there is no filler between the multiple word index definitions, cols. 9-18.

tab11_ind

The `tab11_ind` table is used to assign fields to direct indexes. This table is limited to 500 lines.

tab_word_breaking

This table contains word breaking specifications. For word breaking procedures, refer to that section in Sorting and Word Breaking on page 36.

tab_filing

This table is used for filing and normalization procedures that are used when building headings and for defining filing procedures that are used when building index entries and sort keys. For more information, refer to Sorting and Word Breaking on page 36.

tab_expand

Defines expand procedures which are activated when an index is created. For more information, refer to Expand Routines, Tables and Indexing Expanded Fields on page 46.

tab_character_conversion_line

Character conversion tables are used to define equivalencies for characters. All characters are sorted by their Unicode values by default. In order to force a different sort order, you can set an equivalency for sorting. The system uses the character conversion table that is listed in column 5 of the `tab_character_conversion_line` table of the `$alephe_unicode` directory

For example, the character conversion tables assigned to the WORD-FIX instance are used to define equivalencies of characters for the purpose of creating words in the words file.

tab22

The `tab22` table of the library's tab directory defines which fields will be included in the short bibliographic record. The table is also used to determine how these fields are created. The Short Bibliographic Record is built by the system, according to the definitions of this table, when records are uploaded into the database (when the indexing parameter is set to 'Full'), or when records are added or updated through the Cataloging module.

tab_sort

The `tab_sort` table defines the fields and subfields assigned to a sort key (sort keys are used for sorting a set of records in OPAC). Up to five alternative field / subfield combinations can be defined for each sort key.

Keys 02 and 03 are used by some services to define the sort by Author and by Title respectively. The actual way in which the sort by Author and sort by Title works is configurable. For example the Author can be sorted according to field 100, or field 110 or a combination of both. However, you must make sure that this configuration retains key 02 for Author and key 03 for Title.

tab_aut

Establishes which headings indexes should be subject to authority control. This table also designates - per index - which authority database is to be checked for a match.

tab_base

This table defines the logical and physical databases that can be accessed via the Web OPAC and through the Search function in the GUIs. Logical bases are defined by setting up a FIND command that serves as a pre-filter or scope. The FIND command can be up to 500 characters in length.

In order to set up a logical base that includes all records **except** for a specified group

of records, use *alldocuments* to define all records, together with *not*. For example, *alldocuments not wst=suppressed* sets up a logical base that includes all records except for those that contain *suppressed* in the *wst* word group.

tab20

Defines the rules by which a headings index is enriched from an Authority record, creating headings that are cross-referenced.

8 Sorting and Word Breaking

This chapter includes the following sections:

Sort Headings and Indexes

Sorting Item Lists

Word Breaking

8.1 Sorting Headings and Indexes

Each Heading (Z01), Index (Z11) and Sort key (Z101) record has what are termed "filing keys" or "filing text". This is the form of the heading or index term for filing (sorting) purposes. The rules that govern the values of the filing keys are set in the library's `tab_filing` table.

Headings have an additional related feature, called normalization. Normalization refers to the process whereby diacritics, most punctuation, special characters, and case differences are stripped from headings. This is in order to neutralize slight differences. This is important in the headings index where each unique heading is stored only once.

Normalization routines are defined in the `tab_filing` table together with filing routines and display text routines.

When a new heading is added to the database, if there is already another heading with the same normalized text (Z01-NORMALIZED-TEXT), even if the display text (Z01-DISPLAY-TEXT) is different, both headings are considered to be the same. In this case, no heading record (Z01) is registered for the new heading and the display text of the first heading is taken.

The filing key of the headings is built in two stages:

Display text (Z01-DISPLAY-TEXT) to normalized text (Z01-NORMALIZED-TEXT).

Normalized text (Z01-NORMALIZED-TEXT) to filing text (Z01-REC-KEY).

For this reason, when creating the filing form of the heading, it is not necessary to perform routines that have already been performed in order to create the normalized text of the heading.

The library's `tab/tab_filing` table defines the normalization and filing routines that are used. The order of the subroutines within a routine is important; for example, you

cannot relate to a subfield code if you have previously set "del subfield code". The table includes the following columns:

Column 1 - Identifier
Contains the two-digit identifier of the filing routine. This identifier is used in column 5 of the tab00.lng table (for headings and direct indexes) and column 3 of the tab01.lng table (for sort keys).

Column 2 - Routine Usage
This column is relevant only for headings (Z01). It is not relevant for direct indexes (Z11) or for sort keys (Z101). It defines the usage of the routine. The available options are:

D = Display text procedures

N = Normalized text procedures

F = Filing text procedures

Note that the routines defined in the `tab_filing` table require an 'F' section, so even if this section is not needed (because the lines for 'N' suffice), an 'F' line still needs to be present with "no" as the filing procedure.

Column 3 - Procedure
This column contains the name of the filing or normalization procedure.

Column 4 - Parameters
Parameters for the filing/normalization procedure (when relevant). If the parameters are characters, and a character is out of the ASCII range, type this character in Unicode notation.

The available filing and normalization procedures are:

abbreviation: compress a dot between single characters (for example, I.B.M. changes to IBM). This routine works only with characters in the 7-bit ASCII range.

add_prefix_hash: adds a hash (#) sign immediately after the subfield code specified in the parameters column. It is mostly required in order to correctly sort headings derived from the fix_doc_aut_duplicate program.

bbk: special procedure for Russian filing standards, in which the sorting sequence is special characters, followed by Cyrillic characters, followed by Latin characters, followed by numbers.

char_conv: perform the 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`.

chi: translates each character to pinyin, using the Oracle table that contains the pinyin form of Chinese characters (Z114), and adds the Unicode value for each character. The Unicode values added in order to differentiate between different characters have the same pinyin value. Since the pinyin filing-text is sequenced together with regular Latin characters, this routine should be used for browse lists that use the language code from the 008 MARC 21 field to separate by language, and are separate for Chinese (for example, AUTC). Note that separate browse lists can be created by using the expand_doc_bib_lng_cjk expand program.

cjk: adds ! before each CJK "character", and translates the characters to pinyin, using the Oracle table that contains the pinyin form of Chinese characters (Z114) and adds the Unicode value in decimal notation for each character. This causes the pinyin filing-text to be sequenced separately from regular Latin characters.

compress_blanks_cjk: Deletes space between two CJK characters.

hangul_call_no: Filing item's call number routine (column 3) for Korean libraries. The routine uses Unicode values of each character and places Hangul characters first, followed by Latin, followed by numbers. Additional value 0 or 1 can be set in column 4 of tab_filing.

0 – Default parameter, apply LC routine for hangul_call_no.

1 - Apply Dewey routine for hangul_call_no.

hangul_call_no routine includes special handling for sorting Chronology Number part of Call Number's Item part information ($$i of Z30-CALL-NO and Z30-CALL-NO-2). The numbers within the chronology number will not be sorted according to their sequential value, but according to the year they represent. This part of the sorting algorithm uses the $$9 subfield which holds the full year information.

The hangul_call_no routine uses a special entry in column 1 of tab_character_conversion_line: FILING_HANGUL_CALLNO. The library may define in column 5 of tab_character_conversion_line any character conversion table that should be used for FILING_HANGUL_CALLNO. The hangul_call_no sorting routines consult the defined character conversion table to apply the sorting routine.

The following is an example of ./alephe/unicode/tab_character_conversion_line that defines filing_hangul_callno table for FILING_HANGUL_CALLNO:

```
FILING_HANGUL_CALLNO ##### # line_utf2line_utf   filing_hangul_callno
```

The ./alephe/unicode/filing_hangul_callno table must be set to define the Unicode values to be used by the hangul_call_no sorting routine.

Each subfield of the Korean call number undergoes a different manipulation for creating the filing value of the call number. Therefore, in order to be able to search correctly, the search query must include the subfield codes.

comma: All commas are turned into blanks except for the first comma in subfield $a (particularly used for the normalization of names).

compress: Compress (i.e. strip) the characters listed in column 4. You can compress characters by specifying their Unicode values in the parameters column. In the following example, the comma (Unicode value 002C) and the period (Unicode value 002E) are compressed:

```
!1 2        3                                  4
!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
01 N compress             U+002CU+002E
```

compress_blank: delete blanks

compress_perblnk: compresses a blank when it is preceded by a period ("."). This is needed for the filing of certain LC Call Numbers.

del_lead_space: deletes any leading spaces after a subfield. For example: $$a XXX is changed to $$aXXX. The database data never has a space after the subfield code, but

the to_blank routine could create a space. Accordingly, the del_lead_space routine must be placed before del_subfield.

del_subfield: delete subfield sign ($$x)

del_subfield_code: the "$$" sign is retained, but the subfield code is replaced by a hyphen ("-"). This is used for normalization, so that headings will match when the subfield content is the same, even if the subfield codes are different.

dewey_call_no: special procedure for the correct sequencing of Dewey Call Numbers.

end_punctuation: deletes the characters listed in column 4 of the `tab_filing` table when this is the last character in the heading. Note that this function is mostly used to remove / : = and so on at the end of a title, and so on. It is intended for routines of type "D" - display text conversions.

end_sub_punctuation: This routine deletes the characters listed in column 4 of the tab_filing table from each subfield comprising the heading.

expand_num: expand number. This routine adds leading zeroes to fill numbers to a certain number of digits for numeric filing. The maximum number of digits can be specified as a one or two digit number by using the parameters column in `tab_filing` (col. 4). If column 4 is left blank, then a default of 7 digits is used.

Following is a sample of the `tab_filing` table:

```
!1 2        3                                      4
!!-!-!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

11 F expand_num           13
```

get_subfields: use only the subfields, or subtract some using "-" as listed in col.4.

get_subfields_order: this procedure is similar to the get_subfields routine except that it retains the order of the subfields specified in col. 4.

icelandic_name: changes the order of subfields 7 and 1, placing subfield 7 after subfield 1. Intended for sorting OPAC browse lists. For example, the display text:

```
$$aAlexander $$7Alfred $$1Jonsson$$c1943-
```

becomes the filing text:

```
Alexander Jonsson Alfred 1943-
```

jpn: 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 the 008 MARC 21 field to separate by language, and that are separate for Japanese (for example, AUTJ). Note that separate browse lists can be created by using the `expand_doc_bib_lng_cjk` expand program.

kor: 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 the 008 MARC 21 field to separate by language, and for those that are separate for Korean

(for example, AUTK). Note that separate browse lists can be created by using the `expand_doc_bib_lng_cjk` expand program.

lc_call_no: special procedure for correct sequencing of LC Call Numbers. Note that this routine adds the following three characters to the index records it creates: ! " #. 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 recommended that del_subfield be added.

lc_call_no_2: same as lc_call_no, except that this routine enables non LC call numbers to be included in call number indices.

mc_to_mac: change initial mc to mac

no: this routine is used when the procedure defined in the `tab_filing` table does not contain an 'F' section. The procedures defined in the `tab_filing` table require an 'F' section, so even if this section is not needed (because the 'N' section - used for normalization purposes - suffices), an 'F' line still needs to be present with "no" as the filing procedure. For example:

```
01 F no
01 N to_blank             !"()-{}<>;:.?/\@*%=^_`~
01 N comma
01 N del_subfield_code
01 N char_conv            FILING-KEY-10
```

non_filing: drop initial text using non-filing indicator

non_numeric: delete non-numeric characters

none: This routine builds the call number key as the call number itself.

numbers: compress a comma and a dot between numbers (for example, 2,153 changes to 2153)

pack_spaces: compresses all multiple spaces to a single space

subfield_mab: 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.

suppress: this program drops all text contained within the signs << and >>, including the characters themselves. You can also add comma-delimited parameters to the *suppress* routine in Column 4 of `tab_filing`. Here is an example from `tab_filing`:

```
!1 2        3                                    4
!!-!-!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
94   suppress             88-89,<<>>
```

If the parameter *88-89* is specified, the control characters U+0088 and U+0089 will be used instead of << and >>. The parameter <<>> acts the same way as the default.

If both parameters are specified, the input text will undergo suppression twice: once with << and >> as delimiters, and again - with U+0088 and U+0089 as delimiters (or vice versa).

The following parameter combinations are allowed:

<<>> The same as the default

88-89 Suppression only with U+0088 and U+0089 as delimiters.

88-89,<<>> (2) and then (1)

<<>>,88-89 (1) and then (2)

to_blank: change characters listed in col.4 to blank.

You can compress characters by specifying their Unicode values in the parameters column. Enter the notation in the form U+<Unicode value> - for example "U+0153".

to_blank_2: changes to blank the characters listed in the parameters column (col. 4). The character is changed to a blank only if it is followed by a blank or if it is at the end of the field.

For example, if the comma is listed in the parameters column under this procedure, then "Schiller, Friedrich" is changed to "Schiller Friedrich", but "one,two,three" is not changed.

to_lower: change to lowercase

to_carat: change subfield sign to ^^ (for hierarchical sorting of headings). WARNING! Although this will file a heading such as "$$aArt $$zZambia" before "$$aArt, Canadian", when the system performs a browse search, the search query is taken word-by-word, character-by-character, transforming multiple blanks to a single blank. Therefore, it is not possible to zero in on "art ^^zambia" in a browse search, and although the list will be hierarchically arranged, it will be difficult to use.

year_uu: replaces *u* with zero (*0* in the year formats where a date element is unknown, for example, 19uu, 197u. Note that it should be activated before the **expand_num** routine.

Following is a sample of the `tab_filing` section used for title headings:

```
11 D end_punctuation      :,=;/
11 N to_lower
11 N to_blank             !@#%^&*()_+-={}[]:";?,./~`
11 N pack_spaces
11 N del_subfield_code
11 F del_subfield
11 F suppress
11 F numbers
11 F to_blank             $<>
11 F expand_num
11 F non_filing
11 F compress             '
11 F pack_spaces
11 F char_conv            FILING-KEY-01
11 F cjk
```

8.2 Sorting Item Lists

The sorting order of items throughout most modules of the system is determined in a single table, `tab_z30_sort` in the Administrative library.

This table includes sort options for both issue and non-issue type items for each module/function which includes lists of items.

The sort options are made up of two elements:

Sorting type - defines the various levels of sorting. For example, serial items can be sorted by volume, then by issue number, then by part number.

Sorting order - defines whether the sorting will be ascending or descending.

The following table includes the modules/functions which are dealt with by this table and their codes:

Module / Function	**Code**
Web OPAC	WWW-A
Web Course Reading	WWW-R
Serials client	SERIAL
Search client	SEARCH
Circulation client	CIRC
Items client	ITEM
Items for binding	ITEM-BIND
Acquisitions/Serials client	ACQ
Services	BATCH
Lost Item Report	CIR-16
for use by fix_doc_create_86x procedure	86x
Navigation window	TREE

8.2.1 Sort Options

The available sorting types are:

For **issue type** items:

00 - if chronological-i(year) is spaces and enumeration-a(volume) is spaces, then description + item-sequence. if chronological-i(year) is not spaces, then chronological-i(year) + enumeration-a(volume) + enumeration-c(part) + enumeration-b(issue) + item-sequence.

01 - if chronological-i(year) is spaces and enumeration-a(volume) is spaces, then description + item-sequence. if chronological-i(year) is not spaces, then chronological-i(year) + enumeration-a(volume) + enumeration-b(issue) + enumeration-c(part) + item-sequence.

02 - if chronological-i(year) is spaces and enumeration-a(volume) is spaces, then description + item-sequence. if chronological-i(year) is not spaces, then hol-doc-

number + chronological-i(year) + enumeration-a(volume) + enumeration-b(issue) + enumeration-c(part) + item-sequence.

03 - sublibrary + item-sequence.

06 - sublibrary + collection code + chronological-i(year) + chronological-j (year) + chronological-k(year) + description + copy-id (site specific).

07 - 85x-type + sublibrary + collection + linking-number + if enumeration is not spacesi, then enumeration. if enumeration is spaces, then description.

08 - 85x-type + sublibrary + collection + linking-number + copy-id + if 85x-type is 4 or 5 then supp-index-o + if chronological is not spaces and enumeration is not spaces, then chronological + enumeration.

12 - sub_library + collection + if chronological-i(year) is spaces and enumeration-a(volume) is spaces then description + if chronological-i(year) not spaces then chronological-i(year) + enumeration-a(volume) + enumeration-b(issue) + enumeration-c(part) + copy.

13 - by barcode (using filing routine code 98)

For **non-issue type** items:

00 - if enumeration-a(volume) is not spaces, then: enumeration-a(volume) + enumeration-b + enumeration-c(part) + sublibrary + collection. if enumeration-a (volume) is spaces, then: description + enumeration-c(part) + sublibrary + collection.

01 - enumeration-a(volume) + enumeration-b + enumeration-c(part) + description + sublibrary.

02 - if enumeration-a(volume) is not spaces, then: enumeration-a(volume) + enumeration-b + enumeration-c(part) + sublibrary. if enumeration-a(volume) is spaces, then: description + sublibrary.

03 - if description is blank, then: enumeration-a(volume) + chronological-i (year) + enumeration-b + enumeration-c(part) + sublibrary + item status. if description is not blank, then: description + sublibrary + item status.

04 - if enumeration-a(volume) is not spaces, then: hol-doc-number + enumeration-a(volume) + enumeration-b + enumeration-c(part) + sublibrary + collection. if enumeration-a(volume) is spaces. then: hol-doc-number + description + sublibrary + collection.

05 - sublibrary + item-sequence.

06 - sublibrary + collection code + description + copy-id (site specific).

12 - sub_library + collection + if chronological-i(year) is spaces and enumeration-a(volume) is spaces then description + if chronological-i(year) not spaces then chronological-i(year) + enumeration-a(volume) + enumeration-b(issue) + enumeration-c(part) + copy.

13 - by barcode (using filing routine code 98)

The available sorting orders are:

A - Ascending

D - Descending.

For the Items, Serials and Circulation modules, you can define more than one sorting option in tab_z30_sort, but each option must be given its specific code (for example, CIRC-1, CIRC-2, and so on).

These options must then be also defined in the `pc_tab_exp_field.lng` table of the Administrative library, so that they will be available as drop-down menu options in the Item List windows of these modules.

8.3 Word Breaking

For word indexing, individual words of a field are written in the words table. Basically, a word is a group of characters between white spaces. The library's `tab_word_breaking` table is used to define instances where other factors are taken into account. For example, is a hyphen a space, or is a hyphen compressed and treated as if it were not there at all?

Text undergoes word breaking using the characteristics that are listed in one routine in the `tab_word_breaking` table. The structure of the table includes four columns:

col.1: Two-digit identifier of the word breaking routine. This identifier is used in column 6 of tab11_word.

col.2: Used only for identifying CJK range.

col.3: Name of the word breaking procedure

col.4: Parameters for the word breaking procedure (when relevant)

Facets for the word breaking procedures are:

2_hyphen: this routine changes two adjacent hyphens (--) to a blank. This word breaking procedure can be used, for example, to change to blank consecutive hyphens in the MARC 21 field 505 (FORMATTED CONTENTS NOTE) that uses hyphens as separators. The following is an example of a 505 field:

```
5050  L $$aHow these records were discovered -- A short sketch
of the Talmuds -- Constantine's letter.
```

compress: compress (that is, strip) the characters listed in col.4

compress_blank: delete blanks

del_subfield: change subfield sign ($$x) to blank

Force_delimiter: This routine changes the subfield sign ($$x) to blank-z-blank in order to prevent words across subfields being considered adjacent.

to_blank_2: change characters to blank if the character is followed by a blank

to_blank: change characters listed in col.4 to blank

subf_to_sign: change second and subsequent subfield signs to the single character listed in col.4

ccl_brackets1: This routine searches for a term that includes parenthesis (brackets) in a word. The parameter of the routine "ccl_brackets1" can be either "c" (compress brackets) or "b" (replace by blank). If the routine "to_blank" is used before

"ccl_brackets1", make sure that the parenthesis are not included; otherwise, "to_blank" removes all parenthesis (brackets).

blank_to_carat: change blanks to caret (^)

numbers: compress a comma and a dot between numbers, for example, 100,000 or 100.000 -> 100000

abbreviation: compress a dot between single characters. For example, I.B.M. becomes IBM.

marc21_41: used for separating in MARC21 041 field

Notes:

The procedures must be listed in logical order. For example, numbers must be listed before compress or change_to_blank if a comma or a dot are included in them. Otherwise, they will no longer be present when the numbers procedure is used.

Word breaking procedures are defined in the tab11_word table (column 6) of the library's tab directory. A line can be listed several times in the tab11_word table in order to index it multiple times, with different word breaking procedures each time.

In the following example, words from the 100 field are indexed according to the word breaking procedures 01 and 02:

```
100##                   -6                   01     WRD  WAU
100##                   -6                   02     WRD  WAU
```

In `alephe/unicode`, there is a table called `unicode_to_word_gen`. The system automatically uses this table when building the word breaking. The table can (and does) include values that change a character to a blank (by assigning the value 0020) or can compress a character (by assigning the value 0000). When browsing a word index in the OPAC, special characters are always displayed in their converted state. So, if the `unicode_to_word_gen` table sets umlaut to ue, the word is displayed with ue, and not with an umlaut.

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

The "hyphen" and the apostrophe must be left with their actual value in the `alephe/unicode/unicode_to_word_gen` file, and both the hyphen and the apostrophe must not be entered in any of the word breaking procedures in the library's `tab/tab_word_breaking` file.

Use procedure 90 in `tab_word_breaking` when parsing a Word search query.

Use procedure 97 in `tab_word_breaking` in the ILL library for parsing records during the ILL Locate process.

9 Expand Routines, Tables and Indexing Expanded Fields

This chapter includes the following sections in order of appearance:

Expand Record

Expand Routines

Expand-Related Tables

Indexing Expand Fields (Virtual Fields)

9.1 Expand Record

The ALEPH integrated library system holds information in different types of records and in different types of "libraries" (databases).

A standard system has a BIB library for storing bibliographic data and an ADM library for storing administrative data. Most installations will also have an AUT library for storing authority information and a HOL library for storing holdings and location information.

Libraries may want to display information from the non-bibliographic databases together with the linked bibliographic record. Libraries may also want to enable the user to search the bibliographic database using information from other databases, such as location.

ALEPH enables the installation to "expand" information from one database record to another. This is possible because there are links between the records in the various databases. The information that is "expanded" can be used for display and/or for indexing. "Expand" routines can also be used to "expand" data within a record.

The "expand" function works with the tab_expand table located in the library's tab directory. Every library has such a table, although it is the bibliographic library that uses the table the most.

The tab_expand table defines three aspects:

The system function in which the expand program works.

The expand program that defines which data from the record can be expanded.

Additional parameters for the expand program, if required.

Following is a sample of the `tab_expand` table:

```
!   1                   2                                   3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC    expand_doc_fmt
U39-DOC    expand_doc_join
```

Key to tab_expand table:

Column 1 - System context
This column contains the "context" in which the expand is operative.

The available "context" functions are:

ACC
Context: Headings index creation/update (p_manage_02).

BIBHOL-MAB

Context: Use in the holdings library in MAB format with `expand_doc_bib_hol_mab`. Displays holding expanded fields in the bibliographic record.

BUF-Z403

Context: Used to enrich the BUF-Z403 , which is used to govern access to electronic resources. This is required when, for example, expanding the 856 field, or any other field with electronic contents, into the holdings library. Note that expand_doc_bib_z403 expand procedure may not be used with expand menu BUF-Z403.

CREATE-Z13
Context: Short bibliographic record creation/update (p_manage_07).

E-DOC-<format number>
Context: Specific format display. Used for running expand programs that should be applied only to specific formats. For example, the `expand_doc_uni_merge` program should be functional only when the record is displayed in ISBD format.

The format number of the instance should match the format number defined in the `edit_doc.lng` table for the desired format. For the `expand_doc_uni_merge` example mentioned above, if the ISBD format has been defined as 038 in the `edit_doc.lng` table, then the `tab_expand` table should be defined as follows:

GUI-ACCREF
Context: Authority record display from bibliographic heading (Search module).

GUI-BRIEF
Context: Brief display (Search module).

GUI-DOC-D
Context: Full display (Search module).

GUI-DOC-P
Context: Full print (Search module).

HOL-LOC
Context: Use in the holdings library with `expand_doc_hol_loc_1_a` and

`expand_doc_hol_loc_2_a`.

HOLDING
Context: Display of item list.

INDEX
Context: Direct index creation/update (p_manage_05).

PRE-MERGE

Context: Adds expanded fields to the merged display of a record in Union catalog/view

PRINT-CAT
Context: Print catalog (p_print_04).

PRINT-CUST
Context: Print custom format (p_print_01).

PRINT-COL
Context: Print columnar format (p_print_08).

PRINT-REC

Context: Print Catalog Records with "Non-preferred" Headings (print-05).

RET
Context: Retrieval of records (p_ret_01) and sorting (p_ret_21).

SORT-DOC
Context: Sort keys creation/update (p_manage_27).

U39-DOC
Context: Record display through UTIL F/4/DOC.

UE-08
Context: UE-08 (for expanding authority records for UE-08 procedures).

WEB-ACCREF
Context: Authority record display from bibliographic heading (Web OPAC).

WEB-BRIEF
Context: Brief display (Web OPAC).

WEB-FULL
Context: Full display (Web OPAC).

WEB-FULL-1
Context: Full display - format 01 (Web OPAC).

WEB-MAIL
Context: Full print - mail (Web OPAC).

WEB-SCNIND
Context: Title display when browsing Direct indexes (Web OPAC).

WORD
Context: Word index creation/update (p_manage_01).

X-AVAIL

Context: Retrieves the current availability status of a document (can be expanded using "expand_doc_bib_avail" and parameters in Col.3 of tab_expand).

Z00R
Context: Creation of a Z00R record (p_manage_07, cataloging, UE_01).

Z39-SERVER
Context: Z39-SERVER.

Column 2 - Expand program
The expand program that defines which data is expanded.

Column 3 - Parameters
Certain expand programs require additional information, such as field codes. This column is used to define additional parameters for expand programs. Note that the documentation for each expand program indicates whether or not parameters are needed (see for example, expand_doc_sort_field).

9.2 Expand Routines

The following are the available expand routines:

expand_doc_acronym_title on page 52

expand_doc_adm_bib on page 53

expand_doc_adm_hol on page 53

expand_doc_aut_aut on page 54

expand_doc_bib_001on page 54

expand_doc_bib_852_1 on page 54

expand_doc_bib_852_title on page 55

expand_doc_bib_880_n on page 55

expand_doc_bib_accref on page 55

expand_doc_bib_accref_1 on page 56

expand_doc_bib_adm on page 56

expand_doc_bib_avail on page 57
expand_doc_bib_avail_hol on page 58
expand_doc_bib_hol on page 60
expand_doc_bib_hol_ana on page 60
expand_doc_bib_hol_usm on page 61
expand_doc_bib_hol_usm_2 on page 61
expand_doc_bib_inv on page 61
expand_doc_bib_lng_cjk on page 61
expand_doc_bib_loc_1_a on page 61
expand_doc_bib_loc_1_b on page 61
expand_doc_bib_loc_1_b2 on page 61
expand_doc_bib_loc_1_c on page 62
expand_doc_bib_loc_1_c2 on page 62
expand_doc_bib_loc_3_a on page 62
expand_doc_bib_loc_4_a on page 62
expand_doc_bib_loc_4_b on page 62
expand_doc_bib_loc_4_c on page 62
expand_doc_bib_loc_5_c on page 62
expand_doc_bib_loc_cleanup on page 62
expand_doc_bib_loc_dedup on page62
expand_doc_bib_loc_disp on page 63
expand_doc_bib_loc_n on page 63
expand_doc_bib_loc_usm on page 65
expand_doc_bib_local_notes on page 67
expand_doc_bib_multi_lng on page 68
expand_doc_bib_ndu on page 69
expand_doc_bib_psts on page 69
expand_doc_bib_psts_disp on page 70
expand_doc_bib_subtype on page 70
expand_doc_bib_tab04 on page 70
expand_doc_bib_z30 on page 71
expand_doc_bib_z30 on page 71
expand_doc_bnu_initials on page 72
expand_doc_course on page 72
expand_doc_crs_bib on page 72

expand_doc_date_yrr on page 73
expand_doc_del_fields on page 73
expand_doc_deleted on page 73
expand_doc_duplicate_field on page 73
expand_doc_extract on page 73
expand_doc_extract_holding on page 74
expand_doc_fix_abbreviation on page 74
expand_doc_fmt on page 75
expand_doc_fmt_mgu on page 76
expand_doc_hld_stmt on page 76
expand_doc_hol_852_disp on page 79
expand_doc_hol_86x on page 79
expand_doc_hol_bib on page 81
expand_doc_hol_loc_1_a on page 82
expand_doc_hol_loc_2_a on page 82
expand_doc_hol_z30_86x on page 83
expand_doc_isbn_13 on page 83
expand_doc_ismn_13 on page 83
expand_doc_issn_isbn on page 83
expand_doc_join on page 84
expand_doc_join_filter on page 85
expand_doc_join_permute on page 85
expand_doc_join_simple on page 86
expand_doc_last_cat on page 86
expand_doc_link_to_doc on page 87
expand_doc_link_to_ros on page 87
expand_doc_open_cat on page 87
expand_doc_own on page 87
expand_doc_primo_plk on page 87
expand_doc_ros_id on page 87
expand_doc_rotate on page 87
expand_doc_section on page 87
expand_doc_sort on page 88
expand_doc_sort_field on page 88
expand_doc_sort_loc_a on page 88

expand_doc_sort_loc_b on page 88

expand_doc_sort_loc_x on page 63

expand_doc_split on page 89

expand_doc_split_external on page 89

expand_doc_split_sub1 on page 90

expand_doc_sysno on page 90

expand_doc_type on page 91

expand_doc_uni_merge on page 92

expand_doc_union_add_852 on page 92

expand_doc_union_exclude_lib on page 92

expand_doc_yr on page 92

Note that some expand programs have suffixes like *usm* and *mab*. This convention is used for expand programs dependent on the MARC format (such as MARC21, MAB, UNIMARC, and so on).

---

expand_doc_acronym_title

The expand_doc_acronym_title routine facilitates acronymic indexing of long titles that have common words (for example, Report of the …Association). This expand creates a new index which will help users search more easily for serial titles. It takes the first four letters from the first word of the title connected to the first three letters of the second word, connected to the first two letters of the third word, connected to the first letter of the fourth word. If the words are short, only the existing letters will be used. For example: The title “Gen. hosp psych” will be indexed as genhosps.

Stop words such as "the”, "and”, or "to", which reside in tab03 in the bibliographic library are not considered when building the "Acronymic Title".

This expand should be used when WORD indexing is performed.

Column 3 needs to be defined to include the record format, field and subfield to index and the tab_word_breaking procedure that will be used.

```
!   1                   2                            3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD       expand_doc_acronym_title       SE,245##a,92
WORD       expand_doc_acronym_title       BK,245##a,92
```

This expand should use a tab_word_breaking procedure such as the following:

```
  92 # del_subfield
  92 # to_blank             !@#$%^()_={}[]:";<>,.?|\
  92 # compress             '
  92 # to_lower
```

Note that the word breaking routine number in column 1 is definable.

In this following example, a word index was created to support searching/browsing:

tab11_word:

```
! 1      2    3       4              5               6  7 8  9    10    11
12
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!-!-!-
!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!
ACT##                     a                      92      WAC
```

tab00.lng:

```
!   2      3    4 5  6  7 8   9   10             11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H WAC   W-026       00        00        W-act.
```

expand_doc_adm_bib

The `expand_doc_adm_bib` program adds bibliographic data to the administrative record.

To include or exclude specific BIB fields in the expanded ADM record, use column 3 of `tab_expand` of the ADM library.

The parameters in Column 3 of `tab_expand` are up to 10 comma-separated fields (five characters each, can include hashes, for example 100##), to be included or excluded in the ADM record.

To exclude specific fields, the list of fields to be excluded should be preceded by a dash - , for example -245##,100##.

If column 3 is blank, then ALL fields from the BIB record are added to the ADM record.

In the following example, the 260## and 100## fields will be excluded from the expanded ADM record

```
!   1                    2                                3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_adm_bib              -260##,100##
```

Note that this expand program must be defined in the `tab_expand` table of the administrative library (XXX50). This can be useful for creating short_doc (Z13) (using manage-07) in the ADM library.

Note that this expand routine's behavior is the same as expand_doc_hol_bib.

expand_doc_adm_hol

The `expand_doc_adm_hol` expand routine is used in the `tab_expand` table of an ADM library in order to add HOL data to the ADM record.

Parameters in column 3 of tab_expand can be up to 10 comma-separated fields (five characters each, can include hashes, for example, 85###). If column 3 is blank, then all fields from the HOL record are added to the ADM record.

To exclude a specific field, precede the list of fields with a dash - , (for example, -85###, -86###).

Example of `expand_doc_adm_hol` in `tab_expand` (ADM library):

```
!   1                    2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_adm_hol             85###,86###
```

If "MERGE-TYPE=" is used in Column 3 of `tab_expand` for `expand_doc_adm_hol`, it is possible that more than one HOL record will be merged into the ADM record. For example, in `expand_doc_adm_hol`, if more than one HOL record is linked to the ADM record, the first HOL record is merged into the ADM record and then the second HOL record is merged into the already merged ADM record, and so on.

Note that this expand routine's behavior is the same as expand_doc_hol_bib.

expand_doc_aut_aut

For multilingual applications, the `expand_doc_aut_aut` program identifies the authority record of a heading that is a "see also" in the authority record. The program adds all forms of the heading from the "main" authority record. This program builds the "See also" field for all languages for Broader, Narrower, and "See also" terms.

expand_doc_bib_001

The `expand_doc_bib_001` program builds a 001 field that contains the system number of the record. The field is built only if the 001 field does not already exist in the record.

expand_doc_bib_852_1

The `expand_doc_bib_852_1` program expands the 852 MARC21 location field into the bibliographic record. The field is brought from the holdings record and/or built from the information in the Z30 (item record). If the holdings record has an 866 MARC21 field (textual holdings statement), the field is appended to the 852 field from the holdings record that is expanded into the bibliographic record.

Sublibrary (subfield $b) and collection (subfield $c) codes are expanded into subfields $4 and $5 in which the sublibrary code and collection code are replaced by names using the `tab_sub_library` (sublibrary definitions) and `tab40` (collection definitions) tables. Items barcode are expanded into 852$$p.

The second call number (Z30-CALL-NO-2) is expanded using the same subfield definitions used for expanding the regular call number (Z30-CALL-NO), but in uppercase. For example:

```
852  $$bULINC$$cGEN$$HHG939.5 D38 1970$$bLincoln
Library$$cGeneral$$HHG939.5 D38 1970
```

You can define the field, subfield and subfield contents to filter the holdings records that are expanded. This can be done by defining the field, subfield and subfield contents in the parameters column (col. 3) of the library's tab_expand table. For example, if the tab_expand table contains the following line:

```
PRINT-REC  expand_doc_bib_852_1           852##,b,ULINC
```

When the holdings information is expanded into the bibliographic record, the holdings data is included only if subfield $b of the 852 field contains the value 'ULINC'. Holdings records that do not match this definition are not included.

The format for the filtering definitions is the following:

```
FIELD,Subfield,CONTENTS
```

To prevent item barcodes from being expanded into field 852 when using expand_doc_bib_852_1, use the "BARCODE=N" parameter.
Note that this parameters must be sent as the fourth parameter, even if the other three parameters are not used.

For example:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
GUI-DOC-D  expand_doc_bib_852_1           ,,,BARCODE=N
GUI-DOC-D  expand_doc_bib_852_1           852##,b,WID,BARCODE=N
```

expand_doc_bib_852_title

The `expand_doc_bib_852_title` program expands the subfields $a, $b, $c of the Main Title field (245 MARC21 field) into the 852 MARC21 field ($$a, $$b, $$c) and creates a new CLN field. For example, if the bibliographic record contains the following fields:

```
24510 L $$aWall shadows; :$$ba study in American prisons,
852   L $$bUELEC$$hhE183.8.B7$$iL494
```

then the `expand_doc_bib_852_title` program creates the following new CLN field:

```
CLN01 L $$hhE183.8.B7$$iL494$$1Wall shadows; : $$2a study in
American prisons,
```

If there are multiple 852 fields, then each one is taken and the same 245 field is added to each.

expand_doc_bib_880_n

The `expand_doc_bib_880_n` program creates two new concatenated fields out of 2 fields linked by subfield $6. Subfield $6 contains data that links fields that are different script representations of each other. The two new fields are concatenated once as [romanized form] + [vernacular form], and once as [vernacular form] + [romanized form]. Concatenation is subfield by subfield and subfield codes are retained, differentiated by lowercase and uppercase. For example, if the record has the following linked fields:

```
245 10 $601$aSosei to kako$bNihon Sosei Kako Gakkai shi.

245 10 $601$1$a<Title in Japanese script>$b<Subtitle in
Japanese script>.
```

then the `expand_doc_bib_880_n` program creates the following two new virtual fields:

```
245 10 $$603$$aSosei to kako$$A<Title in Japanese
script>:$$bNihon Sosei Kako Gakkai shi.$$B<Subtitle in
Japanese script>.

245 10 $$603$$a<Title in Japanese
script>$$ASosei to kako$$b<Subtitle in
Japanese script>.$$BNihon Sosei Kako Gakkai shi.
```

expand_doc_bib_accref

The `expand_doc_bib_accref` program adds non-preferred terms to the bibliographic

record in order to build word entries from cross-references. This feature allows the user to perform a "find" search on preferred or non-preferred terms with the same result.

The `expand_doc_bib_accref` should only be used with the WORD system function.

If ADDITIONAL-INFO is sent as a parameter to `expand_doc_bib_accref`, the following additional information is added for each heading (the original authorized field and the added terms) in all formats (MARC21, MAB, UNIMARC and DANMARC):

Aleph Authority Record ID:

- If a related authority record exists in Aleph, a subfield of the authority record ID is added in the following format:

```
<Library><doc number>
```

For example:

```
USM10000000123
```

- The subfield where the authority record ID is added is different for each format:
    - MARC format: the authority record ID is added in subfield 0
    - UNIMARC format: the authority record ID is added in subfield 3
    - MAB format: the authority record ID is added in subfield 9
    - DANMARC format: the authority record ID is added in subfield I
- If the original field already has an authority record ID stored in a subfield as described above, it is removed.

preferred/non-preferred indicator:

- If related authority record exists in Aleph, subfield P is added with an indication if the term is the preferred term or not:
    - For non-preferred terms, the value of subfield P is N
    - For preferred terms, the value of subfield P is Y
- If no related authority record exists in Aleph, subfield P is not added.

expand_doc_bib_accref_1

This expand program works like expand_doc_bib_accref. The difference is that the cross reference information expanded by expand_doc_bib_accref_1 is put into lines named after the acc code of the relevant Z01 record of the bibliographic library.

If, for example, the acc code of the relevant Z01 record is AUT then the line to which the information is expanded is called: AUT. The expand_doc_bib_accref_1 should only be used with the WORD system function.

expand_doc_bib_adm

This expand program takes the ADM record fields and expands them in the connected bibliographic record.

Note that the expand works only in a single ADM environment.

expand_doc_bib_avail

The expand program, expand_doc_bib_avail, brings items and holdings availability information. This program relies also on the special consortia item record (z300).

The expanded information is presented in 'AVA' field which has the following sub fields:

$$a ADM library code

$$b Sub library code

$$c Collection text – If there are several items in different collections in one sub library, only the first collection of the sub library is presented.

$$d Call number – If there are several items in different collections in one sub-library, only the first call number in the collection is presented.

$$e Availability status – Can be 'available', 'unavailable', or 'check_holdings'. Available status is assigned if the total number of items minus unavailable items is positive. Unavailable is assigned if the total number of items minus unavailable items is zero or negative. If a record has no linked items (only Holdings records) the status is 'available'. This subfield may be affected by the value of the THRESHOLD parameter in column 3 of tab_expand.

$$f Number of items (for the entire sub library not just location).

$$g Number of unavailable items (for the entire sub library not just location).

$$h Multi-volume flag (Y/N) – If first item's Z30-ENUMERATION-A is not blank or 0 then =Y otherwise = N.

$$i Number of loans (for the entire sub library not just location).

$$j Collection code of the item.

$$k including the call number type as following:

- If the AVA field is built from holding information:
    - If the first indicator of field 852 in the holdings record is 7, the value of 852 subfield $$2 is copied to AVA$$k.
    - If the first indicator of field 852 in the holdings record is not 7, it is copied to AVA$$k.
- If the AVA field is built from item information:
    - If the item's call number type (Z30-CALL-NO-TYPE) is 7, the value of subfield $$2 of the item's call number (Z30-CALL-NO) is copied to AVA$$k.
    - If the item's call number type (Z30-CALL-NO-TYPE) is not 7, it is copied to AVA$$k.

Note that the above description is applied for all the formats (MARC, UNIMARC, DANMARC, and MAB).

$$p A number that represent the priority of the item by its location. The expand_doc_bib_avail routine consults ./bib_lib/tab/ava_location_priority and AVA$$p is created with a number that represents the location priority.

ava_location_priority list sublibrary and collection by their priority (the items at the top of the list have higher priority).

If there is no match with the ava_location_priority table, no subfield p is created.

$$t Contains a translation of the content of subfield 'e' according to the text of the messages entered in table ./errror_lng/expand_doc_bib_avail.

An item is unavailable if it matches one of the following conditions:

- It is on loan (it has a Z36).
- It is on hold shelf (it has Z37_status=S).
- It has a processing status and does not have a value in Z30-DEPOSITORY-ID.

Items with process statuses are considered ‘unavailable’. Note that Col.3 of tab_expand can be used to specify item process statuses that their items should be treated as ‘available’.

To ignore item process statuses that should be treated as ‘available’, set the following parameter in column 3 of tab_expand: AVA=BD,MK. It is possible to specify more than one process statuses, delimited by a comma “,”.

If you want to take reshelving time into account; set the following parameter in column 3 of tab_expand: RESHELVING=Y. This is mostly relevant for Real Time Availability functionality (availability X service).

To retrieve the availability information by collection (in addition to sublibrary), set the following parameter in column 3 of tab_expand: COLLECTION=Y.

To define the maximum number of items to check per sublibrary, set the following parameter in column 3 of tab_expand: THRESHOLD=080. In this example, the maximum number of items per sublibrary is 80. Note that the number set in this parameter must have three digits. This parameter affects the content of subfield $$e (Availability status) and sub field $$t (translation of $$e).

expand_doc_bib_avail_hol

The expand program, `expand_doc_bib_avail_hol`, brings holdings and item availability information, based on the Holding records 852 field and subfields.

For each HOL record, an AVA line is created with the holding information and its availability.

The expanded information is presented in the AVA field which has the following sub fields:

$$a ADM library code

$$b Sub library code, based on the 852$$b of the HOL record

$$c Collection text, based on the 852$$c of the HOL record

$$d Call Number - The HOL record’s 852 subfields which are set in aleph_start variable: correct_852_subfields (can be 1 or more of the following subfields: hijklm)

$$e Availability status - Can be “available”, “unavailable”,“check_holdings”, or “temporary_location”. Available status is assigned if the total number of items minus unavailable items is positive. Unavailable is assigned if the total number of items minus unavailable items is zero or negative. If a record has no linked items (only

Holdings records) the status is “check_holdings”. If all the items linked to the Holding records are in a temporary location the status is “temporary_location”. This subfield can be affected by the value of the THRESHOLD parameter in column 3 of tab_expand.

$$f The number of Items that are linked to the HOL record. If no items are linked to the HOL record, it is set to 0.

$$g The number of unavailable items (for the entire sub library not just location) that are linked to the HOL record. If no items are linked to the HOL record, it is set to 0.

$$h Multi-volume flag (Y/N) – If the first item’s Z30-ENUMERATION-A is not blank or 0 then =Y otherwise = N.

$$i The number of loans (for the entire sub library not just location). Based on the HOL record’s linked items. If no item is linked, it is set to 0.

$$j Collection code (852$$c of the HOL record).

$$k Call Number type 1st indicator of 852. If the first indicator of field 852 in the holdings record is 7, the value of 852 subfield $$2 is copied to AVA$$k.

$$p Location priority. A number that represent the priority of the item by its location.

The expand_doc_bib_avail_hol routine consults ./bib_lib/tab/ava_location_priority and AVA$$p is created with a number that represents the location priority.

If there is no match with the ava_location_priority table, no subfield p is created.

If there are 2 HOL records, for example, with the same sublibrary + collection values, then two AVA$$p subfilelds are created with the same priority rank.

$$t Availability text translation. Contains a translation of the content of subfield 'e' (available status), according to the text of the messages entered in table ./errror_lng/expand_doc_bib_avail (the same one as used by expand_doc_bib_avail).

$$7 Holdings ID - Contains the HOL library code and the HOL record number (e.g. USM60000000741) for which the AVA is created. Non-relevant for AVA fields of items with no linked HOL record.

Routine Additional Parameters

To define additional subfields added to the AVA fields from the 852 field of the HOL, set the parameter `SF` in column 3 of tab_expand (for example: `SF=z,t`) to copy subfields $$z and $$t. Note that the additional subfields that are copied from the HOL record 852 to the AVA field override the subfields created this expand program.

To define the maximum number of items to check per sublibrary, set the following parameter in column 3 of tab_expand: `THRESHOLD=080`. In this example, the maximum number of items per sublibrary is 80. Note that the number set in this parameter must have three digits. This parameter affects the content of subfield $$e (availability status) and sub field $$t (translation of $$e).

Items with process statuses are considered “unavailable”. Note that Col.3 of tab_expand can be used to specify item process statuses that their items should be treated as “available”.

To ignore item process statuses that should be treated as “available”, set the following parameter in column 3 of tab_expand: `AVA=BD,MK`. It is possible to specify more than one process statuses, delimited by a comma “,”.

If you want to take reshelving time into account; set the following parameter in column 3 of tab_expand: `RESHELVING=Y`. This is mostly relevant for Real Time Availability functionality (availability X service).

For example:

```
!   1                   2                              3
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
FULLP      expand_doc_bib_avail_hol       THRESHOLD=050;AVA=BD,NW;SF=z,t
```

In this example:
The THRESHOLD parameter limits the number of items per sublibrary to 50.
The AVA parameter defines items with process status BD or NW as available.
The SF parameter adds 852$$z and 852$$t to the AVA$$z and AVA$$t. The SF parameter supports multiple subfield occurrences.

Note:

For items that are not related to any HOL records, an AVA field is created for each sublibrary and collection combination similar to expand_doc_bib_avail, as if `COLLECTION=Y` is defined and without consulting the `SF` parameter.

expand_doc_bib_hol

The `expand_doc_bib_hol` program adds holdings data (the holdings record) to the bibliographic record.

Note that this expand routine's behavior is the same as `expand_doc_hol_bib`.

The program arguments (column 3 in tab_expand) are:

1. MERGE_TYPE=merge no. Note that it is possible for more than one HOL record to be merged into the ADM or BIB record.

2. SUP-HOL=Y/N. To enable the expanding of all holdings fields into the linked bib document even when the holding record has the STA=SUPPRESSED field.

3. Fields such as 856.

expand_doc_bib_hol_ana

the expand_doc_bib_hol_ana program finds an LKR field in the BIB record with subfield 'a' = "ANA", takes the related BIB record number from subfield 'b', and expands its Holdings record information in the same way as expand_doc_bib_hol works.
You can use col. 3 in `tab_expand` for parameter definition, just as in `expand_doc_bib_hol`.

Here is an example:

BIB record 10, has an LKR field, $$aANA$$b000000020.

BIB record 20, has a HOL record number 30.

The result of applying this expand program to BIB record 10 is that fields from HOL 30 will be added to it.

expand_doc_bib_hol_usm

The `expand_doc_bib_hol_usm` program takes the 866 MARC21 field (textual holdings - basic bibliographic unit) of the holdings record and concatenates it with the 852 MARC21 field (location) of the holdings record, creating a new 866 field. In addition to the new 866 field, the program adds the holdings record to the bibliographic record.

Note that if the holdings record has field STA, the record is displayed in the Web OPAC only if the field text is "DISPLAY". If the record does not have an STA field, the record will be displayed.

Note: To control the display of the 852 subfield $z in the 866 field, column 3 of tab_expand should include the parameter SUPPRESS_SF_Z as in the following example:

```
!   1                    2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_bib_hol_usm         SUPPRESS_SF_Z
```

expand_doc_bib_hol_usm_2

The `expand_doc_bib_hol_usm_2` program takes the 866 MARC21 field (textual holdings - basic bibliographic unit) of the holdings record and concatenates it with the 852 MARC21 field (location) of the holdings record, creating a new 866 field.

expand_doc_bib_inv

The `expand_doc_bib_inv` program builds an INV field from the inventory number of the item record (Z30-INVENTORY-NUMBER). The structure of the INV field is:

```
INV $a [inventory number]
```

expand_doc_bib_lng_cjk

The `expand_doc_bib_lng_cjk` expand program adds subfield $$9 with the language code (chi, jpn, and kor) to each CJK field, in order to differentiate between Chinese, Japanese and Korean. This subfield facilitates separate indexing for CJK languages. The expand program adds the subfield to all fields that have CJK characters.

expand_doc_bib_loc_1_a

This expand imports the holdings library code (XXX60), the holdings system number and the indicators and subfields of the MARC21 location field (852) into the temporary PS1 field.

expand_doc_bib_loc_1_b

This expand imports items into the temporary PS1 field using links of type ADM. This should be used in the bibliographic library (XXX01) and in Course Reading libraries (XXX30). If the parameter 'HOL-LIBRARY=N' is set for this expand, the PS1 fields are created even if the expanded BIB record is not connected to any HOL library. In this case, the ALEPH string is used instead of the value of the HOL library in the PS1$$r subfield.

expand_doc_bib_loc_1_b2

This expand imports items into the temporary PS1 field using links of type ITM. The expand should be used in Course Reading libraries (XXX30) and in any regular bibliographic library that uses ITM links (for example, analytical records).

Note that for expand_doc_bib_loc_1_b and expand_doc_bib_loc_1_b2 the Z16 is not included as in expand_doc_bib_loc_usm and expand_doc_bib_psts.

expand_doc_bib_loc_1_c

In order to set priorities for processing status over item status, this expand stores the item process in subfield $e. If the item is not in process, this expand routine takes the loan status of the item and stores it in subfield $d.

Creates PST directly from the holdings record, bypassing the creation of the temporary PS1, if there are no items linked to the holdings record.

This program should be used for sites where the items and the holdings records are linked.

expand_doc_bib_loc_1_c2

This program is like the `expand_doc_bib_loc_1_c` program except that it does not create the PST directly from the holdings record if there are no items linked to the holdings record.

This program should be used for sites where the items and the holdings records are not linked.

expand_doc_bib_loc_3_a

This expand program adds the following subfields (replaces codes by names) for display purposes:
$3 - Material type (display form)
$4 - Sublibrary name
$5 - Collection name
$6 - Item loan status (display form)
$7 - Item process status (display form)

expand_doc_bib_loc_4_a

Imitates `expand_doc_bib_loc_usm` creating a LOC field.

expand_doc_bib_loc_4_b

Imitates `expand_doc_bib_psts` creating a PSTS field.

expand_doc_bib_loc_4_c

Imitates `expand_doc_bib_loc_usm` creating the SBL, LOC and STS fields for linked item and holdings records.

expand_doc_bib_loc_5_c

Imitates `expand_doc_bib_loc_usm` creating the SBL, LOC and STS fields for linked subscription records.

expand_doc_bib_loc_cleanup

This program removes the intermediate PS1 fields.

expand_doc_bib_loc_dedup

The expand_doc_bib_loc_dedup program prevents duplicate locations for serials with subscriptions containing the same call number as the items. To prevent duplicate locations, a line with expand_doc_bib_loc_dedup must be added to tab_expand, after the line with expand_doc_bib_loc_usm.

expand_doc_bib_loc_disp

The `expand_doc_bib_loc_disp` program expands subfields $b, $c and $o of the LOC field created by `expand_doc_bib_loc_usm`, adding subfields $4 (sublibrary), $5 (collection) and $3 (material type) in which the codes are replaced by names.

expand_doc_bib_loc_n and expand_doc_sort_loc_x

The following expand programs are used to include location information in bibliographic indexes and displays. They are a modular set of expand programs that integrate the functionality of expand_doc_bib_loc_usm and expand_doc_bib_psts.

This expand mechanism generates intermediate PS1 fields; the PS1's are sorted and deduplicated into PST fields. Codes (for example, sublibrary) in the PST fields are expanded into display forms.

Structure of the PST field:

1st indicator: call number type (0-8).
2nd indicator: undefined, contains a blank.
$$0 [origin of the PST field].
If the field originates from an item record, then the subfield contains Z30 ($$0Z30).
If the field originates from a holdings record, then the subfield contains HOL ($$0HOL).
$$1 [unique identifier of the record of origin]
If the field originates from an item record, then the subfield contains the system number of the linked administrative record and the item sequence number (for example, $$11000005921000010). Format: <Z30-DOC-NUMBER> <Z30-ITEM-SEQUENCE>
If the field originates from a holdings record, then the subfield contains the holdings library code and the system number of the holdings record (for example, $$1USM60-000001909). Format: <library code>60;holdings system number>
$$b [sublibrary code].
$$c [collection code].
$$d [item status] if there is no item process status.
$$e [item process status] if there is an item process status in the item record.
$$f [temporary location flag].
If the sublibrary, collection and call number information are temporary (the Temporary Location box is checked), then the subfield contains Y ($$fY).
If the location is not temporary, then the subfield contains N ($$fN).
$$h [call number] if call number type is 0-3 or 6-8.
$$j [call number] if call number type is 4.
$$l [call number] if call number type is 5.
$$n [call number type]
$$o [material type - column 1 of the tab25.lng table] (for example, BOOK). Any item material type with first three letters ISS, such as ISSBD, will get $$0ISSUE.
$$r [linked holdings record]
Contains the holdings library code and the system number of the holdings record linked to the item (for example, $$1USM60-000001909). Format: <library code>-<holdings system number>
$$y [copy number]
$$3 [material type - display form: column 3 of the tab25.lng table] (for example,

Book).
$$4 [sublibrary name]
$$5 [collection name]
$$6 [item loan status - display form]
$$7 [item process status - display form]

The PST field is only created from the linked item and holdings records and not for the linked subscription records.

Following is the list of the programs:

expand_doc_bib_loc_1_a

expand_doc_bib_loc_1_b

expand_doc_bib_loc_1_b2

expand_doc_bib_loc_1_c

expand_doc_bib_loc_1_c2

expand_doc_sort_loc_a

expand_doc_sort_loc_b

expand_doc_bib_loc_3_a

expand_doc_bib_loc_4_a

expand_doc_bib_loc_4_b

expand_doc_bib_loc_4_c

expand_doc_bib_loc_5_c

expand_doc_bib_loc_cleanup

expand_doc_hol_loc_1_a

expand_doc_hol_loc_2_a

In addition, note that a Z07 record is triggered for the bibliographic record linked to the item when one of the following fields of the item record is updated:

Z30-BARCODE

Z30-SUB-LIBRARY

Z30-MATERIAL

Z30-ITEM-STATUS

Z30-COLLECTION

Z30-CALL-NO-TYPE

Z30-CALL-NO

Z30-CALL-NO-KEY

Z30-CALL-NO-2-TYPE

Z30-CALL-NO-2

Z30-CALL-NO-2-KEY

Z30-DESCRIPTION

Z30-INVENTORY-NUMBER

This ensures that the expanded bibliographic record is updated when information related to the item is changed.

The following is an example of the setup for a site where items are linked to holdings records:

```
!   1                       2
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
XXX-XXX     expand_doc_bib_loc_1_a
XXX-XXX     expand_doc_bib_loc_1_b
XXX-XXX     expand_doc_bib_loc_1_b2
XXX-XXX     expand_doc_bib_loc_1_c
XXX-XXX     expand_doc_sort_loc_b
XXX-XXX     expand_doc_bib_loc_2_a
XXX-XXX     expand_doc_bib_loc_3_a
XXX-XXX     expand_doc_bib_loc_cleanup
```

Following is an example of the setup for a site where items are not linked to holdings records:

```
!   1                       2
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
XXX-XXX     expand_doc_bib_loc_1_a
XXX-XXX     expand_doc_bib_loc_1_b
XXX-XXX     expand_doc_bib_loc_1_b2
XXX-XXX     expand_doc_bib_loc_1_c2
XXX-XXX     expand_doc_sort_loc_a
XXX-XXX     expand_doc_bib_loc_2_a
XXX-XXX     expand_doc_bib_loc_3_a
XXX-XXX     expand_doc_bib_loc_cleanup
```

expand_doc_bib_loc_usm

The `expand_doc_bib_loc_usm` program builds four fields from the Z30 (item record), the Z16 (subscription record), and the 852 field (location) of the holdings record: SBL, LOC, STS and PST.

Another field, HLD, is created based on PST fields with $$0HOL. It contains a link to the relevant items, Electronic Location, Summary Holdings Information, Index Holdings and Supplement Holdings. This field can be presented ONLY in the full view screen. It will not work in brief view.

There is a program argument, OPTIMIZE=N, which, when added to the 3rd column of `tab_expand`, causes all items to be expanded, including those with different call number information. When OPTIMIZE=N is used, the run time is not optimized.

Here is an example:

```
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD       expand_doc_bib_loc_usm         OPTIMIZE=N
```

In Course Reading/Reserves, expand_doc_bib_loc_usm works only for records whose xxx30 doc is linked to the item via a (z103) ADM-type link; it does not work for ITM-type links:
- the xxx30 professors' copies which are linked to the xxx50 item directly have a z103_lkr_type ADM;
- the xxx30 records which are linked to the xxx50 item via an xxx01 bib record have a z103_lkr_type ITM.

The expand_doc_bib_loc_usm is a hard-coded set of calls to expand_doc_bib_loc_n_x programs.
expand_doc_bib_loc_1_b reads items linked via an ADM-type link;
expand_doc_bib_loc_1_b2 reads items linked via an ITM-type link.
expand_doc_bib_loc_1_b2 is not one of the programs called by expand_doc_bib_loc_usm.

In order to include both LKR types (ADM and ITM), the xxx30 tab_expand needs to specify both _b and _b2:

```
CREATE-Z13 expand_doc_course
CREATE-Z13 expand_doc_bib_loc_1_a
CREATE-Z13 expand_doc_bib_loc_1_b
CREATE-Z13 expand_doc_bib_loc_1_b2
CREATE-Z13 expand_doc_bib_loc_1_c
CREATE-Z13 expand_doc_sort_loc_b
CREATE-Z13 expand_doc_bib_loc_2_a
CREATE-Z13 expand_doc_bib_loc_3_a
CREATE-Z13 expand_doc_bib_loc_4_a
```

Structure of the SBL field:

Indicators - both undefined, each contains a blank.
$a [sublibrary code]

Structure of the LOC field:

1st indicator: Call number type (0-8).
2nd indicator: undefined, contains a blank.
$b [sublibrary code]
$c [collection code]
$h [call number] if call number type is 0-3 or 6-8.
$j [call number] if call number type is 4.
$l [call number] if call number type is 5.
$o [material type]

Structure of the STS field:

Indicators - both undefined, each contains a blank.
$a [item status code]

This program uses the same environment variable that is used when ALEPH automatically updates the Z16 (subscription record) and the Z30 (item record) from the 852 field of the linked holdings record. The program only expands the subfields of the 852 field defined in the correct_852_subfields environment variable defined in the `aleph_start` file. In this way, call numbers from the item and the holdings record are treated consistently when they are merged into a single list during the expand.

Additionally, note that a Z07 record is triggered for the bibliographic record linked to the item when one of the following fields of the item record is updated:

- Z30-SUB-LIBRARY
- Z30-MATERIAL
- Z30-ITEM-STATUS
- Z30-COLLECTION
- Z30-CALL-NO-TYPE
- Z30-CALL-NO
- Z30-CALL-NO-KEY
- Z30-CALL-NO-2-TYPE
- Z30-CALL-NO-2
- Z30-CALL-NO-2-KEY
- Z30-DESCRIPTION
- Z30-INVENTORY-NUMBER

A Z07 for the bibliographic record is also triggered when the linked subscription record is updated.

This ensures that the expanded bibliographic record is updated when information related to the linked item or to the linked subscription information is changed.

Structure of the PST field:

See above.

expand_doc_bib_local_notes
This program is used to expand into the bibliographic record - for display and indexing purposes - local tags stored in the holdings record. The application for storing local tags in a holdings record is in a consortial environment where a single bibliographic record is shared by multiple institutions and an institution would like to include local tags that not everyone can see. Which local tags are moved to the holdings record from the bibliographic record (see the entry for `fix_doc_create_hol_local_notes`) and which local tags from the holdings records are displayed in the OPAC can be configured by the local institution.

When the records are displayed in their full format (WEB-FULL routine from `tab_expand`) or indexed, the program consults the `tab_base.conf` table of the `alephe` directory, and only the local tags of the selected base are indexed and displayed.

The following are the tables involved:

The `tab_expand` in the tab directory of the bibliographic library (XXX01):

Determines the instances in which the program is performed. The following is a sample of the setup needed for displaying the local tags from the holdings record together with the bibliographic record:

```
WEB-FULL   expand_doc_bib_local_notes
```

The tab_expand_local_notes.conf of the bibliographic library (XXX01):

This table is used to define the owner codes for the holdings records to display local tags in the OPAC. The table also defines the tags that are to be expanded. In the sample below, XXX01_AA will display the local tags that are in the holdings records where the OWN tag is AA. XXX01_PUB will display local tags for both AA and BB owners:

```
[XXX01_AA]
owners list = AA
owner tag = OWN
owner subfield = a
owner alternative tag = 590,690
owner alternative subfield = 9
mapping section = LCN-2-BIB

[XXX01_PUB]
owners list = AA,BB
owner tag = OWN
owner subfield = a
owner alternative tag = 590,690
owner alternative subfield = 9
mapping section = LCN-2-BIB
```

The `tab_mapping` table in the tab directory of the bibliographic library (XXX01):

This table is used to map the local tags into the new virtual tags to be indexed or displayed. In the sample below, if the holdings record has a 690 field, then the program expands this field into the virtual LCS field in the bibliographic record:

```
LCN-2-BIB       541## abcde LCN   abcde                                Y Y
LCN-2-BIB       541## fho39 LCN   fho39                                Y Y
LCN-2-BIB       561## ab39  LCN   ab39                                 Y Y
LCN-2-BIB       590## ab9   LCN   ab9                                  Y Y
LCN-2-BIB       690## ab9   LCS   ab9                                  Y Y
```

Note that for the indexing and/or display of the new virtual fields, it is necessary to further customize the standard display and indexing tables (for example, edit_doc_999.lng, tab11_word).

expand_doc_bib_multi_lng

For multilingual applications, the `expand_doc_bib_multi_lng` program adds other language fields to the record. The program adds all the headings (Z01 records) that are linked to the same authority record as the heading field and that are not cross-references.

Note that `expand_doc_bib_accref` includes `expand_doc_bib_multi_lng`. Therefore there is no need to have it listed under the WORD system function. The `expand_doc_bib_multi_lng` program must only be used with the ACC system function.

In order to populate subfield $$6 in the BIB record after its enrichment, set column 3 of the tab_expand table with the parameter `add_sf6=Y`.

expand_doc_bib_ndu
The `expand_doc_bib_ndu` program creates a virtual TIT5 field for the following MARC 21 fields:

130, 245, 730 (from all subfields)
240, 242, 243, 246, 770 (from subfield $a)
700, 710, 711, 773, 780, 785 (from subfield $t)

The program strips punctuation, capitalizes text and removes initial articles (if suppressed for filing).

expand_doc_bib_psts
The `expand_doc_bib_psts` program builds a PSTS field from the Z30 (item record), the Z16 (subscription record), and the 852 field of the holdings record. This routine shows the processing status of the item record if available, as well as the call number, collection and sublibrary.

Structure of the PSTS field:

Indicators - both undefined, each contains a blank:
$b [sublibrary code]
$c [collection code]
$h [call number] if call number type is 0-3 or 6-8.
$j [call number] if call number type is 4.
$l [call number] if call number type is 5.
$d [item status] if no item process status.
$e [item process status] if there is an item process status.

Note that the item process status is stored in subfield $e. If the item is not in process, the expand routine takes the loan status of the item and stores it in subfield $d.

This program uses the same environment variable that is used when ALEPH automatically updates the Z16 (subscription record) and the Z30 (item record) from the 852 field of the linked holdings record. The program only expands the subfields of the 852 field defined in the `correct_852_subfields` environment variable defined in the `aleph_start` file. In this way, call numbers from the item and the holdings record are treated consistently when they are merged into a single list during the expand.

Note that the `expand_doc_bib_psts` program extracts items linked to the bibliographic record through ADM and ITM links. It is necessary to extract ITM links so that the expand program will display items in a Course Reading library whether or not the ADM record is linked to the Course Reading document or to a bibliographic document.

This expand routine skips those holdings records that have been suppressed (STA$$aSUPPRESSED).

Additionally, note that a Z07 record is triggered for the bibliographic record linked to

the item when one of the following fields of the item record is updated:

- Z30-SUB-LIBRARY
- Z30-MATERIAL
- Z30-ITEM-STATUS
- Z30-COLLECTION
- Z30-CALL-NO-TYPE
- Z30-CALL-NO
- Z30-CALL-NO-KEY
- Z30-CALL-NO-2-TYPE
- Z30-CALL-NO-2
- Z30-CALL-NO-2-KEY
- Z30-DESCRIPTION
- Z30-INVENTORY-NUMBER

A Z07 for the bibliographic record is also triggered when the linked subscription record is updated.

This ensures that the expanded bibliographic record is updated when information related to the linked item or to the linked subscription information is changed.

expand_doc_bib_psts_disp
The `expand_doc_bib_psts_disp` program expands subfields $b and $c of the PSTS field created by `expand_doc_bib_psts`, adding subfields $4 (sublibrary name) and $5 (collection name) in which the codes are replaced by names.

Note that `expand_doc_bib_psts` is intended for indexing, while `expand_doc_bib_psts_disp` is intended for display.

expand_doc_bib_subtype

This expand routine retrieves a list of subject fields as a parameter and adds the subject type to each of them, as follows:

- If the second indicator is 4: The source is not specified. No changes are done.
- If the second indicator is 7: The source is already specified in subfield 2. No changes are done.
- If the second indicator is neither 4 nor 7, subfield 2 with the value of the second indication is added to the subject fields.
- If the field already has a value in subfield 2, it is removed (unless the second indication is 7 or 4).

expand_doc_bib_tab04
The `expand_doc_bib_tab04` program is primarily intended for the Z39_SERVER instance in the `tab_expand` table of the library's tab directory. This program can be used to translate alphabetic tags into numeric values (for example, LOC to 952). Note that the Z39 protocol does not recognize non-numeric tags and ALEPH fields (such as LOC, CAT, Z30, and so on) need to be converted using the `expand_doc_bib_tab04` program.

The program works with the `tab04` table of the library's `tab` directory. The `tab04` table is used to set up the specification for the conversion of one set of cataloging tags

to another. The expand_doc_bib_tab04 program uses entries defined under the conversion routine 90. Following is a sample from the tab04 table:

```
90 Z####            9#### N
90 CAT##            956   N
90 #####            ##### N
```

Note that the last line in this sample should always be present.

expand_doc_bib_z30

This program is used to expand the item's information into the bibliographic record. The expand_doc_bib_z30 program is used with the expand_doc_bib_z30 table of the library's tab directory. The expand program creates a new virtual field - Z30-1 (for copy items) or Z30-2 (for issue items) - that contains the item's information. The table is used to define which fields from the item record are expanded and to determine the subfield structure of the new expanded field.

The parameters column of the tab_expand table can be used in order to determine whether to consult the tab15.lng table, which table used be used to determine the structure of the new field and which field should be created. The following parameters are available:

- CONF - This option can be used in order to determine that a different table should be used instead of the default expand_doc_bib_z30 table. Following is a sample of the usage of the variable:

  ```
  !   1                   2                          3
  !!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!->
  U39-DOC    expand_doc_bib_z30             CONF=exl_z30
  ```

  If CONF is not set, the default expand_doc_bib_z30 table is used.

- TAB15 - This parameter can be used to specify whether or not column 10 of the tab15.lng table should be consulted. This column is used to specify whether the copy should be displayed in the Web OPAC. If the TAB15 variable is set to Y in the tab_expand table, then if column 10 of the tab15.lng table contains an 'N' for the type of items attached to the bibliographic record, the expand does not create new virtual fields for these items. The following options are available in order to specify that virtual fields should be created disregarding the specifications of the tab15.lng table:

  o Set TAB15 to N (TAB15=N) or it is not defined.
  o Column 3 of the tab_expand table contains the string ALL.

  Following is a sample line for the usage of this parameter:

  ```
  !   1                   2                          3
  !!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!->
  WEB-FULL   expand_doc_bib_z30             TAB15=Y
  ```

- TAG - This parameter can be used in order to specify that a different new expanded field should be created instead of the defaults Z30-1 (for copy items) or Z30-2 (for issue items).

  Following is a sample line for the usage of this parameter:

  ```
  !   1                   2                          3
  !!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!->
  ```

```
U39-DOC     expand_doc_bib_z30              TAG=LOC
```
If this parameter is not set, Z30-1 and Z30-2 will be used.

Note that this program should be used with special care as it can create performance problems, due to the potential overflow of the buffers. This program should be avoided by libraries that have numerous item records linked to one bibliographic record (usually due to the individual listing of serial issues).

expand_doc_bib_z403 (functional for ADAM)
The expand_doc_bib_z403 program can be used to expand the object's data information into the bibliographic record. The expand_doc_bib_z403 program is used with the expand_doc_bib_z403 table of the library's tab directory. The expand program creates a new virtual field, Z403, that contains the object's information. The table also determines the subfield structure of the new expanded field.

The following parameters can be specified in the tab_expand table in order to modify the program's defaults:

- TAG - This parameter can be used in order to specify that a different new expanded field should be created instead of the default Z403 field.
- USAGE-TYPE - This parameter can be used in order to specify the type of objects that are included in the expand window. By default, the program only expands information from objects where the Z403-USAGE-TYPE is set to VIEW. This parameter can be used to specify that other object types should be included. In order to include all types of Z403-USAGE-TYPE, the parameter can be set to ALL. In order to include more than one type but not all types you can either repeat the line in the tab_expand table or repeat the parameter in the same line (for example, USAGE-TYPE=INDEX,USAGE-TYPE=VIEW).
- CONF - This option can be used in order to determine that a different table should be used instead of the default expand_doc_bib_z403 table.
- DISPLAY-LINK - If this parameter is set to "N", the object's data is not expanded when the expiry date of the object has been reached or/and when the Show in OPAC flag is set to No.

expand_doc_bnu_initials
This expand program adds a virtual subfield $G to UNIMARC fields 701, 702, 711, 712 and 200. The virtual subfield is built from the pinyin translation of these fields which is stored in subfield $A. The subfield contains the initials of the contents of subfield $A (initials of the intellectual responsibility and/or title). Note that this expand program is to be used by Chinese installations only and enables the retrieval of records using initials.

expand_doc_course
The `expand_doc_course` program must be used for the implementation of Course Reading Management. The program should be present in the tab_expand table of the Course Reading library (XXX30) under all instances (system function).

expand_doc_crs_bib
The expand_doc_crs_bib is useful when the BIB record is expanded from another BIB record and the course record should have information from both the related BIB record and its expanded document. The program can be present in the tab_expand table of the Course Reading library (XXX30) under all instances (system function).

When setting expand_doc_crs_bib, define in col. 3 of tab_expand the fields that will be expanded to the course document.

For example: ./xxx30/tab/tab_expand

```
CREATE-Z13                  expand_doc_crs_bib              331-2,1##-2
```

expand_doc_date_yrr

The expand_doc_date_yrr expand routine facilitates indexing date ranges that are related to the BIB record. The expand creates YRR fields for the range of dates that are indicated in the Publication Date 1 and Publication Date 2 of USMARC records (positions 06,07-10 and 11-14 of field 008) and UNIMARC records (positions 08, 09-12 and 13-16 of subfield $$a of the 100 field).

expand_doc_del_fields

This expand program deletes all the fields in the record except the fields specified in Col.3 of tab_expand. The fields in Col.3 are separated by a comma.

Example:

```
!   1                   2                              3
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
TEST       expand_doc_del_fields          245##,260##,500##
```

Only 245##, 260##, 500## and AVA## fields will be retained.

expand_doc_deleted

This routine deletes all the expanded lines of a deleted record and should be added at the end of the sections relevant for indexing.

expand_doc_duplicate_field

The expand_doc_duplicate_field program is used to duplicate a field, assigning a new field tag plus indicators. The expand_doc_duplicate_field program is used with the tab_expand_duplicate_field table of the library's tab directory. This table defines the field to be duplicated and the new assigned field tag. For example, if the tab_expand_duplicate field contains the following line:

```
! 1   2
!!!!!-!!!!!
260## IMP
```

then the 260 field is duplicated and assigned to a new IMP field. The expand_doc_duplicate_field procedure and expand_doc_duplicate table can be used to overcome the problem created when using expand_doc_split, which does not retain the source field. When expand_doc_split is based on a field created by the expand_doc_duplicate_field program, the source field is retained. In the above example, the IMP field can be later processed by expand_doc_split without losing the original 260 field.

expand_doc_extract

The expand_doc_extract program is used with the tab_expand_extract table of the library's tab directory. This table defines extraction of subfields for indexing. For example, if the library wants to create a headings list of "chronological subdivisions" for "subject added entries - topical terms", it is possible to define that MARC21 650 field, subfield $y is to be expanded into a new tag (for example, y650). The virtual field may then be indexed or displayed.

Note that the fourth column in the tab_expand_extract table can be used to specify the number of subfield occurrences for which the new virtual field is created. For example, you can define that only the first occurrence of subfield $y in the 650 field should be used for the creation of the new field. The following is an extract from the table:

```
! 1   2  3    4
!!!!!-!-!!!!!-!
650## y Y650  1
```

expand_doc_extract_holding

This program moves the 852 field from the holdings record into the linked bibliographic record. In addition, the contents of subfields $a and $z of the 866/7/8 fields of the holdings records are added to the new 852 field under subfield $3. Note that between each additional subfield and field, three asterisks are inserted (***). Subfield $a of the new field is built based on the library code (first three positions, for example USM). For example, if the holdings record contains the following fields:

```
86631 L $$aav. 37-52$$zSome issues missing
```

a new 852 field is added to the linked bibliographic record as follows:

```
8520  L $$aUSM$$bUELEC$$hhE183.8.B7$$iL494$$&
#36;3av. 37-52***Some issues missing
```

The parameters column of the tab_expand table can include two program arguments: the first calls a section in tab_fix of the holdings library (XXX60) and the second lists the item processing status that should be considered as suppressed (the expand should skip the holdings records linked to the items). Note that in order to use this expand routine, a line containing ADM USM60 USM50 must be present in the library_relation table. For example, if tab_expand is defined as follows:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_extract_holding     HOL,OI,BP,CA,CL,CT
```

then HOL is a section from the tab_fix table of the holdings library and OI, BP, CA, CL and CT are suppressed item processing statuses. Up to 10 item process statuses can be defined.

In addition, the program does not expand holdings records where the STA field is set to "SUPPRESSED". Additionally, the program does not check records in SE format.

Note that the hash (#) character can be used as a wildcard and that you can specify the list to have values such as A# or #B. This will filter out all process status codes starting with A or ending in B.

expand_doc_fix_abbreviation

The expand_doc_fix_abbreviation program is used to change abbreviations into full text. The routine can be used to replace any text string in a record with a different text string. There are two options:

A new duplicate field is added to the record with the non-abbreviated form of the text.

The abbreviated form of the text is changed into full text in the original field.

In order to specify whether a new duplicate field is added to the record with the non-abbreviated form of the text or whether the abbreviated form of the text is replaced in the original field with the new form, use the parameters column in `tab_expand` (col. 3).

The available values are ADD and REPLACE. ADD adds a new field to the record with the non-abbreviated form and REPLACE replaces the abbreviated form with the new form in the original field without duplication. If the column is left blank, the system duplicates the field (as ADD). Following is a sample of the `tab_expand` table:

```
!   1                    2                                3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_fix_abbreviation    ADD
```

The `expand_doc_fix_abbreviation` program is used with the `tab_abbrev` table of the library's `tab` directory (see `tab_abbrev`). The `tab_abbrev` table contains the list of "abbreviations" and the whole forms into which the shortened form is to be changed in the new virtual field added to the record.

Following is a sample of the `tab_abbrev` table:

```
! 1   2         3                              4
!!!!!-!-!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
225## Y 1st            FIRST
043## Y u-at---        Australia
043## Y u-at-ne        Australia New South Wales
```

Based on the above sample, the following document:

```
043   L $$au-at-ne

1001  L $$aWilliams, David

2451  L $$aThe 1st man
```

is expanded into:

```
043   L $$au-at-ne

1001  L $$aWilliams, David

2451  L $$aThe 1st man

043   L $$aAustralia New South Wales

2451  L $$aThe FIRST man
```

Note that this program also can be used as a fix program (using `tab_fix` in place of `tab_expand`). The difference is that instead of adding virtual fields (for indexing or/and display), when used as a fix program, the fields with the whole forms are actually added to the record.

expand_doc_fmt

The `expand_doc_fmt` program builds a TYP field from the record's format (FMT field).

Structure of the TYP field:

Indicators - both undefined, each contains a blank:
$a [record's format code]
$b [record's format name]

expand_doc_fmt_mgu

The `expand_doc_fmt_mgu` program builds a TYP field indicating the type of record. The TYP field is created based on coding in the LDR (positions 06 and 07), 006 (position 00), 007 (positions 00 and 01), and 008 (position 23) fields.

Structure of the TYP field:

Indicators - both undefined, each contains a blank.
$a [<type of record>]

The TYP field is generated with one of the following:

```
<Electronic Resource>
<Web Resource>
<Microform>
<Serial>
<Electronic Journal>
<Web Journal>
<Microform Serial>
<Computer File>
<Map>
<Digital Map>
<Score>
<Sound>
<Archive/MSS>
<Visual>
<Graphic>
<Kit>
<Realia>
```

expand_doc_hld_stmt

The `expand_doc_hld_stmt` routine generates 863/4/5 enumeration and chronology data in HOL records, using item records that are linked to the HOL record. The fields that are generated can then be used to create compressed holdings statements for display.

As an expand routine, the compressed holdings statements are built on-the-fly when the record is displayed. Note that a similar program, fix_doc_hld_stmt, can be applied in the cataloging module (after setting the tab_fix and the fix_doc.eng table of the HOL library (XXX60). When the fix_doc_hld_stmt program is performed, the 863/4/5 fields are added to the HOL record.

This routine is relevant to HOL libraries only. Note that this routine is relevant only if the 85x/85xX Publication Pattern fields reside in the HOL document record.

The following item records are taken into consideration when generating the 863/4/5 fields:

Items that have HOL no. in the Hol. Link field.

Items for which the enumeration field is equal to or greater than the value entered in subfield $$a of S63/4/5 (Starting Point field) in the HOL record. If there is no S63/4/5 field, enumeration in the item record is not checked. See paragraph 5 for more details.

Item groups:
All items which share the same Linking Number identifier in $$8 and the same Copy-ID (if there is one) are taken into consideration for a single holdings statement; if one HOL record has more than one linked Copy-ID, each Copy-ID creates a separate holdings statement. See paragraph 5 for more details.

"Starting Point" and "Copy-ID":
A special field, S63/4/5 in the HOL record, defines the item's Starting Point and the Copy-ID for generating a holdings statement. This field should be used by libraries that already have holdings statement fields for item records. Use field S63/4/5 with subfield $$a for the volume's Starting Point and subfield $$t for Copy-ID.

For example: if the field "S63" contains in subfield $a the value "3", only items with the first enumeration level of "3" (or higher) will be included in the holdings statement.

If the field "S63" subfield $t contains the value "2", only items with the value "2" in the copy number field (Z30-COPY-ID) will be included in the generated holdings statement.

```
S63   L  $$a3 $$t2

C6340 L $$81.1$$a3$$b2-4$$i2002$$j03-07$$wg$$t2$$9Y
C6340 L $$81.2$$a4$$b1-2$$i2003$$j01-03$$wg$$t2$$9Y
C6340 L $$81.3$$a4$$b4-5$$i2003$$j07-09$$wg$$t2$$9Y
```

Note: it is possible to have several "S63" fields in the HOL record. In this case, a set of holdings statement (863/4/5) fields will be created for each "S63" field that includes subfield $t (copy-ID).

For example:

```
86340 L $$81.13$$a1$$i20000-2000$$t1$$9Y
86340 L $$81.14$$a2$$b1-3$$i2001$$j01-05$$wg$$t1$$9Y
86340 L $$81.15$$a2$$b5-6$$i2001$$j09-11$$t1$$9Y

86340 L $$82.13$$a1$$i2000$$t2$$ $$9Y
86340 L $$82.14$$a2$$b1$$i2001$$j01$$wg$$t2$$9Y
86340 L $$82.15$$a2$$b3$$i2001$$j05$$wg$$t2$$9Y
```

The following procedures are performed by the routine:

Holdings statements (863/4/5 fields) which have $$9 with the value "Y" are deleted.

(Note: $$9Y is used to denote that the holdings statement field should be regenerated each time the procedure is used. When a holdings statement field (863/4/5) is generated, it always includes $$9Y. If the library has holdings statement fields that it wants to retain when running the procedure, the fields should contain $$9N.

Sets of "considered" item records (as defined in section 4 above) are compressed to create ranges by enumeration and chronology.

For example:

```
86340 L $$82.1$$a1$$b1-6$$i2000$$j01-06$$t00001$$9Y
86340 L $$82.2$$a2$$i2001$$t00001$$9Y
```

Each range is written in a separate 863/4/5 field, with the correct break indicator in $$w, if needed. Subfield $$w with the value "g" indicates a gap break (i.e., there are issues that did not arrive). Subfield $$w with the value "n" indicates a non-gap break (i.e., there are issues that were not published).

For example:

```
C6340 L $$82.3$$a3$$b2-5$$i2002$$j03-09$$wg$$t1
```

In the above example, subfield $$w contains the value "g" to indicate that there is a gap break. Issues no. 1 and no. 6 of Volume no. 3 did not arrive.

Break indicator ($$w) is generated for items that are not included in the summary holdings statement. These are Items with the process status of: NA or NP, or items that were mapped to be considered as if they have processing status NA or NP, using the `tab_hld_stmt` table of the administrative library (XXX50).

Since different institutions might use different codes to describe the process-status Not Arrived or Not Published, `tab_hld_stmt` is used to map those different codes into "NA" or "NP".

By using this table, we can also map items that have a particular sublibrary and/or collection, and/or item status, and/or item process status, and/or break indicator to be considered as NA or NP. Those items will be excluded from the holdings statement.

For example, a library that might want to exclude missing items from the holdings statement should map the process status MI (missing) as NA in `tab_hld_stmt`, as follows:

```
! 1     2   3  4  5 6
!!!!!-!!!!!-!!-!!-!-!!
##### ##### ## MI # NA
```

`expand_doc_hld_stmt` should be listed in col.2 of `tab_expand` of the HOL library; for example:

```
!   1                    2                    3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!>
WEB-FULL   expand_doc_hld_stmt
WEB-FULL   expand_doc_hol_86x
```

The routine will create 863/4/5 tags in UTIL F/4/doc and in WEB FULL display.

Note: It is possible to define alternative field tags instead of 863/4/5 fields in order to differentiate between 863/4/5 fields already present in the record, and field tags generated by the routine.

For example, if `tab_expand` is defined as follows:

```
!   1                    2                    3
```

```
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_hld_stmt    CMPRS_TAG=C63
```

The holdings statement field tag will be C63:

```
C6340 L $$81.1$$a1$$i2000$$t2$$9Y
C6340 L $$81.2$$a2$$b1-5$$i2001$$j01-09$$wg$$t2$$9Y
C6340 L $$81.3$$a3$$b2-4$$i2002$$j03-07$$wg$$t2$$9Y
```

This feature can be used if and when you are checking the generated data and want to separate it from data that has actually been written in the record.

To enable the generation of summary holdings statements for item records that are not linked to a subscription record or do not hold a value in the Copy ID field in the subscription record, the parameter:

```
GET_Z30_BY=HOL
```

should be defined in col.3 of `tab_expand` as follows:

```
!   1                   2                            3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL   expand_doc_hld_stmt            GET_Z30_BY=HOL
```

For item records that are linked to a subscription record and which hold a value in the Copy ID field in the subscription record and the item records, the parameter:

```
GET_Z30_BY=COPY_ID
```

can be defined in col.3 of `tab_expand` as follows:

```
!   1                   2                            3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL   expand_doc_hld_stmt            GET_Z30_BY=COPY_ID
```

Note: If no parameter is defined in col.3 of `tab_expand` the system will use the default parameter:

```
GET_Z30_BY=COPY ID
```

expand_doc_hol_852_disp

The `expand_doc_hol_852_disp` program expands subfields $b and $c of the 852 MARC21 location field adding subfields $4 and $5 in which the sublibrary code and collection code are replaced by names.

expand_doc_hol_86x

The `expand_doc_hol_86x` program is used to create the textual holdings statement fields (866, 867 and 868) based on the 853-855 (Captions and Pattern) and 863-865 (Enumeration and Chronology) fields from the holdings record.

The textual holdings statement fields are created according to the ANSI/NISO Z39.71 - 2006 standard (Holdings Statements for Bibliographic Items) with stripped punctuation.

expand_doc_hol_86x creates the text holding statement so that enumeration and chronology data are displayed separately with or without parentheses. This form of display is called Separate. The following is an example of a separate display:

```
vol.10:no.2-12(2003:Feb.-Dec.)
```

If the holdings record contains the following fields:

```
85320 L $$81$$av.$$bno.$$u9$$vr$$i(year)$$j(month)$$wm
86332 L $$81.1$$a1-4$$i1941-1944
86332 L $$81.2$$a6-86$$i1945-1987
```

then the textual holdings statement fields are created as follows:

```
86631 L $$av.1-4(1941-1944)
86631 L $$av.6-86(1945-1987)
```

If the holdings record includes an 866/7/8 field with the same sequence number in subfield $8 as the sequence number registered in subfield $8 of the 863/4/5 field, then no other 866/7/8 field is generated by the program.

If the holdings record includes an 866/7/8 field with sequence number 0 registered in subfield $8, then no other 866/7/8 field will be generated by the program.

Note that this expand program must be defined in the tab_expand table of the holdings library (XXX60).

In order to update the 866/7/8 fields of the HOL record when item arrival is registered, it is recommended to set the following line in tab_z105 of the ADM library. For example: ./usm50/tab/tab_z105

```
  1            2   3     4     5     6     7     8     9    10     11
12
!!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!!!!!-!!!!!
UPDATE-ENUM     n USM60
```

expand_doc_hol_86x_iso

This expand routine is the same as expand_doc_hol_86x_, except that expand_doc_hol_86x_iso creates the textual holding statement so that enumeration and chronology data are displayed adjacent to each other, with enumeration recorded first. This form of display is called Adjacent. The corresponding chronology is enclosed in parentheses. The following is an example of adjacent display:

```
vol.10:no.2 (2003:Feb) –vol.10:no.12(2003:Dec.),
```

If the holdings record contains the following fields:

```
85320 L $$81$$av.$$bno.$$u9$$vr$$i(year)$$j(month)$$wm
86332 L $$81.1$$a1-4$$i1941-1944
86332 L $$81.2$$a6-86$$i1945-1987
```

then the textual holdings statement fields are created as follows:

```
86631 L $$av.1(1941)-v.4(1944)
86631 L $$av.6(1945)-v.86(1987)
```

The textual holdings statement fields are created according to the ANSI/NISO Z39.71-2006 standard (Holdings Statements for Bibliographic Items) with stripped punctuation.

Note that this expand program must be defined in the `tab_expand` table of the holdings library (XXX60).

In order to update the 866/7/8 fields of the HOL record when item arrival is registered, it is recommended to set the following line in tab_z105 of the ADM library. For example: ./usm50/tab/tab_z105.

```
!  1              2    3      4      5      6      7      8      9     10
11    12

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

UPDATE-ENUM     o USM60
```

expand_doc_hol_bib

The `expand_doc_hol_bib` program adds bibliographic data to the holdings record.

The following tables define how the holdings record is updated/merged:

`tab_expand` in the tab directory of the holding library (XXX60):

The parameters column (Col. 3) specifies the routine name to be applied

Column 3 syntax is as follows:
MERGE-TYPE=< routine name listed in the first column in tab_merge>

The following is an extract from the `tab_expand` table:

```
!   1                   2                        3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!>

WEB-FULL   expand_doc_hol_bib           MERGE-TYPE=WEB-FULL
```

`tab_merge` of the `tab` directory of the holdings library (XXX60):

Column 1 of `tab_merge` should match column 3 of `tab_expand`

Column 2 of `tab_merge` identifies the merge program to be used.

Column 3 contains the merge section to be used from the `tab_merge_overlay` table.

The following is an extract from the `tab_merge` table:

```
!   1                 2                          3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!>
WEB-FULL   merge_doc_overlay              99
```

`tab_merge_overlay` of the `tab` directory of the holdings library (XXX60):

This table is used to define how the holdings record is updated (merged) if it already exists.

The table contains the merge set used in `tab_merge` column 3.

The following is an extract from `tab_merge_overlay` table:

```
!1 2 3           4
!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
99 1 Y #####
99 2 Y 245##
99 2 Y 1####
```

According to the above example, the holdings record will display all the fields cataloged in the holdings records with an expand of the bibliographic record's title and main entry, as follows:

```
FMT   L HO
LDR   L ^^^^^nx^^a22^^^^^1i^4500
008   L 0207042u^^^^8^^^4001uueng0000000
LKR   L $$aHOL$$lUSM01$$b000000120
CAT   L $$aYOHANAN$$b10$$c20020704$$lUSM60$$h1317
8520  L $$bULINC$$cGEN$$hB819$$i.G3 1968
10010 L $$aGabriel, Leo,$$d1902-
2451  L $$aExistenzphilosophie :$$bKierkegaard, Jaspers,
Heidegger, Sartre. Dial
og der Positionen /$$cLeo Gabriel. --
```

The expand parameters in Column 3 of `tab_expand` can contain up to 10 comma-separated fields (five characters each, including hashes, for example, 85###). If Column 3 is blank, then all fields from the HOL record are added to the ADM record. To exclude a specific field, precede the list of fields with a dash - (for example, -85### -86###).

You can expand the record that is merged into the expanded record. For example: in `expand_doc_lib1_lib2`, it is possible to expand the lib2 record and merge it into the lib1 record. You do this by using a special section in the format "LIB1-LIB2" in the `tab_expand` table of LIB2. For example:

In the case of `expand_doc_hol_bib`, if the section "HOL-BIB" is defined in the `tab_expand` table of the BIB library, the BIB record is first expanded by using this section, and only then are its fields added to the expanded HOL record or merged into it (according to the content of Column 3 of `tab_expand` in the HOL library).

If "MERGE-TYPE=" is used in Column 3 of `tab_expand` for `expand_doc_bib_hol`, it is possible that more than one HOL record will be merged into the BIB record. For example, in `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, and so on.

expand_doc_hol_loc_1_a
This program retrieves the item information of all the items attached to the holdings record.

expand_doc_hol_loc_2_a
This program creates an STA field with $aSUPPRESSED-TEMP if all the items attached to the holdings record - retrieved by expand_doc_hol_loc_1_a - are in a temporary location. SUPPRESSED holdings records are not displayed at the head of the items list in the Web OPAC.

Note that both `expand_doc_hol_loc_1_a` and `expand_doc_hol_loc_2_a` are defined in the `tab_expand` table of the holdings library tab directory under the HOL-LOC expand routine.

expand_doc_hol_z30_86x

The `expand_doc_hol_z30_86x` program is used to create holdings fields (V66, V67 and V68) based on the 853-855 (Captions and Pattern) and 853X-855X (Enumeration and Chronology) fields from the holdings record together with the information of the linked item records (from the description field) . One holdings field is created for each item.

V66 is created from 853/853X, V67 is created from 854/854X, and V68 is created from 855/855X.

If the holdings record contains the following fields:

```
85331 L $$av.$$bno.$$u3$$vr$$i(year)$$j(month)$$wt$$ypm01,05,09
853X  L $$a2$$b1$$i2000$$j1$$320000101$$80
```

then the holdings field is created as follows:

```
V6651 L $$av.2:no.1(2000:Jan.)$$80
```

Note that this expand program must be defined in the `tab_expand` table of the holdings library (XXX60).

In addition, you can filter the items by their status. You do this by using the parameters column of the `tab_expand` table. Up to 30 item statuses can be defined (separated by commas). The expand program only creates new fields for the items with the statuses included in the column. If a minus (-) is defined in the first position, then the holdings fields are not created for those items with statuses included in the column.

expand_doc_isbn_13

The expand_doc_isbn_13 expand procedure adds a field with a 13-digit ISBN if the record contains a 10-digit ISBN, and vice versa. Expand_doc_isbn_13 should be added to the "INDEX" section of tab_expand, and the Direct Index (p_manage_05) should be rebuilt in order to apply the expand.

expand_doc_ismn_13

The expand_doc_ismn_13 routine is used for 13-digit ISMN codes (field 024 indicator 2 in USMARC and 541 in MAB) and their existing 10-digit counterparts. It is used for converting 10-digit ISMN code to 13-digit code and vice versa.

The procedure should be set in tab_expand with the ISMN field index code as a parameter in col.3 (the name of the ISMN field index code from tab00.eng). If no parameter is entered, the default index code is ISMN.

The following is an example of setting expand_doc_ismn_13 in tab_expand:

```
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
INDEX      expand_doc_ismn_13             ISMN
```

expand_doc_issn_isbn

The `expand_doc_issn_isbn` program creates virtual 020/022 fields from the 020/022 fields that contain characters within parentheses. The program removes both the parentheses and the characters.

For example, if the bibliographic record contains the following field:

```
020   L $$a0226123693 (pbk. : v. 2 : alk. paper)
```

Then the program creates the following new virtual field:

```
020   L $$a0226123693
```

expand_doc_join

The `expand_doc_join` program is used with the `tab_expand_join` table of the library's `tab` directory. This table creates a virtual field out of two or more MARC fields. The `tab_expand_join` table determines which fields and which of its subfields should be joined, their order and what the resulting field should be called.

For example, if the library wants to create a headings list of authors and titles, it is possible to define that MARC21 fields 100 and 245 are to be expanded into a new tag (for example, TMP01). You may then send the new virtual tag and 7XX fields that have subfield $t to a common author/title list.

If there are multiple occurrences of the fields, joining is done in pairs. For example, if `tab_expand_join` is set to join 595 and 596 as new field 599, and the record contains "595 a1", "595 a2", "596 b1", "596 b2", "596 b3" the system will create "599 a1 b1" and "599 a2 b2". "596 b3" will be ignored, since it does not have a 595 pair.

Note that you can redirect information to a new subfield in cases where the original field does not have subfields (for example, when joining the MARC 21 001 field).

In the following example, the contents of the 001 field are redirected into subfield $x for the creation of a virtual field (SYS01) created from the 001 field and the 245 field:

```
! 1     2     3     4     5     6     7
!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!
SYS01 001         x     245## a     t
```

For example, if the cataloging record contains the following fields:

```
001   L 000003333
24504 L $$aThe Yearbook of Medicine
```

then the new expanded field is created as follows:

```
SYS01 L $$x000000333$$tThe Yearbook of Medicine
```

Entering the TYPE=ALL parameter in Column 3 of the `$data_tab/tab_expand` table creates a virtual field out of two or more MARC fields.

Here is an example of the `tab/tab_expand` table with the definition for this functionality:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC    expand_doc_join                TYPE=ALL
```

This functionality also uses the used `tab_expand_join` table of the library's `tab` directory. The `tab_expand_join` table determines which fields and which of its subfields should be joined, their order and what the resulting field should be called. The difference is that this program builds a new virtual field for every combination of the field specified in the table. For example, if three fields are to be joined and the first occurs three times, the second occurs three times and the third occurs two times in the record, then 18 new virtual fields are built by this program.

Column 4 in **expand_doc_join** (the new subfield code column) works according to the following methods:

Take one or more subfields from the text and substitute them with the corresponding number of new subfields. For example:

Take *abc* and change to *xyz*

Take all of the text (or several subfields) and change them to one new subfield. For example:

Take *abcd* and change to *x*

Take all but a few subfields (- sign) and change them to one subfield.

Take -*cd* and change to *x*

Take a few subfields (or all the text) and change them to fewer new subfields. It changes all of them to the first one. For example:

Take *abcd* and change to *xy*. It takes the *abcd* and changes it to *x*.

Take all but a few defined subfields and change them to a few new subfields, changing the text to the first new subfield. For example:

Take -*abcd* and change to *xy*. It takes all text except for *abcd* and changes it to *x*.

Here is an example of a possible configuration in `tab_expand_join` which adds virtual fields to a bibliographic record, through the `expand_doc_join` procedure.

```
MAY00 260## abc   xyz   245## abcd  t
MOD00 260## abc   s     245## abcd  t
DEZ00 260## -a    d     245## abcd  t
GOS00 260## abc   gd    245## abcd  t
ROD00 260## -b    os    245## abcd  t
```

expand_doc_join_filter

The expand_doc_join_filter program is used with the tab_expand_join_filter table of the library's tab directory.

This table creates a virtual field out of up to four fields. The tab_expand_join_filter table determines which fields and which of its subfields should be joined, their order, and what the resulting field should be called. It includes the ability to set filter values and the grouping value for joining. Only fields that contain the filter value and have same grouping value are joined.

For more details, see the table: tab_expand_join_filter.

expand_doc_join_permute

This expand procedure uses `tab_expand_join` and creates all the possible permutations of a field.

For example, if `tab_expand_join` contains the following setup:

```
! 1     2     3     4     5     6     7
```

```
!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!! ->
AU700 700## abc   abc   245## ab    fg
AU700 700## abc   abc   740## a     f
```

and if a record has the following lines:

```
000000508 LDR   L 00000cas^^2200589^^^4500
000000508 001   L 000000508
000000508 005   L 20040725104218.0
000000508 008   L 750907c19349999pauqr1p^^^^^^^0uu1a0engdd
000000508 1102  L $$aMusic Library Association.
000000508 24500 L $$aNotes.
000000508 26000 L $$a[Philadelphia, PA, etc.] :$$bMusic Library
Association.
000000508 300   L $$av. :$$bill., ports., facsims. ;$$c24-29
cm.
7001  L $$aO'Meara, Eva Judd,$$eed.
7001  L $$aFox, Charles Warren,$$d1904-$$eed.
7001  L $$aHill, Richard S.$$q(Richard Synyer),$$d1901-
1961,$$eed.
7400  L $$5wid$$aNotes (Online)
7400  L $$aNotes (Journal)
```

then the fields will be expanded in the following way:

```
AU700 L $$aO'Meara, Eva Judd,$$fNotes.
AU700 L $$aFox, Charles Warren,$$fNotes.
AU700 L $$aHill, Richard S.$$fNotes.
AU700 L $$aO'Meara, Eva Judd,$$fNotes (Online)
AU700 L $$aO'Meara, Eva Judd,$$fNotes (Journal)
AU700 L $$aFox, Charles Warren,$$fNotes (Online)
AU700 L $$aFox, Charles Warren,$$fNotes (Journal)
AU700 L $$aHill, Richard S.$$fNotes (Online)
AU700 L $$aHill, Richard S.$$fNotes (Journal)
```

The text taken for the virtual field will be changed according to the filing indicator defined in `tab01.eng`.

expand_doc_join_simple

The `expand_doc_join_simple` program is used with the `tab_expand_join_simple` table of the library's `tab` directory. This table creates a virtual field taking specific occurrences or all occurrences of a field and joins it with specific or all occurrences of another field. The `tab_expand_join_simple` table determines which fields and which of its subfields should be joined, their order and what the resulting field should be called.

Note that the `expand_doc_join` program is useful for indexing, while `expand_doc_join_simple` is useful for creating virtual fields for display.

expand_doc_last_cat

The expand_doc_last_cat program adds a field, named LAS, that is identical to the latest CAT field. Using this expand enables, for example, running retrieve jobs that

will fetch only records that have been last updated by a specific cataloger or updated at a specific hour.

expand_doc_link_to_doc

The expand_doc_link_to_doc program adds a field named DRL to a record. This field contains in subfield $a - a direct link to the full view of the record in the Web OPAC.

expand_doc_link_to_ros

For integration between Aleph and Rosetta. In the Aleph Web OPAC, this routine enables the link to the content aggregator in Rosetta for records that are exported to Rosetta. For more information, refer to the Aleph document, *How to Integrate Aleph with Rosetta*.

expand_doc_open_cat

The expand_doc_open_cat program creates a CDATE field from the first CAT field.

expand_doc_own

The expand_doc_own program appends the contents of the OWN field to all fields in the record. The contents are added under subfield $1.

expand_doc_primo_plk

expand_doc_primo_plk provides the linking details from target document to the source. It creates a new field, PLK, which describes a 'DN' link in a document. Using this new field, Primo published documents can be linked the target document to its source (so that the 'UP' and 'DN' linkages are functioning). The fields LKR-TEXT-M and LKR-TEXT-N are created for the PLK field. This field has a limit of 300 characters for each subfield.

expand_doc_ros_id

For integration between Aleph and Rosetta. This expand generates the cms_id tag (Aleph BIB system number). For more information, refer to the Aleph document, *How to Integrate Aleph with Rosetta*.

expand_doc_rotate

The `expand_doc_rotate` program builds a virtual 600 field from subfields $a (personal name) and $t (title of a work) of the MARC21 600 field (subject added entry - personal name). In the new 600 field, subfield $t is sorted before subfield $a.

For example, from 600 $a $c $d $t `expand_doc_rotate` adds 600 $t $a.

expand_doc_section

The `expand_doc_section` program uses the parameters column of the `tab_expand` table to define the section in `tab_expand` (the expand routine - column 1) that should be performed by the program. This program is especially useful when a set of expand programs has to be performed for many instances of the system (for many expand routines). Instead of defining the group of expand programs for each of the routines, the group can be defined once and then all the different instances can point to it by using the `expand_doc_section` program. Following is an example of the `expand_doc_section` program in `tab_expand`:

```
!   1                   2                             3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
```

```
U39-DOC    expand_doc_section             SECTION1
WEB-FULL   expand_doc_section             SECTION1
WEB-BRIEF  expand_doc_section             SECTION1
GUI-BRIEF  expand_doc_section             SECTION1
GUI-DOC-D  expand_doc_section             SECTION1

SECTION1   expand_doc_bib_loc_usm
SECTION1   expand_doc_yr
SECTION1   expand_doc_type                tab_type_config
```

expand_doc_sort

The `expand_doc_sort` program is used with the `tab_expand_sort` table of the library's `tab` directory. The program sorts the subfields of a specific field according to the sorting order setup in the table.

expand_doc_sort_field

This program sorts a specific field according to the parameters defined in column 3 of the library's `tab_expand` table. In the following example, 260 MARC21 field is sorted according to the contents of subfield $b (the name of the publisher, distributor, and so on.):

`tab_expand` is defined as follows:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC    expand_doc_sort_field          260##,b
```

260 fields were cataloged as follows:

```
260   L $$aBogota :$$bOveja Negra,$$c1999.

260   L $$aLondon :$$bHeadline Book Publishing,$$c1994.
```

According to the above definitions of `tab_expand` for `expand_doc_sort_field`, the fields are now sorted as follows:

```
260   L $$aLondon :$$bHeadline Book Publishing,$$c1994.
260   L $$aBogota :$$bOveja Negra,$$c1999.
```

expand_doc_sort_loc_a

This program sorts uniquely the PS1 fields (items + holdings) creating a PST field for each unique PS1. PS1 match for uniqueness is on sublibrary, collection, call number and status.

This program should be used for sites where the items and the holdings records are not linked.

expand_doc_sort_loc_b

This program sorts uniquely the PS1 fields (items) creating a PST field for each unique PS1. PS1 match for uniqueness is on sublibrary, collection, call number, status, and material type. Note that holdings records that do not have linked items already have their PST field created directly by `expand_doc_bib_loc_1_c`.

This program should be used for sites where the items and the holdings records are linked.

Note that as `expand_doc_sort_loc_a` and `expand_doc_sort_loc_b` are mutually exclusive, it is only possible to use one of these at a time.

expand_doc_split

The `expand_doc_split` program, together with the `tab_expand_split` table of the library's `tab` directory, cuts the content of a tag into separate tags on each occurrence of a subfield, taking all the subfields from one subfield to the next. For example, if subfield $a is set in the table, then $a $b $c $a $a $c is split into:

```
$a $b $c
$a
$a $c
```

Since the split occurs on every occurrence of the subfield, the program creates 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. The resulting sections of the tag are redirected into virtual "output" tags.

If the `tab_expand_split` table contains the following line:

```
 1   2      3     4
!!!!!-!-----!!!!!-!!!!!
700## t     A700  T700
```

then the following field:

```
700   L $$aMendelssohn-Bartholdy, Felix$$tLider
ohne Worte,$$mpiano
```

is cut into:

```
A700  L $$aMendelssohn-Bartholdy, Felix
T700  L $$tLider ohne Worte,$$mpiano
```

expand_doc_split_external

This program, together with the `tab_expand_external` table of the library's `tab` directory, is used to split the content of a tag, with multiple occurrences of subfields containing external location (856, 505 and so on), into separate tab fields for each occurrence of the designated subfield.

The program duplicates all subfields besides the designated subfield and places them in each new tag field.

Note that the new lines created by the program have the original tag while the original line is suppressed.

The `tab_expand_external` table is used to define fields to be inspected in order to find multiple occurrences of subfields containing external locations, as follows:

```
! 1   2
!!!!!-!-
856## u
505## u
```

For example, if the cataloging record contains the following field:

```
856   L
$$uhttp://jefferson.village.virginia.edu/pmc/contents.all.html$
$uhttp://
lcweb2.loc.gov/ammem/ead/jackson.sgm
```

then the new expanded fields are created as follows:

```
856   L
$$uhttp://jefferson.village.virginia.edu/pmc/contents.all.html
856   L $$uhttp://lcweb2.loc.gov/ammem/ead/jackson.sgm
```

expand_doc_split_sub1

This program is similar to the `expand_doc_split` program. Both programs, together with the `tab_expand_split` table of the library's tab directory, cut the content of a tag into separate tags on each occurrence of a subfield. The difference is that in the `expand_doc_split_sub1` program the resulting paired fields are assigned subfield $1 with a sequence number that shows the original pairing.

For example, if the bibliographic record contains the following fields:

```
7001  L $$aShakespeare, William,$$d1564-1616.$$tPlays.
7001  L $$aDonne, John.$$tSonnets
```

and the tab_expand_split table is defined as follows:

```
! 1   2      3     4
!!!!!-!-----!!!!!-!!!!!
700## t     A700  T700
```

then the expand_doc_split_sub1 program creates the following new fields:

```
A700  L $$aShakespeare, William,$$d1564-1616.$$1001
A700  L $$aDonne, John.$$1002
T700  L $$tPlays.$$1001
T700  L $$tSonnets$$1002
```

This expand routine is required for the correct building of the brief records (Z0101). The building of the brief records relies on subfield $1 in order to pair the split fields. If this program is not used, the brief records for the above example are created as follows:

```
$$aShakespeare, William,$$d1564-1616.$$tPlays.
$$aDonne, John.$$tPlays.
```

When using the `expand_doc_split_sub`1 program, the program that builds the brief records prefers the T700 that has the same subfield $1 as the original A700 that initiated the building of the brief record. The brief records are created as follows:

```
$$aShakespeare, William,$$d1564-1616.$$tPlays.
$$aDonne, John.$$tSonnets.
```

expand_doc_sysno

The `expand_doc_sysno` program is primarily intended for the Z39_SERVER instance in the `tab_expand` table of the library's tab directory. This program creates a virtual SYS field records the Z39.50 server. This program works with the tab04 table of the library's tab directory. The tab04 table is used to set up the specifications for the conversion of the "SYS" tag to a numeric tag . Note that it should always appear under conversion routine 90. The following is an extract from the `tab04` table:

```
90 SYS##            903## N
90 #####            ##### N
```

Note that the last line in this extract must always be present. In addition, note that the program must be defined under the Z39_SERVER entry before the `expand_doc_bib_tab04` program as follows:

```
!   1                   2
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
Z39_SERVER expand_doc_sysno
Z39_SERVER expand_doc_bib_tab04
```

expand_doc_type

This program can be used to create a new field according to the specifications defined in a configuration table which is itself a parameter that must be defined in the parameters column of the `tab_expand` table (see Configuration Tables (expand_doc_type). 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 field(s) present in the record (for example, a combination of the LDR and the 008 field). In the following example, the new field contains 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 that the name of the configuration table (for example, `tab_type_config`) should be added as a parameter in column 3 of the library's `tab_expand` table.

`expand_doc_type` can be used with repetitive fields. This is especially useful for the EXIST function. For example:

```
!!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!!!!!!-
!!!!!!!!!!-!!!!!!!!!!!!
TYP   EL Electronic material            856## u          EXIST
```

Note the following:

If the second parameter in `tab_expand` is left blank or set to "Y" as follows:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD       expand_doc_type                tab_type_config.eng,Y
```

then the program matches the first tag which matches the tag+subfield condition. Accordingly, the EQUAL, N-EQUAL or MATCH functions cannot be used for repetitive fields with the same subfield. For example:

```
TYP      Publisher                    260## b          EQUAL      Oclc
```

The line above does not work with a record such as:

```
000000000 L 26001 $$aNew York$$bRandom House
000000000 L 26001 $$aNew York$$bOCLC
```

since it only takes the first line.

If the second parameter in `tab_expand` is set to "N" as follows:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD       expand_doc_type                tab_type_config.eng,N
```

then the program will attempt to find matches with every occurrence of the repetitive field and subfields in the record.

For example:

`tab_type_config` includes the following lines:

```
! 1   2           3                    4       5        6            7
```

```
!!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!!!!!!-!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!
LOC      LOC-1                          852## b          EQUAL      WID
LOC      LOC-2                          852## b          EQUAL      HIL
LOC      LOC_3                          852## b          EQUAL      MED
LOC      LOC-4                          852## b          EQUAL      LAW
```

For a record that includes the following fields:

```
852   L $$bLAW
852   L $$bMED
852   L $$bHIL
852   L $$bWID
```

The result of the expand will be:

```
LOC   L $$aLOC-1
LOC   L $$aLOC-2
LOC   L $$aLOC_3
LOC   L $$aLOC-4
```

It is also possible to create TYP fields for multiple subfields occurrences. In the example below the first character is “S” (for subfield), the second character is the subfield itself (b), and the third character is the subfield occurrence (1-9,A). To enable the expand to check the match for multiple subfield occurrences, set the third character to "A" (all). For example:

```
! 1  2        3              4     5       6         7
!!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
```

```
YR  $a [year]
```

In the following example, the `tab_expand` table is set up so that the year is taken from the 008 MARC 21 field. If the 008 field has no publication date, the contents expanded into the new YR field are taken from subfield $c of the 260 MARC 21 field:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC    expand_doc_yr                   008,260##c
```

If the publication year is 2001, for example, then the new field is built as follows:

```
YR    L $$a2001
```

This program can also take the year from the 100 UNIMARC field:

```
!   1                   2                              3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC    expand_doc_yr                   100
```

Note:
In addition to the above list of expand routines, it is also possible to use external expand routines. The external expand routines can be written in any programming language and can be executed without being linked to Aleph. These expand programs are defined in tab_expand and can be used for extended bibliographic information. The external program must reside in $aleph_exe and should not have an extension.

9.3 Expand-Related Tables

9.3.1 Configuration tables (expand_doc_type)

The **tab_type_config** table in the library's `tab` directory is a sample of a table that can be used with the `expand_doc_type` program. The table defines specifications for the creation of a new field. Additional similar tables can be added, after which they can be listed as parameters in the `tab_expand` (in col.3) table.

In the following example, the new virtual field contains 'FILM' according to 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 a sample from a configuration table:

```
! 1   2            3            4        5         6         7
8
!!!!!-!!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!!!!!!-!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!
982      Media               LDR   F06-01     EQUAL      [g,k,r,o]
```

```
982   BK Book                 LDR   F06-01     EQUAL      a

982      Film                 LDR   F06-01     EQUAL      g
                              008   F33-01     EQUAL      m

982      Videorecording       LDR   F06-01     EQUAL      g
                              008   F08-01     EQUAL      v
```

Note that the tables used by the `expand_doc_type` program are language-sensitive. If, for example, the `tab_expand` table is set up as follows:

```
!   1                   2                            3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD       expand_doc_type                tab_type_config
```

then the system looks for the matching `tab_type_config` in the following order based on the language:

$data_tab/tab_type_config.<lng>

$data_tab/tab_type_config.<control_lng> (as defined in aleph _start)

$data_tab/tab_type_config

Key to the configuration tables (for example, tab_type_config):

Column 1 - Target tag
Target field created with the contents of column 2 and/or column 3 by the expand_doc_type.

Column 2 - Format code
Format code, for example, BK (for book). The value entered in this column is expanded into subfield $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 subfield $a of the new field. For example, if the table contains the following line:

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

then a new TYP field with the following structure is 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
```

Column 3 - Format name
Format name, for example, Book. If a format code is present (column 2), then the format name is added/expanded into subfield $b of the new field. If no format code is defined, then the format name is added/expanded into subfield $a of the new field. For example, if the table contains the following line:

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

then a new TYP field with the following structure is 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 is created/expanded as follows:

```
TYP   L $$aBook
```

Column 4 - Field tag
Field from the record used to determine 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
```

Column 5 - Subfield(s) or fixed field position
This column contains the subfield codes or the fixed field positions (of the field defined in column 4) to be checked. In the following line, the program checks position 06 of the LDR:

```
TYP   BK Book                 LDR   F06-01     EQUAL      a
```

In the following line, the program checks subfield $a of the 490 field:

```
TYP    Thesis          4901# a     MATCH     [masters*,education*]
```

You can specify positional conditions in this column by the following syntax:

S<subfield><occurrence>FNN-NN

For example

```
!!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!!!!!!-!!!!!!!!!!-
!!!!!!!!!!!!
TYP   AB Book                          100   Sa1F06-01  EQUAL      x
```

This example shows that if in field 100, the first subfield is *a*, which occurs once, and position 6 length 1 contains *x*, then an expanded TYP field will contain *AB book*.

Column 6 - Match criteria
This column is used to define the match criteria in relation to the contents of the subfield or the fixed field positions defined in column 5. Following are the available

options: EQUAL, N-EQUAL, EXIST, N-EXIST, MATCH and N-MATCH.

Usage:

EQUAL (N-EQUAL) - checks for direct match

EXIST (N-EXIST) - checks if the field exists without checking the field contents. For example, if a record has a 027 MARC 21 field, then the record is a technical report (contents are irrelevant).

MATCH (N-MATCH) - checks for a match that is not case-sensitive

Column 7 - Contents
This column contains the contents of the field or of the fixed field position used for matching (according to the match criteria defined in column 6). Use [] to enclose multiple values for matching (up to 15 different comparison values). 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]
```

Column 8 - Case-sensitive matching
Case-sensitive matching flag. Values are:

Y = Matching is case-sensitive

N = Matching is not case-sensitive

Note that if this column is left blank, the default is matching which is not case-sensitive.

9.3.2 tab_expand_split

The **tab_expand_split** table is used to set up the specifications for the `expand_doc_split` program. This program splits a field into separate parts, splitting on the subfield specified in column 2 of the `tab_expand_split` table. The split occurs 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, if subfield $a is set in the table, then $a $b $c $a $a $c split into:

```
$a $b $c
$a
$a $c
```

The following is a sample of the `tab_expand_split` table:

```
! 1   2      3     4
!!!!!-!-----!!!!!-!!!!!
260## b     PLA   PUB
```

```
700## t     100   240
```

Key to the tab_expand_split table:

Column 1 - Tag and Indicators
Contains the tag and indicators of the document field to be treated by the expand program. # can be used for the third to fifth positions to indicate truncation of numeric additions to the field code (for example, 245## is used for 2451, 2452, 24501, and so on).

Column 2 - Subfield
Contains the subfield to be split on.

Column 3 - "Up-to" Tag
Output field tag and indicators for text up to the breaking subfield.

Column 4 - "After" Tag
Output field tag and indicators for text after the breaking subfield.

Based on the sample table, the following field:

```
260   $$aBoston: $$bLittle, Brown, and company,$$c1933.
```

is cut into:

```
PLA   $$aBoston:
PUB   $$bLittle, Brown, and company,$$c1933.
```

9.3.3 tab_abbrev

The `tab_abbrev` table used is in conjunction with the `expand_doc_fix_abbreviation` program.

The `expand_doc_fix_abbreviation` program is used to add a duplicate virtual field to the document where the shortened form (abbreviation) of a word or a phrase is changed to the whole word or phrase. In actual fact, the routine can be used to replace any text string in a bibliographic record with a different text string. The `tab_abbrev` table contains the list of "abbreviations" and the whole forms into which the shortened form is to be changed in the new virtual field added to the record by the `expand_doc_fix_abbreviation` program. The text exchange is defined per field tag (including indicators).

Note that this program can be also used as a fix program (using `tab_fix` in place of `tab_expand`). The difference is that instead of adding virtual fields (for indexing or/and display), when used as a fix program, the fields with the whole forms are actually added to the record.

Following is a sample of the tab_abbrev table:

```
! 1   2         3                              4
!!!!!-!-!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
225## Y 1st                   FIRST
043## Y u-at---               Australia
043## Y u-at-ne               Australia New South Wales
```

Key to the tab_abbrev table:

Column 1 - Field Tag and Indicators
Contains the tag and indicators of the document field to be treated by the expand program. Field tags can be grouped by using # as a wildcard (for example, 2#### can be used to include all field tags that begin with "2").

Column 2 - Usage Code
Select the appropriate usage code. The following values are available:

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 columns 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 #. For example, the following line in the table changes Ft. to FORT in all fields that begin with 2:

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

If the table contains the following:

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

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

Y: Defines that the text exchange should occur.

Column 3 - Abbreviation
Contains the text that should be changed in the new field added to the record.

Column 4 - Expanded Form
Contains the expanded form for the abbreviation in column 3.

Based on the sample table, the following document:

```
043   L $$au-at-ne
1001  L $$aWilliams, David
2451  L $$aThe 1st man
```

is expanded into:

```
043   L $$au-at-ne
1001  L $$aWilliams, David
2451  L $$aThe 1st man
043   L $$aAustralia New South Wales
2451  L $$aThe FIRST man
```

9.3.4 tab_expand_duplicate_field

The `tab_expand_duplicate_field` table is used to set up the specifications for the expand_doc_duplicate_field program. This program is used to duplicate a field, assigning a new field tag plus indicators.

The following is a sample of the `tab_expand_duplicate_field`

```
! 1   2
!!!!!-!!!!!
260&335;# IMP
```

The expand_doc_duplicate_field procedure and expand_doc_duplicate table can be used in order to overcome the problem created when using expand_doc_split, which does not retain the source field. When expand_doc_split is based on a field created by the expand_doc_duplicate_field program, the source field is retained. In the above example, the IMP field can be later processed by expand_doc_split without losing the original 260 field.

Key to the tab_expand_split table:

Column 1 - Input Field Tag and Indicators
Contains the tag and indicators of the document field to be duplicated by the expand program.

Column 2 - Output Field Tag and Indicators
Contains tag and indicators after duplication.

Based on the sample table, the following field:

```
2600 L $$aBoston, :$$bRoberts Bros
```

Is duplicated into:

```
IMP   L $$aBoston, :$$bRoberts Bros
```

9.3.5 tab_expand_external

The `tab_expand_external` table is used together with the `expand_doc_split_external` program.

The `expand_doc_split_external` program is used to split the contents of a tag with multiple occurrences of subfields containing an external location (856, 505 and so on) into separate tab fields for each occurrence of the designated subfield.

The `tab_expand_external` table is used to define fields to be inspected in order to find multiple occurrences of subfields containing external locations, as follows:

```
! 1   2
!!!!!-!-
856## u
505## u
```

Key to the tab_expand_external table:

Column 1 - Input Field Tag and Indicators
Contains the tag and indicators of the document field to be inspected by the expand program.

Column 2 - Output Field Tag and Indicators
Contains subfield tag occurring multiple times in the field.

Based on the sample table, in the case of the following field:

```
5054  L $$:a505 FORMATTED CONTENTS
NOTE$$:uhttp://lcweb.loc.gov/marc/bibliograph
ic/ecbdnot2.html#mrcb555$$:uhttp://www.loc.gov/standards/mets
```

the new expanded fields are created as follows:

```
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
```

9.3.6 expand_doc_bib_z30

The **expand_doc_bib_z30** table is used to define the information from the item record that is expanded by the expand_doc_bib_z30 program. The table defines the following: which fields are taken from the item record; in which cases these fields should be taken (for issues, for copy items, and so on,); in which subfields of the expanded field the information should be stored; and for some specific item fields, whether the codes should be replaced by names (for example, the sublibrary code can be replaced by the sublibrary name).

9.3.7 expand_doc_bib_z403

The expand_doc_bib_z403 table is used to define the information from the object's data information that is expanded by the expand_doc_bib_z403 program. The table defines the following: which fields are taken from the object's data record (Z403); in which subfields of the expanded field the information should be stored; and for some specific fields of the object's data record, whether the codes should be replaced by names (for example, the sublibrary code can be replaced by the sublibrary name).

Note that column two of this table usually contains the field from the Z403 record (for example, z403-sub-library), in order to create a valid 856 URL field, the column should contain url and the expanded field should be set in tab_expand to be 856. This option should only be used in special cases such as the X and Z39.50 servers.

9.4 Indexing Expand Fields (Virtual Fields)

In order to index, follow these steps:

Set up the appropriate expand table. For example:

tab_expand_extract

```
! 1   2   3
!!!!!-!-!!!!!
650## a SUBJ
```

Add the expand field to tab11.

Call the expand program in the ACC/ WORD/ INDEX entry of tab_expand:

For keyword indexes, the expand program must be added to the WORD entry in tab expand.

For headings, the expand program must be added to the ACC entry.

For direct indexes, the expand program must be added to the INDEX entry, and for sort indexes, it must be added to the entry SORT-DOC.

For display, it must be added to WEB-FULL, WEB-FULL-1, WEB-BRIEF, GUI-DOC, GUI-BRIEF.

Example:

tab_expand

```
WORD        expand_doc_fmt_mgu
WORD        expand_doc_join
!
ACC         expand_doc_bib_loc_usm
ACC         expand_doc_bib_ndu
ACC         expand_doc_join
!
INDEX       expand_doc_extract
INDEX       expand_doc_bib_loc_usm
SORT-DOC    expand_doc_bib_loc_usm
!
WEB-BRIEF   expand_doc_join_simple
```

!

9.4.1 tab_expand_extract

This table defines the extraction of subfields for indexing. For example, if you want to create a headings list of chronological subdivisions for subject added entries - topical terms, you can define that MARC21 650 field, subfield $y is to be expanded into a new tag (for example, y650). The virtual field may then be indexed or displayed.

Note that the fourth column in the tab_expand_extract table can be used to specify the number of subfield occurrences for which the new virtual field is created. For example, you can define that only the first occurrence of subfield $y in the 650 field should be used for the creation of the new field. The following is an example of extraction from the table:

```
! 1   2   3   4
!!!!!-!-!!!!!-!
650#:#y CHRON  1
```

Key to the tab_expand_extract table

Column 1 - Input field tag and Indicators
Tag and indicators of the document field containing the subfield to be extracted by the program.

Column 2
Contains the subfield to be extracted by the program and expanded into a new field tag.

Column 3 - Output Field Tag and Indicators
Contains the new field tag for indexing

Based on the sample table, the following subfield y:

```
650-0 L $$y19th century$$aProtestants$$zFrance.
```

Is extracted into the virtual field tag CHRON as follows:

```
CHRON L $$y19th century
```

9.4.2 tab_expand_join

This table creates a virtual field out of two or more MARC fields. The `tab_expand_join` table determines which fields and which of its subfields should be joined, their order and what the resulting field should be called.

For example, if you want to create a headings list of authors and titles, you can define that MARC21 fields 100 and 245 are to be expanded into a new tag (for example,

AUTIT). You can then send the new virtual tag to an author/title list. The virtual field is used for indexing purposes.

Following is a sample from the `tab_expand_join` table:

```
! 1     2     3     4     5     6
!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!
AUTIT 100## a           245## abcd
```

Key to the tab_expand_join table

Column 1 - New virtual field tag and indicators

Column 2 - First tag for building the new virtual tag

Column 3 - Subfields to take for the match
If this column is empty, all subfields are taken. If you want a selection of subfields, enter the subfield codes.

Column 4 - New subfield code

Column 5 - Second tag for building the new virtual tag

Column 6 - Subfields to take for match
If this column is empty, all subfields are taken. If you want a selection of subfields, enter the subfield codes.

Note

If there are multiple occurrences of the fields, joining is done in pairs. For example, if tab_expand_join is set to join 595 and 596 as the i new field 599, and if the record contains "595 a1", "595 a2", "596 b1", "596 b2", "596 b3", the system will create "599 a1 b1" and "599 a2 b2". "596 b3" will be ignored, since it does not have a 595 pair.

Based on the sample table, the following fields:

```
10010 L $$aOrcibal, Jean.24510 L $$aLouis XIV et les protestants,
:$$b"La cabale des accommodeurs de reli
```

Are concatenated into the virtual field tag AUTIT as follows:

```
AUTIT L $$aOrcibal, Jean.$$aLouis XIV et les protestants,
$$b"La cabale des accommodeurs de religion", la caisse des
conversions, la revocation de l'dit de Nantes.
```

9.4.3 tab_expand_join_simple

This table creates a virtual field taking specific occurrences or all occurrences of a field and concatenates it with specific or all occurrences of another field. The `tab_expand_join_simple` table determines which fields and which of its subfields

should be concatenated, their order and what the resulting field should be called. The virtual field is used for display purposes.

Following is a sample from the `tab_expand_join_simple` table:

```
! 1     2   3    4     5     6    7   8     9
!!!!!-!!!!!-!!-!!!!!-!!!!!-!!!!!-!!-!!!!!-!!!!!
ATS02 100## AA a     a     245## AA       y
```

Key to the tab_expand_join_simple table

Column 1 - New virtual field tag and indicators

Column 2 - First tag for building the new virtual tag

Column 3 - First tag occurrence; nn for index, AA for All

Column 4 - Subfield to take for match

Column 5 - New subfield code

Column 6 - Second tag for building the new virtual tag

Column 7 - Second tag occurrence; nn for index, AA for All

Column 8 - Subfield for match

Column 9 - New subfield code

Based on the sample table, the following fields:

```
10010 L $$aOrcibal, Jean.
24510 L $$aLouis XIV et les protestants, :$$b"La cabale des
accommodeurs de religion", la caisse des conversions, la revocation
de l'dit de Nantes.
```

Are concatenated into the virtual field tag ATS02 as follows:

```
ATS02 L $$aOrcibal, Jean.$$aLouis XIV et les protestants,
:$$b"La cabale des accommodeurs de religion", la caisse des
conversions, la revocation de l'dit de Nantes.
```

10 Other Indexes

This chapter includes the following sections:

Update Short Bibliographic Records (manage-07)

Update Sort Index (manage-27)

Update Indexes for Selected Records (manage-40)

10.1 Update Short Bibliographic Records (manage-07)

A short bibliographic record is an abbreviated version of the bibliographic record in standard Oracle table format. It contains up to six fixed (system-defined) fields (year, call number/call number key, author, title, imprint and ISSN/ISBN) limited to 100 characters each (except for the call number key field that can be up to 80 characters long). Additionally, it can contain up to fifteen user-defined fields, each limited to 500 characters.

The purpose of a short bibliographic record is to provide bibliographic information in an effective and timely manner, particularly for instances where bibliographic information is an adjunct to administrative information.

A short bibliographic record is built by the system according to the definitions in the `tab22` table.

When to Update Short Bibliographic Records

Run this service after making any change in the tab22 table that affects the short bibliographic records.

Short bibliographic records must be updated if a large number of records have been uploaded into the database in the "partial" mode, since the system does not build short bibliographic records automatically for these documents.

10.2 Update Sort Index (manage-27)

This service updates the Sort Index of the database.

The Z101 table contains sorting keys. When a list of brief records is displayed in the Web OPAC, the Z101 records are used to arrange the list in a specific order. The fields that are used for building sorting keys are defined in the library's `tab/tab_sort` table in the library's tab directory.

10.3 Update Indexes for Selected Records (manage-40)

This service writes the requested document numbers in the Z07 Oracle table, after which the records are re-indexed through the library's usual updating process (ue_01).

When to Run This Service

Run this service when you need to re-index selected document records.

11 Preparation for Index Jobs

p_manage_01, p_manage_02, p_manage_07 (for z00r), and p_manage_32 can require a large amount of disk space when they are run on large databases if this is the first time they are being run in the specific library and may take a long time to run.

11.1 Clean temp/scratch Directories

Make sure that the `xxx01 $data_scratch`, the `xxx01 $data_files`, and the `$TMPDIR` directories are cleaned of any extraneous, temporary files. If, based on calculations in the preceding disk space section, the disks on which the `$data_scratch`, `$data_files`, or `$TMPDIR` reside may not have enough space, consider moving them temporarily to a different location.

11.2 Check Oracle Space

You can use UTIL A/17/11 (Check Space Utilization of Oracle Tables) to make sure that the relevant files (Z97/Z98 for Words, Z01/Z02 for Headings, and Z11 for Direct index) are nowhere near their maximum number of extents (505, as delivered by Ex Libris) or their maximum tablespace sizes. If they are, the extent sizes specified in the `xxx01/file_list` or the tablespace(s) sizes should be increased prior to running the job.

11.3 Cancel Jobs Which Might Interfere

manage_nn index jobs may need to run when backup jobs or other jobs normally run. You need to make sure that jobs which would interfere with the index job are cancelled. The mechanisms for automatic running are the ALEPH job daemon and the Unix cron. The ALEPH job daemon can be cancelled via UTIL E/15 or individual jobs can be commented out via UTIL E/16. Consult your Unix systems administrator in regard to cron jobs which might interfere.

12 Parallel Indexing

Parallel indexing is used to rebuild an OPAC index parallel to the online ALEPH system without downtime while the index is being created. This process also lets you change indexing parameters and check the results without losing current indexes.

The indexing is carried out in a separate library. This library is set up with a pointer to the document records in the actual library, and indexes are located in the indexing library. After indexing has been completed, a pointer is created in the actual library to the index in the indexing library.

In the examples in the following sections, USM01 is used as the actual library, and USM21 is used as the indexing library.

To start the indexing process in the USM21 library, assume that all index tables (for example, for Word tables this would be Z97, Z98, and so on) are defined as local in the files list, and the documents file as a logical synonym to USM01. You can now run the indexing job (for example, p_manage_01) in USM21. It reads records from USM01 via the logical synonym, but creates index tables locally. You can also

change the index setup in the indexing library to create different index codes or to use different filing procedures.

After the index is built successfully, check it in USM21 using Web OPAC.

Finally, if you want to switch to the new index, create a logical synonym from USM01 to USM21 for all relevant tables. Thus there is no downtime whatsoever.

The re-indexing of word (W-nnn) and direct (IND) indexes is an autonomous process that does not require any other indexing. Re-indexing the headings (ACC) index, on the other hand, requires a series of indexing jobs.

The following are the steps that should be taken to activate parallel indexing:

Step 1: Open a Library
Open a new BIB library, parallel to the library that is going to be indexed.

Step 2: Add the Indexing Library to library_relation
The library_relation table in the /alephe/tab directory defines relationships among various libraries. For parallel indexing, a PID relationship must be defined, as in the following example:

PID USM21 USM01

In addition, the relationship between the indexing library and the ADM and HOL libraries must be defined in exactly the same way as the relationship between the actual library and its related ADM and HOL libraries. For example:

ADM USM01 USM50 USM51
HOL USM01 USM60

ADM USM21 USM50 USM51
HOL USM21 USM60

PID USM21 USM01

Step 3: Adjust the Library's file_list
In the root directory of each ALEPH library, there is a configuration table called file_list. This table lists all of the library's Oracle tables, their size, extents and location. In this configuration table, you can define that the library uses an Oracle table from a different library, instead of its own Oracle table. You do this by setting a pointer to the other library, using a Logical Symbol (LS) definition.

Initially, in the root directory of the new library, there should be a copy of the actual library's file_list. At this stage, the file_list should contain all the Oracle tables listed in the actual library's file_list, using the same definitions. Later on, the definitions are changed, in both the actual and the indexing library, as required.

In the indexing library, the sequence numbers table, Z52, must always be local:

```
TAB z52          10K          10K          ts0
IND z52_id       10K          10K          ts1
```

In the indexing library, you must always use a logical synonym for the bibliographic documents table, Z00 as well as Z103 table:

```
LS  z00          usm01
LS  z103         usm01
```

In addition, the TAB and IND lines for the z00 and z103 need to be commented out (by placing "!" in front of them).
Before initiating parallel indexing, the above tables should be dropped in the indexing library (the parallel library).
Before doing the SQL drops for this (shown further below), do the following select:

```
 >>s+ usm21
SQL-USM21> select count(*) from  Z00 ;
SQL-USM21> select count(*) from  Z103 ;
```

The result to both should be 0. If it is not, it indicates that you are either:
(1) not in the parallel library or
(2) LS-ed from the parallel library to the production.
Quit immediately.
In the case of #1, do "s+" to the correct, parallel library.
In the case of #2, do util a/17/5/1 to confirm that the synonyms already exist. If so, you do not need to do anything more. The fact that the synonyms exist indicates that the tables have been dropped. (Oracle does not let you create the synonym when the table exists in the parallel library.)
If the result of the "select count" is 0, then continue with the SQL commands for doing the drop:

```
SQL-USM21>drop table Z00;
SQL-USM21>drop table Z103;
```

After changing the file_list and dropping the tables, create the logical synonyms (UTIL A/17/5/2) in the parallel library.
Note: If you are building Words Index and you use synonyms (Z970), all the said above for z00 and z103 is true also for z970.
Do not forget to check if there is enough free space to build the index in the parallel library tablespace (util o/14/1). You can see the size of the index tables in the actual (USM01) library with util a/17/11/2. (Note that the latter result is in KB whereas the first is in MB). The new tables require an equal amount of space in the parallel library tablespaces (-- unless you change the number of indexes or fields indexed).

Step 4: Adjust the Library's Z52 Table

Before initiating parallel indexing, make sure that the following counters are defined as listed. Use UTIL/G/2 to add missing counters, and update the counter values.

The required values are:
last-doc-number (set to the same value as last-doc-number of the actual library)
last acc-number (set to 0)
last-long-acc-number (set to 0)
last-similar-acc-num (set to 0)

last-word-number (set to 0)

Step 5: Set up Indexing Configuration Tables

Define the setup of the indexing tables in the tab directory of the indexing library. You can choose to copy the tables that are used for indexing in the actual library into the tab directory of the indexing library, or you can use the path_convert configuration table to direct the system to the actual library's configuration tables. This option is feasible only if you do not want to change the indexing setup.
You may change $data_tab values in the indexing library in order to improve the indexing. If you do this, be sure to copy these changed tables to the actual library in Step 9.

Step 6: Save Interim Indexing Updates

While the indexing jobs are running, new and updated records in the actual library are indexed, through the ue_01 and Z07 mechanism, on the old indexes. These Z07 records must be saved for re-indexing after the new indexes have been built. In order to save these records, before running the indexing process, activate the E/5/1 utility in the actual library. UTIL E/5 lets you create a history table for Z07 entries handled by ue_01. These stored entries may be later used to re-execute ue_01 on the same records.

Step 7: Run the Indexing Jobs

Word Indexes

Run p_manage_01 to rebuild the Word Index of the database.
Direct Indexes
Run p_manage_05 to rebuild the Direct Index of the database.
Headings Indexes
Optional: Run p_manage_102; to pre-enrich the bibliographic headings index, based on the Authority Database.
Run p_manage_02.
Start UE-08 to build the bibliographic heading - Authority Record connection.

Note: If you have run p_manage_102, starting UE-08 at this stage is not necessary.
Run p_manage_105 in the AUT libraries to add untraced references.
Run p_manage_17 to alphabetize long headings.
Run p_manage_35 to create brief records (z0101) .
Run p_manage_32 to build counters for logical bases.
Note: Since v17, p_manage_32 can be run in the parallel library.

Note: When rebuilding headings (browse) indexes, you must run the additional indexing processes listed above.
If your AUT database does not include untraced headings, there is no need to run p_manage_105.

If you do not have logical bases, or if you have not set "Y" in column 8 of tab_base.lng for any of the bases, then there is no need to run p_manage_32.

If you are not using brief records, you do not have to run p_manage_35.
Other Indexing Jobs to Run
Run p_ manage-07 to update short bibliographic records.

Run p_manage_27 to update the sort index.

Step 8: Check the New Index

Add the indexing library to /alephe/tab/tab_base.lng and add the library to the base list for the Web OPAC (/alephe/www_f_lg/base_list). Access the WEB OPAC, choose the indexing library, and check the new index. If all appears satisfactory, continue with Step 9.

Note: The location does not display in the OPAC Brief or Full displays when performing this test. (This is because there is no ADM or HOL for the parallel library.)

If this run of the parallel indexing is just for the Keywords, then the Browse and the Browse links in the Full display do not work in performing this test. If this run of the parallel indexing is just for the Browse, then the Keyword does not work in performing this test.

If your site is using Union Catalog or Union View, note that they do not work in performing this test unless you add LS's for the z120 and z127.

Step 9: Applying the New Indexes

When the indexing has been completed, stop all running daemons in the actual library (UTIL E).

Option 1: Logical Synonyms

This option uses logical synonyms to point from the actual library to the indexing library Oracle tables in order to apply the new indexes.

The next step required is switching from the current (old) index to the new index. Create a pointer from the actual library Oracle table to the Oracle table in the indexing library, by changing the definition in the actual library's file_list to a logical synonym.

The following example shows the new setup after re-indexing headings:

Replace:

```
TAB z01          2M          1M          ts0
IND z01_id        1M          1M          ts1
IND z01_id2       300K        100K        ts1
IND z01_id3       200K        100K        ts1
IND z01_id4       200K        100K        ts1
IND z01_id5       200K        100K        ts1
IND z01_id6       200K        100K        ts1
```

With:

```
LS  z01          USM21
```

Replace:

```
TAB z02          400K        100K        ts0
IND z02_id        300K        100K        ts1
```

IND z02_id1 400K 100K ts1

With:

LS z02 USM21

The following example shows the new setup after re-indexing words:

Replace:

TAB z95 1M 1M ts0
IND z95_id 1M 1M ts1
TAB z97 2M 1M ts0
IND z97_id 1M 1M ts1
IND z97_id1 1M 1M ts1
IND z97_id2 1M 1M ts1
IND z97_id3 1M 1M ts1
TAB z98 3M 1M ts0
IND z98_id 2M 1M ts1
TAB z980 1M 1M ts0
IND z980_id 1M 1M ts1

With:

LS z95 USM21
LS z97 USM21
LS z98 USM21
LS z980 USM21

Before doing the SQL drops for this (shown further below), do the following select:
>>s+ usm21
SQL-USM21> select count(*) from Z01 ;
SQL-USM21> select count(*) from Z02 ;
Check that these tables are not empty.
Drop the relevant Z tables (mentioned above) in the actual library by using the SQL command, as in the following example:

>>s+ usm01
SQL-USM01>drop table Z01;
SQL-USM01>drop table Z02;
etc.
Then create logical synonyms to the indexing library, using UTIL A/17/5 in the actual library.

Note: If you use this option, the next time you want to re-index, you must create an additional indexing library (in order that both the BIB documents library and the indexing library remain unlocked).

Option2: Oracle Import

Installations that have Oracle DBA expertise can choose to copy the new indexes (that is, the Oracle tables) from the indexing library to the actual library.

Whether Option 1 or Option 2 is chosen, the following actions must be done:
When the new index is a result of the p_manage_02 process, update the last-acc-number and the last-similar-acc-number counters in the actual library (using UTIL G/2) to the same value as the counter in the indexing library.
If the p_manage_02 service has been run in "Update headings index" procedure and Duplicate Mode: Yes; update the last-long-acc-number counter in the actual library (using UTIL G/2) to the same value as the counter in the indexing library.
When the new index is a result of the p_manage_01 process, update the last-word-number counter in the actual library (using UTIL G/2) with the same value as the counter in the indexing library.
When the new index is a result of the p_manage_35 process, update the last-z0101-sequence counter in the actual library (using UTIL G/2) with the same value as the counter in the indexing library.
When the new index is a result of the p_manage_17 process, update the last-long-acc-number counter in the actual library (using UTIL G/2) with the same value as the counter in the indexing library.
If you have changed $data_tab tables in the indexing library in order to improve the indexing (see Step 5, above), copy these changed tables to the actual library. In case some of these files have an earlier timestamp than the ones they are replacing, do util x/7 to clean out the utf_files.
The last action in this section of the re-indexing process is restarting the daemons (UTIL E…) in the actual library.

Step 10: Index Records that Have Been Updated in the Interim

In order to include records that have been updated while indexing has been running in the indexing library, activate UTIL/E/5/2 in the actual library. This copies the saved Z07H records to Z07, deleting duplicate entries. The ongoing ue_01 process in the actual library re-indexes the records stored in Z07.

13 Indexing Services

Each service is identified in the Batch Log and Batch Queue by its procedure name.

Rebuild Word Index (manage-01) Target

This service updates the Word Index of the database. It locks the ALEPH system and should only be run when the library is closed.

Update Direct Index (manage-05)

This service updates the Direct Index of the database. It locks the ALEPH system and should only be run when the library is closed.

Update Headings Index (manage-02)

This service updates the Headings Index of the database. It locks the library and should only be run when the library is closed.

Note that when you run this service in "Rebuild Entire Headings Index" mode, you must run the "Alphabetize Long Headings" (manage-17) service in order to have correct alphabetization.

Update Sort Index (manage-27)

This service updates the Sort Index of the database. It locks the library and should only be run when the library is closed.

Alphabetize Headings - Setup (manage-16)

This setup service alphabetizes the headings according to the rules for alphabetization that are kept in the `tab00.lng` table and the `tab_filing` table.

These rules create a "filing text" by which the heading is alphabetized. The headings are then alphabetized according to the first 69 characters of the filing text of each entry.

The Alphabetize Long Headings (manage-17) service alphabetizes those headings whose filing texts are longer than 69 characters.

After you run this service, always run the Alphabetize Long Headings (manage-17) service. This service locks the library and should only be run when the library is closed.

Alphabetize Long Headings (manage-17)

This service alphabetizes those headings whose filing texts are longer than 69 characters. This service locks the library and should only be run when the library is closed.

Update Short Bibliographic Records (manage-07)

This service updates the Short Bibliographic Records of the database. It locks the library and should only be run when the library is closed.

Note that if the CREATE-Z00R variable of the tab100 table in the library's tab directory is set to Y, this function also updates/creates Z00R records. The Z00R table contains separate Z00R records for each of the fields in all documents of the database. This information can be used for statistical purposes.

Update Indexes for Selected Records (manage-40)

This service writes the requested document numbers in the Z07 Oracle table, after which the records are re-indexed through the library's usual updating process (ue_01).

Build Counters for Logical Bases (manage-32)

This service builds the counters for logical bases. It should be run after building a Headings index (manage-02) if your database setup is configured for using this counter.

Update Brief Records (manage-35)

This service updates and creates brief records. It locks the ALEPH system and should only be run when the library is closed.

Create Links Between Records (manage-12)

This service creates links between records of the database. It locks the library and should only be run when the library is closed.

In order to re-build all links:

1. Empty z103 (=util a/17/1) from BIB and HOL libraries (and CRS if there is one).
2. Run p-manage-12 in ADM with parameters: delete all=Y, check old=Y.
3. Run p-manage-12 in BIB with parameters: delete all=N, check old=Y.
4. Run p-manage-12 in HOL with parameters: delete all=N, check old=Y.

14 Further Reading

The following ALEPH documents contain material not included in the Indexing chapter. They are available from the Ex Libris Documentation Center. Note that differences might appear between these documents and the current version.

How To Run Index Jobs - This document describes how to run index jobs such as Words and Headings. It touches upon such issues as turning off archive logging, number of processes, disk space and file locations, unlocking the library while a job is running, monitoring jobs, troubleshooting, and estimation runtime.

Index Building / Parallel Processing - This document explains how parallel processing can be used in ALEPH to decrease the time of indexing, within specific constraints. It describes key concepts such as cycles, job duration, disk space calculation and acquaints the reader with the batch jobs involved. It also points out when part of a specific task cannot be done in parallel.

System Librarian's Guide – Integrated ILL

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: April 4, 2017

Web address: http://www.exlibrisgroup.com

Table of Contents

1 OVERVIEW 8

List of important terms 9

2 SETTING UP INTEGRATED ILL 12

2.1 Setting up the ILL Library 12
2.1.1 Loading the ILL GUI Column Headings 15
2.1.2 Create the Z700 Oracle Table 16

2.2 Setting up ILL library administrators 16

2.3 Setting up ILL Units 19

2.4 Creating an authorized staff user 21

2.5 Setting up Updating Privileges for the WWW-X and Z39 system users 22

2.6 Setting up external partners 22

2.7 Creating patron records for ILL partners used as ILL requestors 24

2.8 Creating a roster of potential suppliers (for borrowing requests) 25

2.9 Assigning ILL units to library patrons 28

2.10 Create Customer ID Records 29

3 ILL *LOCATE* FUNCTION 29

3.1 Supplier Availability 30

3.2 Title Availability 30
3.2.1 Searching for Titles 30
3.2.2 Tiered Title *Locate* 32
3.2.3 Item Searching 33

3.3 '*Locate*' Possible Results 34

4 BACKGROUND PROCESSES 35

4.1 XXX40 ILL Library 35

4.2 ILLSV Library 35

4.3 General 35

5 SETTING UP UNMEDIATED MODE FOR ILL UNITS 35

5.1 Setting Up the 'Unmediated Mode' 36

5.2 Unmediated Activities - Borrowing 36
5.2.1 Defining how borrowing requests are handled - tab50 37

5.3 Unmediated Activities - Lending 38

6 SETTING UP "LOCALLY-OWNED" CHECKS 39

6.1 Setting up the 'Locally-owned' policy 39

6.2 Setting the system's response to locally-owned results 40

7 SETTING UP PATRON CHECKS WHEN A NEW BORROWING REQUEST IS SUBMITTED 40

8 SETTING UP NON-WORKING DAYS OF POTENTIAL SUPPLIERS 41

9 CHANGE REQUEST STATUS 41

10 CONFIGURING ILL CASH TRANSACTIONS (TAB18.ENG) 42

11 BORROWING - CHARGING PATRONS FOR ARRIVAL OF ILL MATERIAL 43

12 LENDING -CHARGING ILL REQUESTER FOR SUPPLYING ILL MATERIAL 45

13 PATRON'S COPYRIGHT AGREEMENT FOR BORROWING REQUESTS 46

13.1 Determining copyright policy (tab47.eng) 46

13.2 Producing a Copyright Agreement Letter 47

14 THE BORROWING REQUEST'S FORMAT AND MEDIA 48

14.1 Building the 592 expand field (ill_format_type and expand_doc_type program) 48

14.2 Defining media per format (tab_ill_preferred_media) 50

15 LOADING A DEFAULT ROSTER OF SUPPLIERS 51

15.1 tab_z701 51

15.2 Loading the default roster of potential suppliers 53

16 HANDLING BLDSC BORROWING REQUESTS 53

16.1 Setting up the BLDSC as an External Partner and a Potential Supplier 53

16.2 Setting up Customer ID and Keywords 54

16.3 Assigning the BLDSC to a New Request 54

16.4 Sending the new request to the BLDSC 55

16.5 The BLDSC Template Format 56

16.6 Resending requests to BLDSC 59

17 THE BLDSC SERVICES: ADD AND SED SERVICES 59

17.1 Activating the ADD Service 60

17.2 Activating the SED Service 61

18 HANDLING SLNP BORROWING AND LENDING REQUESTS 61

18.1 Setting up the Central ILL Server as an External Partner 62

18.2 Setting up the Partner Libraries to Work with SLNP 62

18.3 Setting Up Items as Available for ILL 62

18.4 Configuring your local SLNP server and starting it 62

19 APDU EXCEPTIONS 62

20 ILL TEMPORARY ITEMS 65

20.1 Circulation Checks 66

20.2 The ILL Item Values 67

20.3 Setting up handling of ILL Items (tab27) 69

20.4 Multi Items Related Configuration 70

21 LENDING LIBRARY - SUPPLYING AN ITEM 71

22 ILL REQUEST LOG 72

22.1 tab46.eng Structure 72

22.2 Request Log Display – Sort Routines 72

23 REQUEST NUMBERS 73

23.1 Borrowing - Request Number 73

23.2 Borrowing – Supplier Request Number 73

23.3 Lending – Request Number 73

23.4 Lending – Requester's Request Number 74

24 ILL CONFIGURATION TABLES 74

24.1 ILL Library Tables (./xxx40/tab) 74

24.2 ADM Library Tables (./xxx50/tab) 78

24.3 The alephe tables (./alephe/tab) 78

24.4 Aleph_start - ILL Related Entries 80

24.5 The ILLSV tables (./illsv/tab) 80

25 TAB100 - RELATED ENTRIES FOR ILL 82

26 PC_TAB_EXP_FIELD.ENG 86

27 LENDING REQUEST - SETTING UP CAPTIONS FOR VOLUME AND ISSUE 88

28 PRINT TEMPLATES 89

28.1 GUI ILL Templates 89

28.2 GUI ILL Services Templates 92

29 DEFINING ILL COLUMN HEADINGS (PC_TAB_COL.LNG) 93

30 PC_SERVER_DEFAULTS - ILL RELATED ENTRIES 94

31 WWW_SERVER.CONF - ILL-RELATED ENTRIES 96

32 PC SETUP - ILL.INI FILE 96

32.1 [General] 96

32.2 [BorrowStatusTree] 97

32.3 [LendStatusTree] 97

32.4 [AdminTree] 98

32.5 [GetDefaultValue] 98

32.6 AddSupplierMethod] 99

33 COPYRIGHT – AUSTRALIAN MODE 99

34 COPYRIGHT – US MODE 100

34.1 Prerequisites 100
34.1.1 ILL US Copyright Mode 100
34.1.2 Setting ILL Units Groups for Copyrights Procedures 100
34.1.3 ILL Unit/Group Copyright Policy 101

34.2 US Copyright Workflow 102
34.2.1 General 102
34.2.2 Initial Copyright Check 103
34.2.3 Additional Copyright Checks 103
34.2.4 Copyright Management GUI ILL Interface 104
34.2.5 Request Log for Copyright Activity 105

34.3 Processing the Copyright Payment Report 106
34.3.1 The Copyright Payment Report (ill-61) 106
34.3.2 Manual Control of Payment Status 109

34.4 Copyright - Statistics Report 110

35 MAPPING FIELDS IN ILL DOCUMENTS 112

1 Overview

The Integrated ILL module is used to manage

- Your library patrons' ILL procurement requests
- Requests directed to your library by other libraries to supply material through ILL
- The administrative records that are required to support the above two services

The ILL records are contained within an ALEPH® library of the ILL type (for example, 'USM40', 'XXX40'). Each ILL library is related to a single ADM library (for example, 'USM50', 'XXX50'), and vice-versa.

The Integrated ILL records and procedures involve two-way traffic:

- Requests by patrons to procure material
- Requests directed to the library to supply material

There are a number of ways in which you can customize your Integrated ILL system regarding:

- The method of communication with ILL partners – mail, e-mail, ISO or SLNP protocols
- The degree to which the system does unmediated handling of both lending and borrowing requests

There are two types of records that must be in place before you and your staff users can begin to work with Integrated ILL: Partners and Roster of potential Suppliers.

- Partners – External and Internal
    - External Partners – the external organizations to which you direct your requests, and from whom you receive requests
    - Internal Partners (ILL Units) – the organizations (administrative units within the ADM libraries in your ALEPH installation) that handle ILL requests (both borrowing (patron requests) and lending (requests from other libraries)). The ILL Unit functions in a manner similar to a sublibrary, and several sublibraries can be grouped into a single ILL Unit. A specific ILL Unit, set in the Global Patron Record, handles the patron's requests.
- Roster of potential suppliers that is used to assign the suppliers to which a borrowing (outgoing) request will be sent. For SLNP, this is not necessary as the potential suppliers are identified by the ZFL server.

There are two types of requests (two request forms), in both the GUI and in the Web OPAC:

- Monograph
- Serial

List of important terms

APDU

The APDU (Application Protocol Data Unit) is the communication unit between a reader and a card. The structure of an APDU is defined by the ISO 7816 standards.

There are two categories of APDUs: command APDUs and response APDUs. As the name implies, the former is sent by the reader to the card: it contains a mandatory 5-byte header and from 0 to up to 255 bytes of data. The latter is sent by the card to the reader: it contains a mandatory 2-byte status word and from 0 to up to 256 bytes of data.

Borrowing

Borrowing refers to the activity of requesting material from another library; although the word "borrow" implies a circulation checkout transaction, the library might send you a copy, which need not be returned (non-returnable).

Borrowing request

A borrowing request is also known as an outgoing request. A Borrowing request is sent by an ILL unit to a potential supplier, in order to fulfill a patron's request for material.

The following list of actions outlines the steps for a borrowing request. Steps marked with an asterisk (*) can be set to unmediated; steps marked with two asterisks (**) can be unmediated only if communication is through ISO ILL.

1. Create request (created by patron in the OPAC, or by staff in Integrated ILL module)
2. * Check whether requested material is locally owned (when a request is placed by the patron in the OPAC, the request can be blocked if it is locally owned)
3. * Assign potential suppliers to the request
4. * Send the request to the potential supplier
5. ** Register supplier response (if communication is through ISO, this will be unmediated)
6. Register receipt of the material
7. If material received is a returnable item, return the material

Bulk handling

The Integrated ILL GUI includes "bulk" processing for material dispatch and receipt, based on reporting a barcode and/or request number.Communication

You can communicate with ILL Partners in different ways:

- Print (Mail/fax)
- E-mail

- ISO ILL 10160/10161 protocol
- SLNP (Simple Library Network Protocol, used in Germany)

Incoming request

An incoming request is received from an ILL partner, requesting that you supply material. An incoming request is also called a lending request.

Lending

Lending refers to the activity of supplying material to another library; although the word "lend" implies a circulation checkout transaction, you can send material as a non-returnable copy.

Lending request

A lending request is also known as an incoming request.

The following list of actions outlines the steps for a lending request. Steps marked with an asterisk (*) can be set to unmediated; steps marked with two asterisks (**) can be unmediated only if communication is through ISO ILL.

8. ** Create request (if communication is through ISO, this can only be unmediated)
9. * Search to verify if the library has the requested material
10. ** Send response to requester if the material is not available in your library
11. Send response other than "unfilled" (not available)
12. Register shipment of the material
13. If material shipped is a returnable item, return the material

Locally-owned

A "locally owned" check can be made when a borrowing request is initially created by staff user via GUI ILL or by the patron via the Web OPAC. This can be set as part of the unmediated processing. It can be set so that patrons cannot place requests for locally owned material.

Mediation

ILL requests (borrowing and lending) can be processed as mediated or unmediated as desired, within the constraints relating to how the ILL unit communicates with an ILL partner. All processing, apart from the physical actions of fetching and sending material, and receiving and returning material, can be unmediated. For partners with whom ISO communication is the standard, message handling can also be unmediated, whereas for non-ISO ILL partners, message handling must be manual.

Partners

An ILL partner is an administrative unit that sends requests to you, and/or that responds to your requests for material. Each library/institution that is used as a player (requester or supplier) in the ILL flow, including your own administrative unit, has a partner record in the system.

There are three types of partners: Internal ILL unit, external shared and external non-shared.

- **Internal Partner – also known as ILL Unit: local administrative unit that** handles the ILL borrowing and lending activities. A unit can be set up to represent a single sublibrary, or a group of sublibraries. The unit fulfills two **types of services:**
    - ILL-UNIT for outgoing requests (borrowing library) - handles the ILL requests for a group of patrons; each patron is handled by a default unit defined in that patron's global information record.
    - ILL-UNIT for incoming requests (lending library) - an administrative unit that handles the ILL requests directed to it; it is authorized to lend items from specific sublibraries.
- **Shared External Partner-** a supplier or a requester library which is not a part of any ADM library of your library system; it is external to your **library system (for example, another library or a supplier like BLDSC). An** external shared partner can be used as a supplier by all ADM libraries.
- **Non-shared External Partner** - a non-shared external partner is the same as a shared external partner, but it can be used as a supplier only by the library which created the partner record.

Each ILL partner (both external and internal) to whom you supply ILL material must have a parallel Patron record.

Patrons

Two types of patrons are part of the Integrated ILL setup:

- Library patrons who are served by ILL. They are allowed to submit ILL requests on their own. The Global Patron record includes the assignment of the ILL Unit that will handle his/her requests, a limit of the number of total requests and a limit of the number of active requests allowed for the patron. All ILL-related circulation activities are registered on the patron record: loans; renewals; returns; and so on.
- ILL partners to whom you supply material (external partners and internal ILL units)

Roster

The Potential Suppliers List sets the roster of ILL partners to which an outgoing (borrowing) request is directed. The potential suppliers list is built for each requesting Unit and material type. The list is consulted by the system when a borrowing request is processed.

Staff users

ILL staff users are charged with setting up administrative records (suppliers, units, rosters, patrons), processing outgoing ILL requests and handling incoming requests received from other libraries. Each staff user is allowed to access and perform actions in a single ILL Unit. ILL users must be empowered for the Integrated ILL module, and for required Circulation processes.

2 Setting up Integrated ILL

Note

In order to carry out the steps described in this chapter, you must have a working ALEPH installation.

This chapter describes the tasks you need to perform in order to set up the Integrated ILL GUI. These steps must be carried out in the following order before you and your colleagues can start working with Integrated ILL:

2.1 Setting up the ILL Library on page 12.

2.2 Setting up ILL library administrators on page 16.

2.3 Setting up ILL Units on page 19.

2.4 Creating an authorized staff user on page 21.

2.6 Setting up external partners on page 22.

2.7 Creating patron records for ILL partners used as ILL requestors on page 24.

2.8 Creating a roster of potential suppliers (for borrowing requests) on page 25.

2.9 Assigning ILL units to library patrons on page 28.

2.1 Setting up the ILL Library

To set up the ILL library:

1. From the command prompt, define the XXX40 libraries in aleph_start.private by adding the ILL library codes in ./alephe/ aleph_start.private as in the following example:

   ```
   setenv    usm40_dev          ${USER_MOUNT}/u18_tst
   setenv    ALEPH_LIBS         "$ALEPH_LIBS usm01 usm10 usm30 usm50 usm51 usm40 usm41"
   ```

2. Re-login to the server so that the above changes take affect.

Note Check license status in util y/11 to make sure you haven't exceeded the number of licenses defined in $alephe_tab/license.

3. From the command prompt, type the following:

```
>ap
>open_new_library
```

4. When asked by the script regarding the size of the new library, use the following guidelines to determine the size:
   VSMALL – up to 20,000 BIB records
   SMALL – up to 100,000 BIB records
   MEDIUM – 100,000 – 1,000,000 BIB records
   LARGE – 1,000,000 – 5,000,000 BIB records
   HUGE – more than 5,000,000 BIB records
5. When asked by the script which library you want to use as a template, use usm40 as a template if this is the first ILL library you have created. For any additional library, you can use the previously-created ILL library as a template.
6. Type N when asked "Do you want all work files (scratch/ & files/) under x_path (give full x_path)/[N]?"
7. Type N when asked "Enter list (of one or more), separated by comma N/Y/pw,usr,z105:"
8. Type Y when asked "Do you want to create Oracle environment (user, table, index, synonym, trigger, sequence); if it exists, all data will be erased!".

The ILL Library (XXX40) must be defined in ./alephe/tab/library_relation.

The relevant relation types are: ILL, PAS, and BIB.

ILL

Defines the relation between the ILL library (XXX40) and the ADM library (XXX50). Note that the ILL library will point to the ADM library (and not in the opposite direction as in other relation types).

For example:

```
ILL USM40 USM50
ILL USM41 USM51
```

ADM relation:

Defines the relation between the ADM library (XXX50) and the ILL library (XXX40) to be able to navigate from an ILL record to its linked item. For example:

```
ADM USM40 USM50
ADM USM41 USM51
```

PAS

Defines the ILL library that will be displayed in the list of libraries that appears in the Access Rights window when creating privileges for an operator. For example:

```
PAS USM50 USM01 USM40 USM10 USM60 USM30 USR00
PAS USM51 USM01 USM41 USM10 USM60 USR00
```

BIB relation**:**

Defines the relation between the BIB library (XXX01) and the ILL library (XXX40). For example:

```
BIB USM50 USM01 USM30 USM40
```

1. Set tab_base.eng – Add the new ILL library code to ./alephe/tab/tab_base.eng. Three entries must be set per ILL library. For example, for ILL library XXX40, set the following three lines: Lines per entry: XXX40, XXX40-B and XXX40-L.

```
XXX40          USMARC ILL (XXX40)      XXX40 N
XXX40-B        USMARC ILL Borr-40      XXX40 N wit=BORROW
XXX40-L        USMARC ILL Lend-40      XXX40 N wit=LEND
```

2. Set up the GUI Services menu – The GUI services menu file of the integrated ILL < menu-ill.xml > is located under ./aleph/pc_b_eng. If you have a local pc_b_eng directory under ./alephe/pc_b_eng, copy menu-ill.xml file from aleph/tab to alephe/tab. If you do not have a local pc_b_eng directory in. /alephe, disregard this section.
3. If you are using ISO ILL, define the following setting in $alephe_root/aleph_start:

```
setenv    ISO_ILL              Y
```

This setting automatically starts the ILL server upon Aleph startup.

Ue_17 runs in the ILLSV library if you are using ISO ILL. Ue_19 runs in the XXX40 library if you are automatically processing the documents such as with the automatic locate functionality. Perform the following procedure to have either of these processes start automatically:

1. For UE_17, make sure the following section exists in $alephe_root/aleph_startup:

```
****************************************#
       #UE_17
****************************************#
        echo "starting ue_17"...
        start_proc00 ILLSV
        source $aleph_proc/util_e_01
        source $aleph_proc/util_e_17
```

```
        echo" "
```

2. For UE_19, configure switch ROSTER_STARTUP_LIBS for xxx40 in aleph_start.private
3. Copy the following files from the tree alephe/tab (sometimes these files are found in alephe.orig/tab of the a tree) to the local $alephe_tab:

   tab_ill_change_status_in

   tab_ill_change_status_out

   tab_ill_held_media_map.lng

   tab_ill_partner_conf

   tab_ill_status_borrow

   tab_ill_status_lend

   tab_ill_supp_close_days

   tab_ill_tree_bor_msg

   tab_ill_tree_lend_msg

2.1.1 Loading the ILL GUI Column Headings

pc_tab_col.lng defines the columns that are displayed in list windows in the GUI clients.

In order to define the column headings, edit the bibliographic library (XXX01) table `pc_tab_col.lng`. For more information about `pc_tab_col.lng`, see the *General Using and Customizing Your GUI* chapter of the *Aleph 20.0 Staff User's Guide - General Using and Customizing Your GUI.*

The following is a list of the ILL windows which use pc_tab_col.lng for formatting data and their identifiers (Column 1 in pc_tab_col.lng):

PC_ILL_PARTNER_LIST: Administration – Partners List
PC_PARTNERS_LIST: Borrowing – Request Expand –Suppliers tab
PC_ILL_B_LIST: Borrowing – List of Requests
PC_ILL_L_LIST: Lending – List of Requests
PC_ROSTER_LIST: Administration – Potential Suppliers List
PC_ILL_*LOCATE*_PREV: Borrowing – *Locate* action – Preview
PC_ILL_*LOCATE*_PREV_L: Lending – *Locate* action – Preview
PC_ILL_MLC: Lending – *Locate* action – Preview – Multi *Locate* preview
PC_ILL_LOG: Borrowing and Lending – Request Log
PC_MSG_LIST: Borrowing and Lending – Request expand – 5th tab: Messages
PC_ILL_APDU_LIST: Administration – APDU Exceptions Lost
PC_ILL_MSG_LIST: Borrowing and Lending – Summary Mode –Messages pane
PC_ILL_STATUS_GROUP: Borrowing and Lending – Summary mode: main pane
PC_ILL_ITEM_LIST: Lending – Request expand – Items tab
PC_ILL_CUSTOMER_ID: Administration – Customer ID

2.1.2 Create the Z700 Oracle Table

The Z700 Oracle table must be created in the pw_library.

1. Check that the merged file_list in your password library contains z700 using util a/17/10/2:

   ```
   >echo $pw_library
   ```

   ```
   >dlib
   ```
   [the result you received from the previous command]

   ```
   >util a/17/10/2
   ```

2. If z700 is not defined in util a/17/10/2, add the following lines to the file_list table found in $data_root of the password library:

   ```
   TAB z700              128K            0K              ts1d
   IND z700_id           128K            0K              ts1x
   IND z700_id1          128K            0K              ts1x
   IND z700_id2          128K            0K              ts1x
   ```

   Then run the following steps:

   ```
   >dlib xxx50
   ```

   >util a/17/1

   Type z700, answer `yes` to create the table.

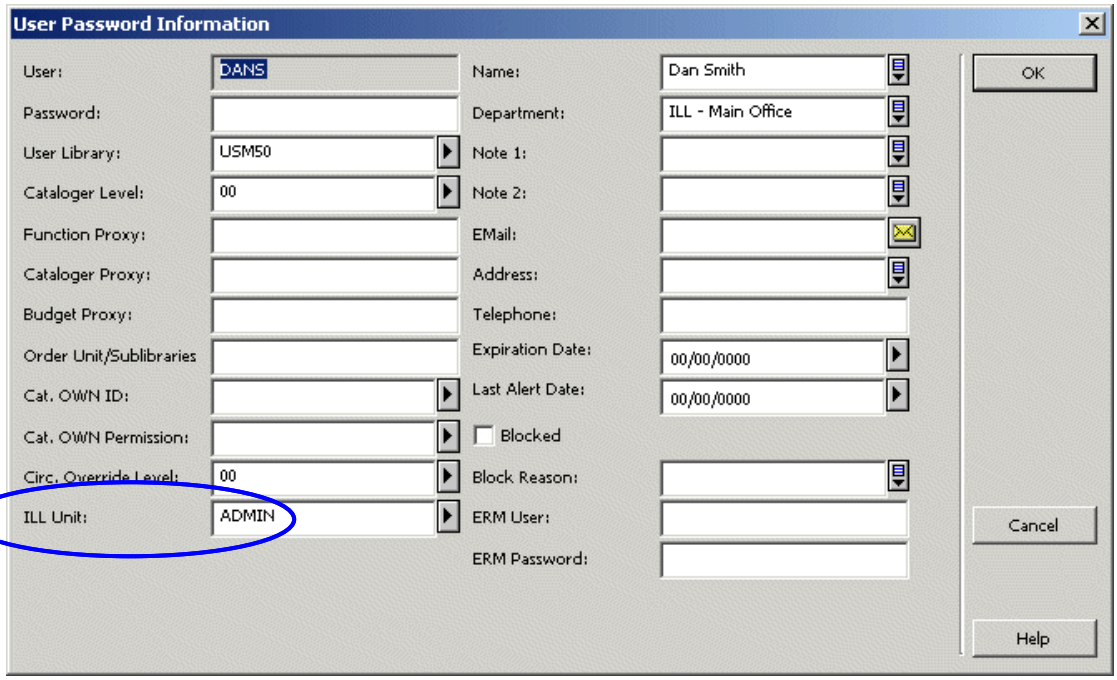
2.2 Setting up ILL library administrators

You need to set up ILL library administrators in order to continue with setting up Integrated ILL.

The ILL library administrator (ADMIN) is in charge of setting up ILL partners: external partners and internal partners (ILL units) for his ILL library. The ILL library administrator can add, update, and delete ILL units (see section 2.3 Setting up ILL Units on page 19).

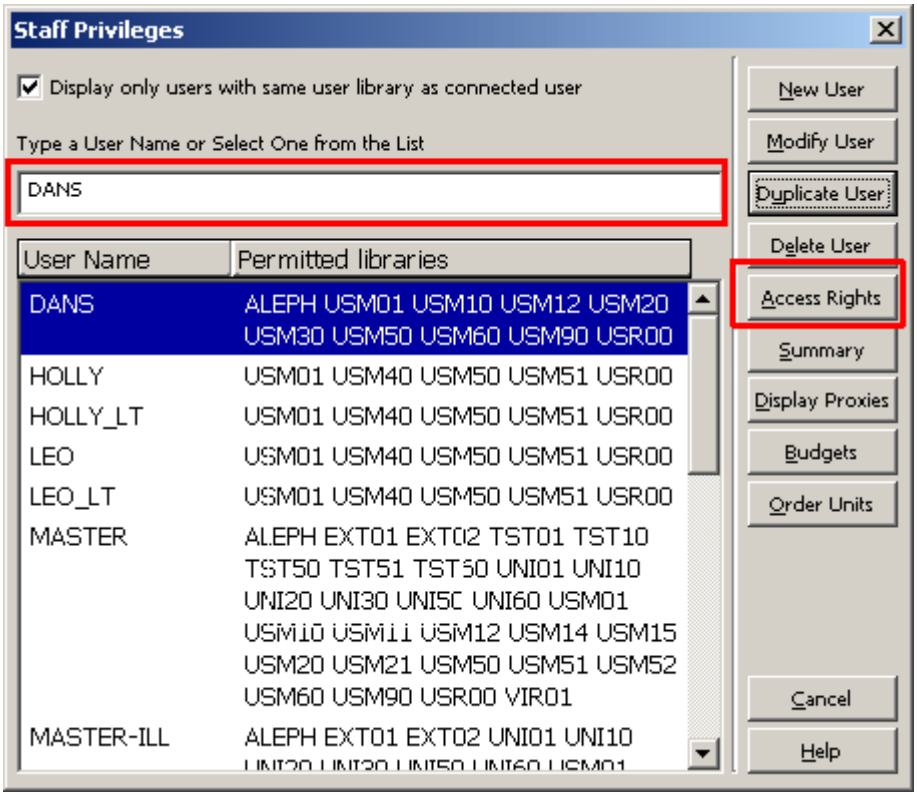
To set up an ILL library administrator

3. Log in to either Circulation or Acquisitions/Serials.
4. From the status bar, right-click .
5. Click **Staff Privileges**.
6. In the Staff Privileges window, select the appropriate user and click **New User**. The User Password Information window opens.
7. Create a staff user record with the value ADMIN in the ILL Unit field. The following screen capture shows an example of setting up an ADMIN ILL user for the USM50 library:



In our example, USM40 is set as the ILL library of USM50; therefore, this ADMIN user can now set ILL Units for USM40. When you have finished filling in the User Password Information Form, click OK to return to the Staff Privileges window.

8. Select your ADMIN user from the list and click **Access Rights**:

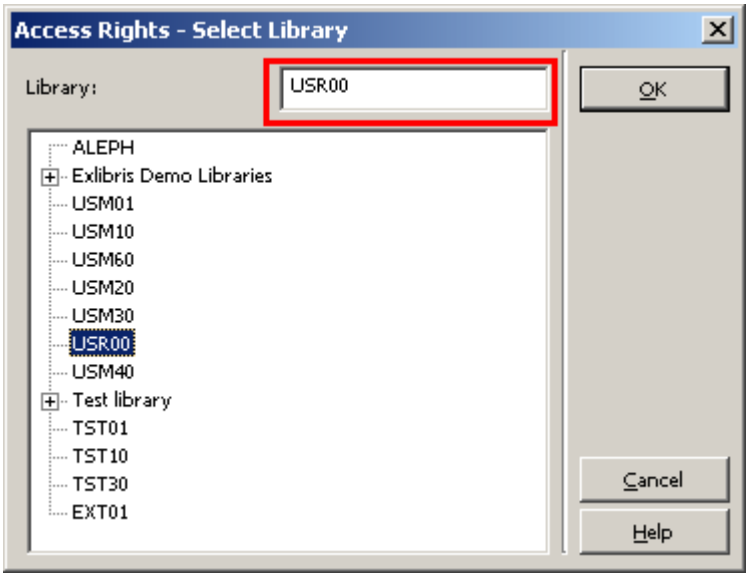


The Select Library window appears.

At the very least, assign to the ADMIN user ILL partner privileges (Integrated ILL – Partner Management) in the password library (pw_library, for example:

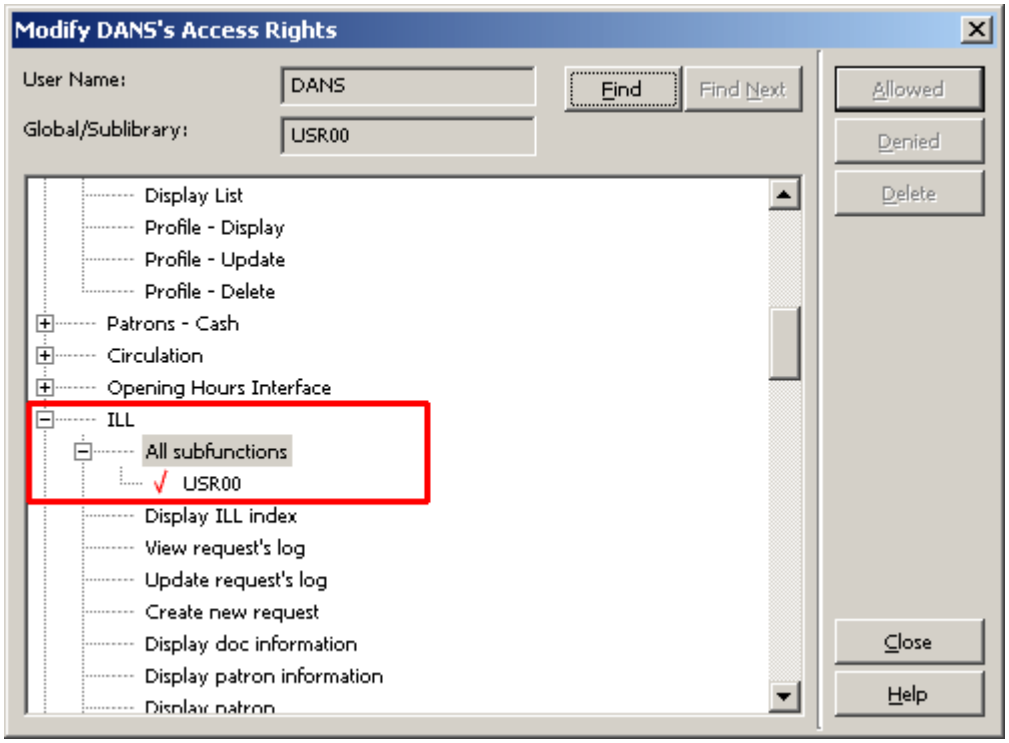
USR00). The ADMIN user does not need any privileges in the ILL library (for example: USM40). Note: if the ADMIN user is also expected to open patron records for the ILL partners that require them, then the ADMIN user will require related patron privileges in the in the password library and ADM library.

9. Select the password library, for example, USR00 and click **OK**:



The Modify <name of user>'s Access Rights window is displayed.

10. In the tree of library functions, scroll down to ILL. Click the + sign and choose the relevant subfunctions (we suggest All subfunctions), then click **Allowed**:



Repeat steps 8-10 to assign access rights to other libraries, for example, to assign patron-related privileges in the ADM library.

Note that the ADMIN user is authorized only for the setting up of ILL partners: Internal partners (also known as ILL units) and external partners. For all other ILL activities, such as building up a roster of potential suppliers, processing borrowing requests, processing lending requests, use the ILL units' authorized users.

The next step in setting up Integrated ILL is setting up ILL Units.

2.3 Setting up ILL Units

Before your library can begin to provide ILL services, ILL units must be entered in the system. Each ILL partner who requests an item from you, or who responds to your requests for material, must have a partner record in your system. In addition, you yourself are an ILL unit. ILL units must be present in the system in order for an ALEPH user to be created (see section 2.4 Creating an authorized staff user on page 21).

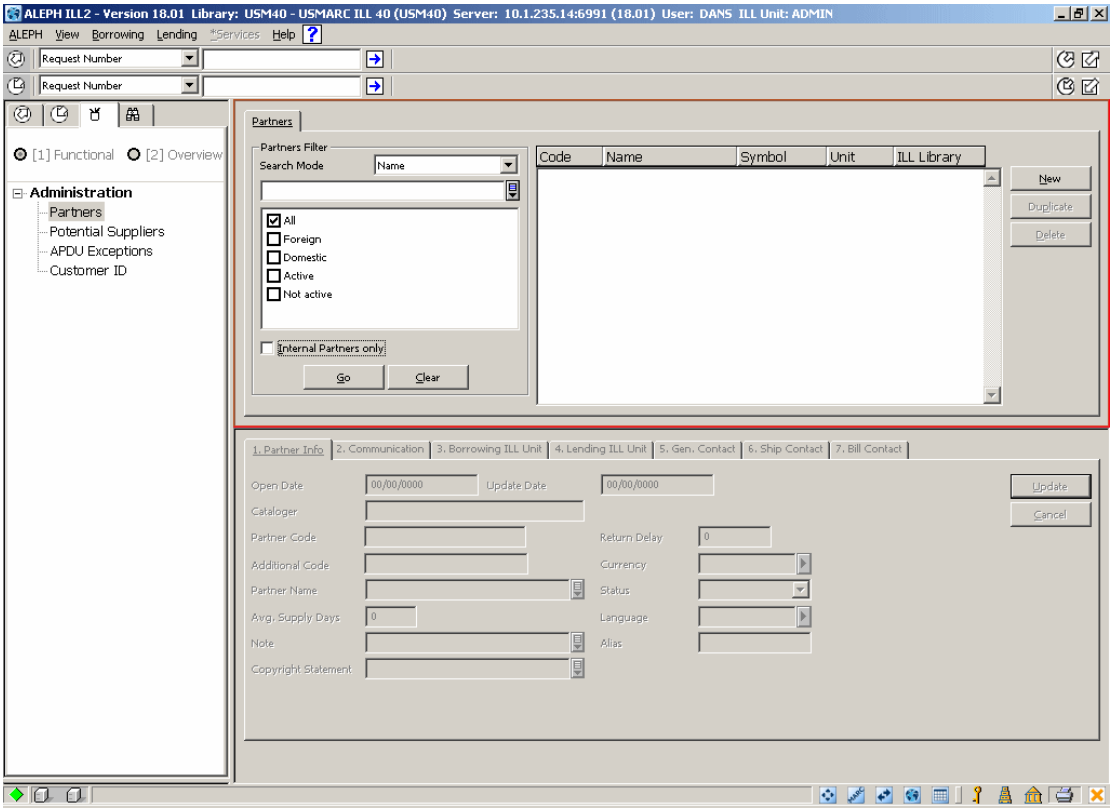
Companion parallel patron records must also be opened for each ILL partner (see section 2.6 Setting up external partners on page 22).

ILL partner records are viewed, created and updated through the Administration/Partner node in the Administration tab of the Integrated ILL module. If you have completed the steps described in sections, you should now be able to open Integrated ILL as the ILL library administrator (ADMIN) and to connect to the ILL library.

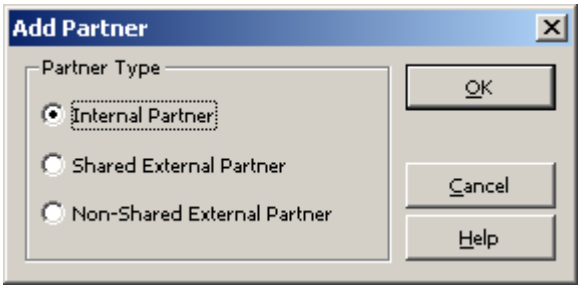
Note: If you are using a sequential input file in PLIF, make sure that the ILL unit name does not exceed five characters. This is the length of the field USER-REC-ILL-LIB in the sequential PLIF input file.

To create a new ILL unit

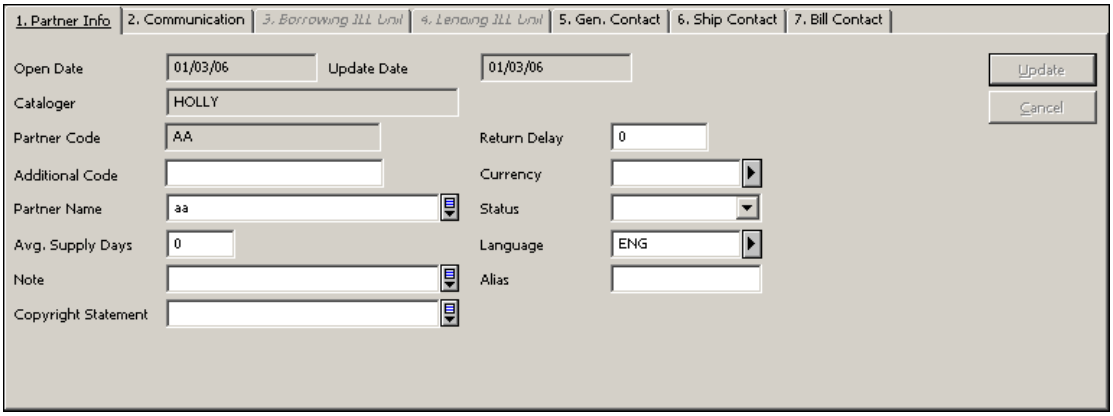
1. Connect to Integrated ILL as the ADMIN user. Select the **Administration** tab in Functional mode. The list of ILL partners (internal ILL units and external suppliers) is displayed in the upper-right pane. As you are setting up ILL units for the first time, it is empty:



2. From the Partners List, click **New**. The Add Partner dialog box appears:



3. Choose Internal Partner from the available Partner Type options and click **OK**. A multi-tab partner record displays in editing mode in the lower pane:



4. When you have finished filling in all relevant fields, click **Update**. The ILL unit is saved in the system.

Note

For detailed information about the various tabs and fields, refer to Integrated ILL online Help.

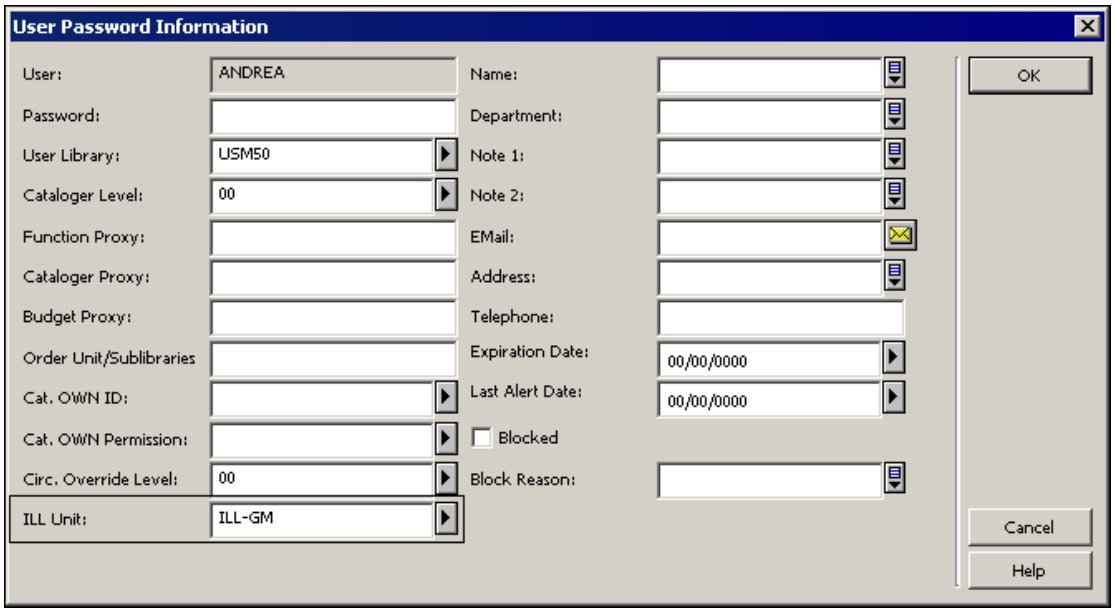
The next step in setting up Integrated ILL is setting up authorized users for each ILL Unit.

2.4 Creating an authorized staff user

In order to empower a staff member for working with the Integrated ILL module and processes, the user must be authorized for an ILL Unit, and relevant authorizations must be assigned to the user, through the Staff Privileges interface.

To assign an ILL Unit to an authorized staff user

1. Log in to either Circulation or Acquisitions/Serials.
2. From the status bar, right-click .
3. Click **Staff Privileges**.
4. In the Staff Privileges window, select the appropriate user and click **Modify User**. The User Password Information window opens:



5. In the ILL Unit field, type the appropriate name, for example, *ILL-GM*.
6. Click **OK**.

The authorized staff user must be assigned ILL privileges in the ILL library (XXX40), patron, circulation and items privileges in the ADM library (XXX50), and in privileges in the password library (pw_library USR00). See section 2.2 Setting up ILL library administrators on page 16 for instructions in assigning privileges.

The next step in setting up Integrated ILL is setting up updating privileges for the WWW-X and Z39 system users.

2.5 Setting up Updating Privileges for the WWW-X and Z39 system users

Libraries that use X-Services for external ALEPH locate and Z39.50 gate search need to assign appropriate privileges to the WWW-X and Z39 system users.

WWW-X and Z39 must be assigned privileges in the ILL Library (XXX40).

To assign ILL library privileges for WWW-X and Z39 users

1. In the Staff Privileges window, select the appropriate user (WWW-X or Z39) and click **Access Rights**.
2. Select the relevant ILL library (for example: usm40) and click **OK**.
3. Mark 'All Functions / All sub-functions' as 'Allowed'.

The next step in setting up Integrated ILL is setting up external partners (requesters and suppliers).

2.6 Setting up external partners

When you create a new partner record, you are prompted to choose the partner type. The choices available to you will be different depending on whether you are logged in as an ILL staff user or as an ILL library administrator (staff ILL unit=ADMIN):

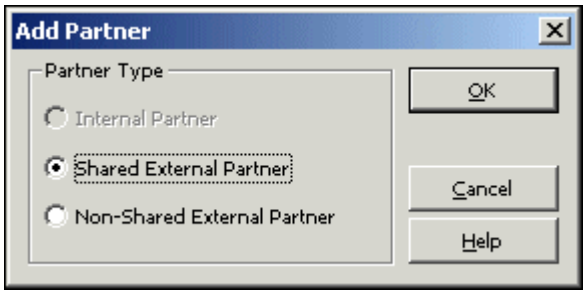
- An ILL staff user may only add External (shared and non-shared) partners.
- An ILL library administrator (staff ILL Unit=ADMIN) can add, update, and delete, both Internal and External (shared and non-shared) partners.

Partner type	*Explanation*
Internal Partner (ILL Unit)	An administrative unit that handles ILL activities for a sublibrary or group of sublibraries within a single ADM environment. You can create or update internal partners within your own ADM environment only. This choice is available only if you are logged in with ADMIN as your ILL Unit.
Shared External Partner	A supplying or requesting library which is not a part of your ALEPH system; it is external to your library system (for example another library or a supplier such as BLDSC). An external shared partner can be used as a supplier/requester by the ILL units of all ADM libraries.
Non-Shared External	A non-shared external partner is the same as a shared external partner, but can be used only by the Units within a single

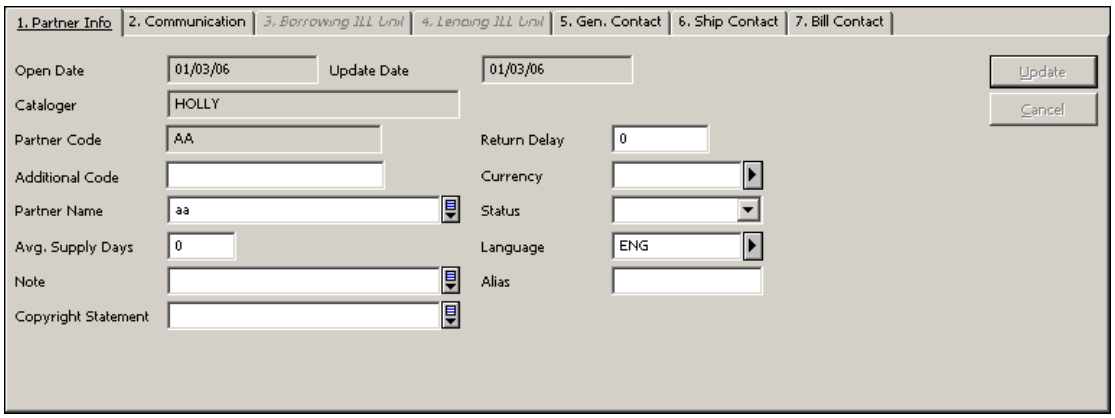
Partner type	*Explanation*
	administrative environment (ADM library).

To set up an external partner

1. Connect to Integrated ILL as either an ILL library administrator (staff ILL Unit=ADMIN) or as a staff user. Select the **Administration** tab in Functional mode. The list of ILL partners (internal ILL units and external suppliers) is displayed in the upper-right pane.
2. From the Partners List, click **New** or **Duplicate**. The Add Partner dialog box appears:



3. Choose a Partner Type from the available options and click OK. A multi-tab partner record displays in editing mode in the lower pane:



4. When you have finished filling in all relevant fields, click **Update**. The external partner is saved in the system.

Note

For detailed information about the various tabs and fields, refer to Integrated ILL online Help.

To	*Do this*
Update a partner	In the Partners list, select a partner. The appropriate partner record displays in the lower pane in editing mode.
Delete a partner	In the Partners list, select a partner, and

To	*Do this*
	click **Delete**. When a partner record is active (has linked active requests), it cannot be deleted.
Duplicate a partner	In the Partners list, select a partner, and click **Duplicate**.

The next step in setting up Integrated ILL is creating patron records for the ILL partners (internal and external) that are used as ILL requestors (that is, your library is used as their supplier.

2.7 Creating patron records for ILL partners used as ILL requestors

In order to allow circulation transactions for the ILL partners (loan, return, recall and renew), each ILL partner must have a global and a local patron record in the ALEPH system.

The ILL partners' patron records are viewed, created and updated through the Patron Registration node in the Patron tab of the Circulation module (see section 2 of the Circulation section of the User Guide for an explanation of how to create and update patron records). You will probably want to set aside a patron status for ILL partners.

To set up a patron record for an ILL partner

In the Circulation GUI interface, create a global patron record for the ILL partner, entering at least the following:

Global Patron Information

- Patron ID: ILL Partner Code
- Name: ILL Partner Name
- Pin Code, Barcode, Barcode Verification and Language (can be system-generated)

ALEPH Local Patron Information

- Status: ILL Borrower Status
- Expiry Date: Dec.31, 2099
- Photo Charge: F
- Privileges: irrelevant, but must be set

The next step in setting up Integrated ILL is creating a roster of potential suppliers for borrowing requests.

Note: Patron records are automatically created if the responder of an incoming lending ISO ILL request (Z700 record) is found to not have a matching patron record. The patron information is automatically populated as described in this chapter. The Status of the Local Patron Information record is set according to the value that is defined in the ill_borrower_status environment variable.

2.8 Creating a roster of potential suppliers (for borrowing requests)

A roster of possible suppliers is used to assign the suppliers to which a borrowing (outgoing) request will be sent. A roster is required for unmediated handling of requests; it is also an aid when mediated handling is performed. Each ILL Unit is responsible for creating its own roster of ILL partners.

The roster comes into play in the following manner – when a borrowing request is processed, the system attempts to '*Locate*' the material on each relevant supplier's database, and to identify which suppliers from the roster have available items. "Successful" suppliers are assigned to the request. When a supplier is unable to fulfill the request, the request is sent to the next supplier in the list, and so on. The order of the suppliers is determined either by their order in the roster, or by random selection (depending on the profiling of the suppliers in the roster).

In case of SLNP, the potential suppliers are determined by the central ILL server. When a supplier is unable to fulfill the request, the central ILL server sends the request to the next supplier in its list. When communicating via SLNP, it is not necessary to configure a roster of potential suppliers.

The roster can include a "last resort" supplier (group level 99). A request that cannot be fulfilled by any of the other suppliers will be sent to this supplier, without checking for availability.

Each ILL Unit configures its own roster of potential suppliers, for each type of material (for example, serials, a-v) and for the format in which the material should be supplied (returnable, non-returnable, hard copy, electronic copy, and so on).

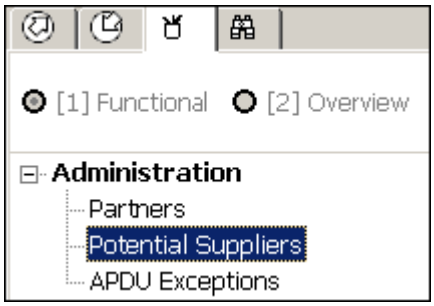
Features of the roster are:

- Each ILL unit controls its own roster
- Entries in the list can be set for particular types of material (for example, Machine-Readable, Serials)
- Entries in the list can be set for particular methods of supply (for example, electronic, loan, photocopy, electronic copy)
- Each entry (supplier) in the list is profiled for:
- Base which is searched. Note: when adding a new line to the potential supplier list, the system will automatically fill in the supplier's default *Locate* Base (as defined in the supplier's partner record). In the event that you want the system to use another base (not the one that is defined in the supplier's partner record), select a base from the list.
- Average Supply Days- Number of days it takes for material to be supplied

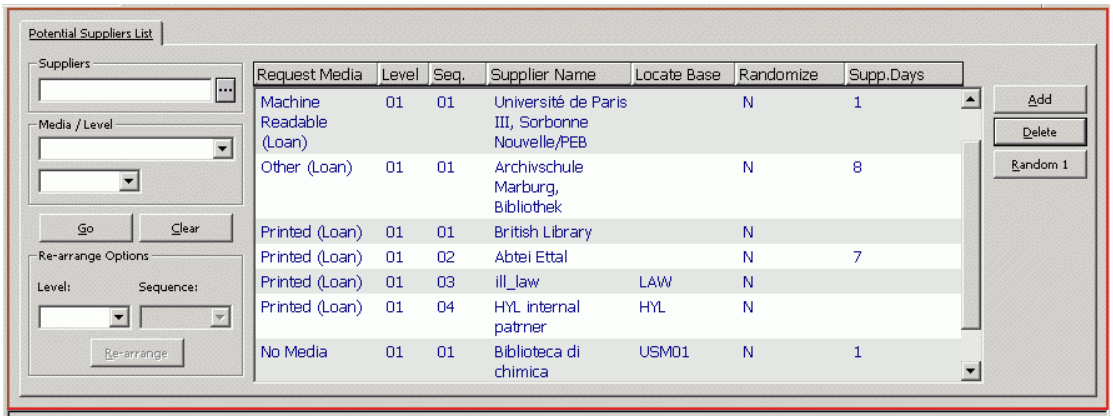
- Return Delay - How long it takes to return material that has been supplied
- Expiry Days – Relevant only for ISO ILL. Number of days after which the request expires.
- The roster can contain levels (groups)
- The suppliers within a level can be sequenced, or can be randomized

To set up a roster

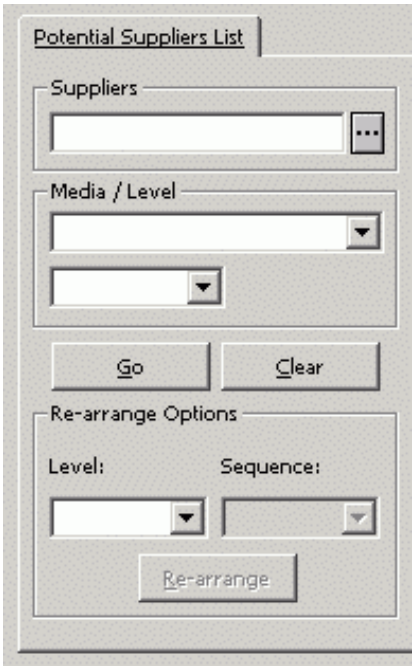
In Integrated ILL, choose the Potential Suppliers node in the Administration tab:



The full list of potential suppliers for the selected ILL units displays in the upper right pane (assuming you have set some up):



You can filter the list by filling in one or more of the fields at the left of the Potential Suppliers list:

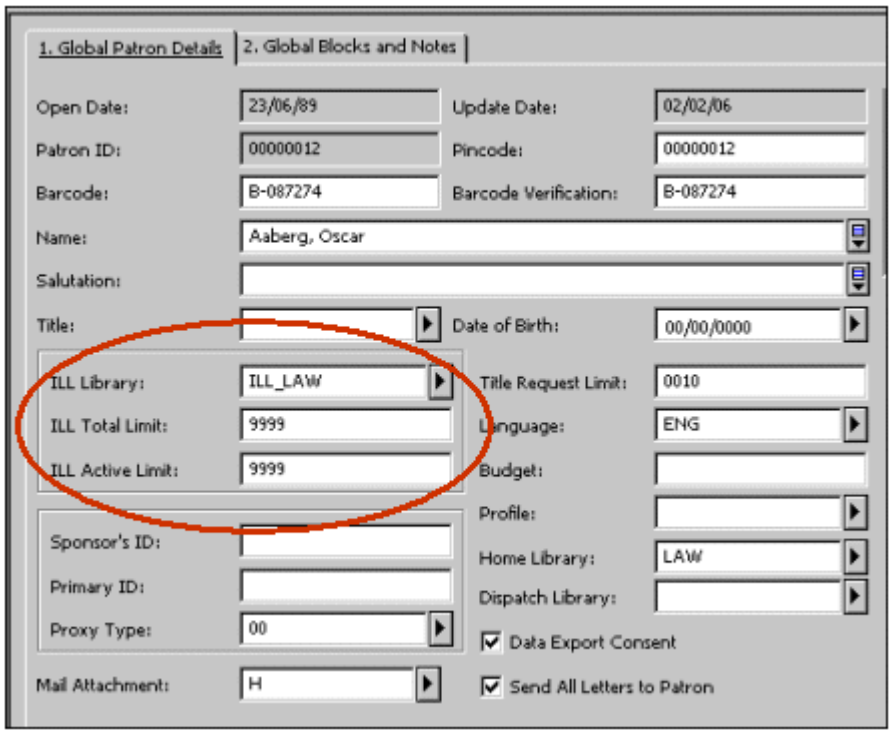


To	*Do this*
View a list of the potential supplier records for one partner	In the Potential Supplier's list, select a Supplier from the **Suppliers** drop-down list and click **Go** (the drop-down list is ordered alphabetically by Partner code). If you want to filter the resulting list further, for example, to see if a potential supplier record exists for the media type Microform (copy), select a media type from the **Media/Level** drop-down list and click **Go**.
View a list of all partners with potential supplier records of one particular media type	In the Potential Suppliers list, leave the Supplier drop-down list empty, choose a media type from the **Media/Level** drop-down list and click **Go**. This list will show you the order in which the system searches for potential suppliers within a media type. If you want to further limit this list so that you only see the group of potential suppliers for one media type within one specific level, select the level (01, 02 and so on) from the **Media/Level** drop-down list and click **Go**.
Start over and perform a different search	In the Potential Suppliers list, click **Clear** to clear the search filters.
Rearrange the order in which the system sets potential suppliers within a group, or	In the Potential Suppliers list, select a potential supplier, choose **Level** from the

To	*Do this*
to move a supplier from one group to another	**Rearrange Supplier** section, choose the Sequence number that you wish to assign to the highlighted supplier, and click **Rearrange**. The potential supplier record will move to the appropriate place in the list.
Randomize the order of the potential suppliers within a level	In the Potential Suppliers list, select one of the lines in the level and click **Randomize** (if the group of suppliers is already randomized, the button will be labeled **De-Randomize**, in which case the suppliers will be sequenced).
Add a new potential supplier	In the Potential Suppliers list, Click **Add**. An empty entry form will open in the lower pane.
View and/or update a potential supplier entry	Select the relevant line in the Potential Suppliers List. The lower pane displays a tab containing the supplier roster information and another tab with the supplier roster form for updating.
	Note For detailed information about the various tabs and fields, refer to Integrated ILL online Help.
Remove a potential supplier from his place in the roster	In the Potential Suppliers list, select a roster line and click **Delete**. This removes the highlighted roster line; it does not remove all the roster lines of this supplier.

2.9 Assigning ILL units to library patrons

Library patrons who have ILL privileges must have an ILL unit assigned in their Global Patron records. You do this in the Circulation GUI (see section 2 in the *Staff User's Guide – Circulation*) by assigning the ILL Unit that serves the patron, and the request limits – total number of requests allowed within a time period (the time period is set by the library, for example, a year or a semester) and the total number of active requests allowed:



Additional checks and blocks, active for limiting ILL privileges (for example, patron owes fines, patron has overdue loans), can be set within the configuration of patron circulation checks.

2.10 Create Customer ID Records

To create a Customer ID Record perform the following procedure:

1. Open Integrated ILL.
2. Connect to the XXX40 library.
3. Select the Administration tab.
4. Select Customer ID.
5. Click **New**.

For more details, see the *Aleph 20.0 Staff User's Guide - Integrated ILL*.

3 ILL *Locate* Function

The ILL *Locate* function is a major feature which is involved in the ILL flow of both borrowing (outgoing) and lending (incoming) requests.

The '*Locate*' can be manually activated by staff users OR automatically by activating the ILL library daemon: ue_19 (see chapter 6 - ILL Library Daemon).

Borrowing *Locate* – the borrowing *Locate* action is taken when a patron's borrowing (outgoing) request is handled. The '*Locate*' is invoked when the system creates the potential suppliers list for a borrowing request. The system attempts to *Locate* the requested title in the partner records which are set in the pre-defined roster of potential suppliers. The initial search is a 'title search' for a matching BIB record,

followed by an 'items search' for a matching holding record. The 'items search' is performed only for suppliers whose holding information can be accessed (depends on the partner's '*Locate* type'). Eventually, only suppliers for whom the *Locate* action has succeeded are assigned to the borrowing request.

Lending *Locate* - The '*Locate*' is invoked when a lending request is checked by the ILL unit to see if it can be fulfilled. The ILL unit base (as defined in the ILL Unit record) is searched according to the definitions set in /alephe/tab/tab_base.eng. The *Locate* action is successful only if a matching BIB record is found and available item/s located for supply.

This chapter describes the considerations that are used by the *Locate* process to reach these goals.

A supplier that has been set in the roster of suppliers will be considered as potentially able to fulfill a request if the following two conditions have been met:

- The potential supplier is available
- The required title is at the supplier's disposal.

3.1 Supplier Availability

A supplier is considered available if the following is true:

- The supplier's status is active, that is, its status is not non-active (Z700-STATUS 'NA').
- The supplier's average supply time does not fall after the request's last interest date.

Any supplier that does not fill these two requirements will not become a potential supplier.

3.2 Title Availability

Title availability depends on two criteria:

- A valid title is found.
- The title has valid items.

3.2.1 Searching for Titles

The potential suppliers are scanned for items that match the requested title. If the requested title already includes information on the linked holdings (the 954 field), and the linked holdings belong to the ADM library from which the supplier can lend (Z700-CIRC-SUB-LIBRARY), then no new search method is required. This is potentially true wherever the requested title has been harvested from an ALEPH catalog.

If no holdings information exists, or if the holdings do not belong to the sublibrary/ies from which the supplier can lend, then a new search method is required. This will always be the case in a lending *Locate*. This will be the true for a borrowing *Locate* only if the requested title was harvested from a non-ALEPH catalog.

If a new search is required, then the method of performing the search depends on how the partner record of the supplier has been defined in your ILL data. The *Locate* procedure behaves according to the partner *Locate* type which is defined in the '*Locate* Type' field of the partner record.

There are six possible *Locate* types: UNIT, UNITLESS, ALEPH, ITEMLESS, EXTERNAL and NONE.

Note

The UNIT and UNITLESS *Locate* types are only relevant for internal ILL partners (ILL Units); therefore, both borrowing and lending *Locate* actions are relevant for these *Locate* types.

All other *Locate* types, ALEPH, ITEMLESS, EXTERNAL, and NONE, are assigned to external partners which are used as your suppliers; therefore only the borrowing *Locate* action is valid for those types. Partners of type ALEPH, ITEMLESS, EXTERNAL and NONE can also act as requesters but in this case their *Locate* action is defined according to their ILL policy and setup.

UNIT: ILL Unit - The partner is one of the defined ILL units and is an internal partner that represents an administrative unit that handles ILL activities for a sublibrary or group of sublibraries within a single ADM.

- **For borrowing *Locate*** - The search in the partner's database is locally (within your ALEPH installation) and uses the holding information (field 954). If there is no holding information, the search is performed in the ILL unit's base as defined in the Locate Base field of the ILL Unit's partner record. The base must be set in your ./alephe/tab/tab_base.eng. The search query is according to borrower's ./<*ill library*>/tab/tab_locate (there is an option to define a tiered *Locate* search - see details in the Tiered Title *Locate* section below).
- **For lending *Locate*** – the local ILL unit's base is searched as defined in the *Locate* Base field of the ILL Unit's partner record. The base must be set in your ./alephe/tab/tab_base.eng. The search query is according to your ./<*ill library*>/tab/tab_*Locate* (there is an option for tiered *Locate* - details in sub-section section below -Tiered Title *Locate*) .

UNITLESS: ILL Unit with no items data - same as *Locate* type UNIT but for units that do not hold items data in ALEPH format. For a UNITLESS partner, the system attempts to search for a matching title but does not proceed to the item searching step. The UNITLESS *Locate* type is valid for both the borrowing request *Locate* and the lending request *Locate*.

ALEPH: ALEPH system - The partner has full ALEPH ILS and uses X-Services to fetch title and items information. The host address to which the X service will be sent is defined in the partner's *Locate* Server which is defined in the Locate Server field of the partner record. The supplier's base is defined in the Locate Base field of the partner record and must be set in your ./alephe/tab/tab_base.eng. The search query is according to the supplier's ./<*ill library*>/tab/tab_*Locate*. The ALEPH *Locate* type is relevant only for a borrowing *Locate* when your library attempts to assign an ALEPH library as a supplier.

ITEMLESS: ALEPH system with no items data - Same as the ALEPH *Locate* type but for systems that do not hold items data in ALEPH format. For an ITEMLESS partner, the system attempts to search for a matching title but does not proceed to the item searching step. The ITEMLESS *Locate* type is relevant only for a borrowing *Locate* when your library attempts to assign an 'ALEPH- no items' library as a supplier.

EXTERNAL: External base - The supplier's search base is on a non-ALEPH installation. The required record is not at hand and a Z39.50 query is used to fetch the record and items. The supplier's base is defined in the Locate Base field of the partner record. The supplier base must be set in your ./alephe/tab/tab_base.eng and relevant Z39.50 base configuration must be set in ./alephe/tab/z39_gate. The EXTERNAL *Locate* type is relevant only for a borrowing *Locate* when your library attempts to assign a Z39.50 library as a supplier.

NONE: No *Locate* - No attempt to perform a *Locate* action in the partner's database. This type is valid only for the borrowing requests *Locate* action when your ILL unit tries to assign the supplier to a request: the supplier will be considered as a 'match' even thought the title has not been located.

Each one of the above *Locate* types requires additional information for the *Locate* action. The following are the relevant partner's fields for this information:

- Catalog Type – The partner's catalog structure. Relevant for *Locate* types: UNIT, UNITLESS, ALEPH and ITEMLESS. The options are:
  - UV – ALEPH Union View
  - SBIB – Single BIB library
- Locate Server – The partner's server IP and port of the Web OPAC, for example, goldcoast01:8991. Relevant for *Locate* types: UNIT, UNITLESS, ALEPH and ITEMLESS.
- Locate Base – The base of the partner's library. The base must be one of the bases defined in the ./alephe/tab/tab_base.lng. However, you can trim down the list of bases the drop down list by listing them in the ILL-LOCATE-BASE section of ./<ill library>/tab/pc_tab_exp_field.lng . Relevant for *Locate* types: UNIT, UNITLESS, ALEPH, ITEMLESS, and EXTERNAL.
- Locate Circ. Library – The partner's ADM library code, for example, MSL50. Relevant for Locate types: UNIT and ALEPH. Drop-down menus can be set in ILL-LOCATE-LIBRARY menu of ./<*ill library*>/tab/pc_tab_exp_field.lng.

3.2.2 Tiered Title *Locate*

When tab_locate queries are invoked in the *Locate* action, you can define tier-locate procedures so that the *Locate* will be performed in levels. This can be done by setting locate-base extensions in tab_locate. For example, for a base with the code WID, you can define a tier-locate by adding an extension to the base's code: WID-1, WID-2, WID-3, and so on.

The *Locate* will first be performed according to the procedure defined in tab_locate for WID-1; if this fails, the system applies the WID-2 procedure (and so on, until WID-9). The WID procedure will be consulted when all previous tiered *Locate* procedures have failed.

Make sure that each *Locate* base is defined in /alephe/tab/tab_base.eng but not the tier-locate base extensions, for example, WID should appear, and not WID-1.

Example of setting a three-level tiered-*Locate* for the base code WID (WID-1, WID-2 and WID) in /<*ill library*>/tab/tab_locate:

```
WID-1                          245##  a                                wti=
Locate_str_0
WID-1                          100##  a                                wau=
Locate_str_0
WID-1                          260##  a                                wpl=
Locate_str_0
WID-1                          260##  b                                wpu=
Locate_str_0


WID-2                           245##  a                               wti=
Locate_str_1
WID-2                           100##  a                               wau=
Locate_str_1
WID-2                           260##  a                               wpl=
Locate_str_1


WID                            245##  a                                wti=
Locate_str_1
WID                            100##  a                                wau=
Locate_str_1
```

3.2.3 Item Searching

For the *Locate* types UNIT and ALEPH; after the requested title is at hand, the title's items are checked according to the following sequence:

1. The matching bibliographic record is searched for items. If the request is for a serial item, then the items must match the volume's year that is required.
2. The found items are checked. They must belong to one the sublibraries that are used by the potential supplier for lending. This is defined in the partner's Circ Sublibraries field.
3. The ability to loan the detected item to the ILL requester is checked according to the supplier's ./adm_library/tab/tab_check_circ. The relevant sections of tab_check_circ are: ILL-L and ILL-C. Section ILL-L is relevant for a returnable request (loan) and section ILL-C is relevant for a non-returnable request (copy or electronic supply).
4. For a lending *Locate* only: the items are checked to see that they are not currently on loan.

3.3 '*Locate*' Possible Results

If all the above checks (title searching and item searching when possible) are successfully passed (that is, an available item which can be loaned to the requester has been found), then the request is considered as successfully located:

- **Borrowing request** – possible suppliers are assigned to the request and its status becomes 'Waiting for Process' – WAP
- **Lending request** – possible items for supply are attached to the request and its status becomes '*Locate*d' – LOC.

If the above checks fail (that is, no matching title is found or no matching items for supply are *Locate*d), then the request is considered as failed:

- **Borrowing request –** suppliers are not attached to the request and its status becomes 'Locate Failed' – LOF.
- **Lending request** – if more then a single BIB has been found, no items are attached to the request and its status becomes 'Multiple Locate' – MLC. If no title is found, or a title is found but no items are found, the request status can change to 'Mediated' – MED, or to 'Answer Unfilled' – AUF (depending on the ILL Unit setup); in the case of Answer Unfilled, a message is sent to the requester. This is a terminal status.

4 Background Processes

4.1 XXX40 ILL Library

- library batch queue - UTIL C/2
- update documents index - ue_01 (UTIL E/1); the process is stopped using ue_02 (UTIL E/2).
- Optional – ILL Library daemon: ue_19 (UTIL E/19); the process is stopped using ue_20 (UTIL E/20) – see details in the chapter: ILL Library Daemon: ue_19.
- Optional - SLNP server (to start with UTIL W/3/7/5); this server enables communication via the SLNP protocol.

4.2 ILLSV Library

The ILLSV library (ILL Server Library) is relevant only in sites that use the ISO ILL protocol type. The ILLSV library is used as repository for all ILL APDU messages that are received at your site.

- update documents index - ue_01 (UTIL E/1); the process is stopped using ue_02 (UTIL E/2).
- handle ILL server library document - ue_17 (UTIL E/17); the process is stopped using ue_18 (UTIL E/18).
- ill_server (to start: UTIL W/3/7/7); this server directs messages from the ILLSV library to the relevant library.

4.3 General

- pc_server (to start: UTIL W/3/3) for GUI ILL Client
- www_server (to start: UTIL W/3/1) for Web OPAC
- Z39_gate (to start: UTIL W/3/4); this server enables searching in remote external databases (external partners).
- Z39_server (to start: UTIL W/3/5); this server enables external users to search in ALEPH databases.

5 Setting up Unmediated Mode for ILL Units

You can set your ILL Unit to handle ILL borrowing and lending request in an 'unmediated mode'.

For the borrowing ILL workflow – the unmediated setup supports the option to *Locate* potential supplier to new requests and to automatically send the request to the suppliers.

For the lending ILL workflow – the unmediated handling supports an automatic *Locate* of all NEW lending requests and support the option to respond to an 'non-*Locate*d' requests.

5.1 Setting Up the 'Unmediated Mode'

Steps

1. Connect to Integrated ILL as a staff user.
2. **For Unmediated handling of borrowing requests** – In the partner record of the ILL Unit, click the Borrowing ILL Unit tab and select the Borrowing Unmediated Locate check box.

   For Unmediated handling of lending requests – In the partner record of the ILL Unit, click the Lending ILL Unit tab and select the Lending Unmediated *Locate* check box of the field.
3. In the server file system, activate UTIL E/1 then UTIL E/19, the ILL library daemon (ue_19). For an explanation of the results of this procedure, see sections 5.2 Unmediated Activities - Borrowing below and 5.3 Unmediated Activities - Lending on page 38.

5.2 Unmediated Activities - Borrowing

When ue_19 is activated and the ILL Unit's Borrowing Unmediated Locate check box is selected, the system attempts to *Locate* potential suppliers for each NEW patron ILL request (Z410), using the roster of potential suppliers (Z701).

Only borrowing requests which have a status of NEW are handled by the daemon. Requests that have been initially created with a status other than NEW will not be handled by ue_19, that is, a 'Locally Owned' request (LOW) and a New-Staff Review request (NEM:-request that contains patron note or staff note) will not be handled by ue_19. If you want LOW and NEM requests to be handled by the daemon, you will have to manually change their status to NEW (see section 9 Change Request Status on page 41).

If the *Locate* fails to find a potential supplier, the patron request status (Z410-STATUS) is changed to '*Locate* failed' (LOF). A '*Locate* Failed' request has to be handled manually by staff users.

If the *Locate* ends successfully, a list of suppliers (Z411s) is created for the request. Each request to supplier is assigned a sequential number (Z411-SEQUENCE) representing its position in the list. The request to supplier with first priority is assigned the status NEW (Z411-STATUS). All other requests to supplier are assigned the status PND (Pending).

Actually, the *Locate* method and results are the same as if the '*Locate*' has been manually activated by a staff user (by clicking **Locate** in the Borrowing Requests List). As for *Locate* procedures, see section 3 ILL Locate Function on page 29.

At this stage (when a list of potential suppliers is attached to the request), ue_19 invokes tab50 (./<*ill library*>/tab/tab50) to determine what is the next action to

perform for each supplier request, according to the filters defined in Col.1 through Col.5 (see section 5.2.1 Defining how borrowing requests are handled - tab50 below).

You may decide that ue_19 should be used only for *Locate* potential suppliers but not for sending the requests to suppliers. If so, leave tab50 empty. Otherwise, you will have to set tab50.

5.2.1 Defining how borrowing requests are handled - tab50

./<*ill library*>/tab/tab50 is involved when the *Locate* of potential suppliers for a request is successful (patron request status is 'Waiting for Process'-WAP). The system refers to tab50 in order to perform the SEND or CANCEL action,

tab50 is also involved when a supplier reply is 'Unfilled' and there is an option to send the request to the next supplier in the list.

tab50 has the following structure:

Col. 1 - ILL Unit – The ILL unit that created the request.

Col. 2 – Supplier code – The code of the potential supplier.

Col. 3 – Media – The requested media (for example, L-PRINTED, C- PRINTED)

Col. 4 - Supplier request status (Z411-STATUS)

Col. 5 – Range of supplier request sequence (Z411-SEQUENCE) – A number that represents the place of the supplier in the list of potential suppliers. It can be set as single sequence (for example, "1" for the first supplier) or as a range of sequences (for example, "2-9" for second through ninth)

Col.6 – Action- the action that is taken: SEND or CANCEL

The possible actions (Col.6) are SEND and CANCEL:

SEND – this is an automatic attempt to send the request to the supplier. *Locate* has already been successfully performed by ue_19, and a request message is sent to the current active supplier. The Request to Supplier Status is changed to 'Sent to Supplier' (SV). The dispatch of the request to the supplier will be as defined in the supplier's partner record: ISO, Letter format (option for e-mail) or BL format. If sending cannot be performed due to a technical problem, the status of the request to supplier is changed to ENN (where NN represents the number of failures), and the status of the main request is changed to Daemon Send Failed (DSF).

For example: If it is an ISO request and the server's attempt to send a request to a supplier fails the first time, the request to supplier status is changed to E00; the second failure will change the status to E01, and so on. In this way, the library can set how many times a request transmission should be attempted. The main request becomes DSF (Daemon Send Failed).

CANCEL - Cancellation of a request to a supplier before it has been sent. If there is a technical problem in dispatching the ILL Request to the target server (due to a problem in the target server), you can define a limit for the number of attempts for a request to be sent to a supplier. When this limit is reached, the request will be automatically canceled (request to supplier status changed to Canceled-CA). This will trigger the activation of the next supplier (that is, the request to supplier will be changed from Pending-PND to NEW).

Suggested setup:

You may want ue_19 to automatically create a list of potential suppliers for all NEW requests and then have a staff user review/rearrange/amend the list. After the staff user has 'confirmed' the list, he can manually send the first request to supplier (SEQUENCE is 000000001) by clicking **Send**.

From now on, you may want that in the event that the first supplier cannot fulfill the request, ue_19 will automatically send the request to the second supplier. If the second supplier also fails to supply the request, the third supplier will become active and so on.

In addition, you can determine that in the case of a technical failure in the dispatch of the request (for example, the target ILL server is shut down), only three attempts to send the request to the 'failed' supplier are allowed. After three unsuccessful attempts, the request to that supplier is canceled and the request moves to the next supplier in the list.

To support the above scenario, set tab50 according to this sample:

```
ILL_LAW ################### ################### NEW 2-99 SEND
ILL_LAW ################### ################### E00 2-99 SEND
ILL_LAW ################### ################### E01 2-99 SEND
ILL_LAW ################### ################### E02 2-99 SEND
ILL_LAW ################### ################### E03 2-99 CANCEL
```

5.3 Unmediated Activities - Lending

You can set your ILL Unit to handle ILL lending request in 'unmediated mode'.

The lending '*Locate*' action that checks if an ILL request can be filled by your ILL unit can be automatically activated by ue_19.

When ue_19 is activated and the ILL Unit field 'Lending Unmediated *Locate*' is set to Yes, the system attempts to *Locate* matching items for supply for all NEW lending requests.

The *Locate* method and results are the same as if the *Locate* has been manually activated by the staff user (by clicking **Locate** in the Lending Requests List). As for *Locate* procedures, see section 3 ILL Locate Function on page 29.

If an available item which can be loaned to the requester has been found, then the request status is changed to *Located* (LOC). The list of *Located* items can be viewed in the Lending Request list, Items tab.

If more than one BIB has been found, no items are attached to the request and its status becomes 'Multiple *Locate*'-MLC.

If no title is found, or a title is found but no items are found, the request status changes to 'Mediated'-MED, or to 'Answer Unfilled'-AUF (depending on the partner record setup of the ILL Unit, shown in the Lending ILL Unit tab); in the case of Answer Unfilled, a message is sent to the requester.

Only lending requests with a status of NEW are handled by the daemon. Requests that have initially been created with a status other than NEW will not be handled by ue_19 that is, New-Staff Review (NEM- a request that contains a requester's note) will not

be handled by ue_19. If you want NEM requests to be handled by the daemon, you will have to manually change their status to NEW (see section 9 Change Request Status on page 41).

To print reports and/or pickup slips for lending located material, use the batch service Lending Library - ILL Requests Report and Slips (ill-65).

This service enables you to retrieve ILL lending requests, produce a report, and print a separate slip for each retrieval.

The service also enables you to update the status of Located, Multiple Located, and New requests upon printing slips. Request there were Located (LOC) are updated to Located and Printed (LPR), requests that were Multiple Located (MLC) are updated to Multiple Located and Printed (MLP) and request that were in NEW status are updated to New and Printed (NEP).

6 Setting up “Locally-owned” checks

You can set your system to check if a title that is being requested via the ILL borrowing services is locally owned by the library. The check is performed when a new ILL request is submitted by a patron via the OPAC or by an ILL staff user via the Integrated ILL GUI (see section 6.2 Setting the system’s response to locally-owned results on page 40).

For an ILL request submitted by a patron, you can set the action to be taken by the system for a locally-owned request (see section 6.2 Setting the system’s response to locally-owned results on page 40).

6.1 Setting up the 'Locally-owned' policy

The 'Locally-owned' policy is defined in the partner record of each ILL unit, Borrowing ILL Unit tab, Check Local Own Procedure field.

The options are:

- **0 - No check** - The system does not check whether or not the requester’s library holds a requested item.
- **1 - Check for items existence** – If the requested item is held by the library, the request will be marked as 'Locally owned'. No further checks for patron blocks and/or item availability are performed.
- **2 - Allow ILL if patron is locally blocked** - If a patron cannot borrow a specific item from his library (because of patron blocks), he will be allowed to request the item via the ILL service. The system marks the ILL request as 'Locally Owned'.
- **3- Block ILL if patron blocked/items avail. locally** - If a patron cannot borrow a specific item from his library (because of patron blocks), he will not be allowed to use the ILL service to get the item from another library. The system will mark his request as 'Locally Owned'. In addition, if the patron is not blocked but items are not available, the system will also block his ILL request (by marking it as "Locally Owned').

For procedures 2 and 3:

The ADM library table, tab_check_circ (./xxx50/tab/tab_check_circ) sets the checks that will be performed when the self-ownership check is activated.

`The relevant` entries of tab_check_circ are: `ILL-L` (for loan request) and `ILL-C` (for copy request and electronic).

6.2 Setting the system's response to locally-owned results

For an ILL request submitted by a staff user via the ILL GUI: Once the request is registered and detected as Locally Owned, the staff user is prompted with a notification: 'Locally Owned Request'. Continue Yes or No.

If the staff user selects Yes - the request is created with the status Locally Owned (LOW).

If staff user selects No - the request is not created.

For an ILL request submitted by patron via Web OPAC: Once the request is registered and detected as Locally Owned, the system applies to ./<*ill library*>/tab/tab100 switch PATRON-LOCAL-OWN and acts according to its setup:

The value can be: 0 or 1

0 = (default) locally owned requests are blocked.

1 = locally-owned requests continue to be processed and they get the status

'Locally Owned' (LOW).

For a system wide setting, refer to Section 25 tab100 - Related Entries for ILL section.

7 Setting up patron checks when a new borrowing request is submitted

You can define check routines for local patrons who submit a new borrowing request.

The check routine is defined in the ADM library table: tab_check_circ (./xxx50/tab/tab_check_circ). The relevant check procedure (Col.1 in tab_check_circ) is PAT-DELINQ.

Example of PAT-DELINQ entries in tab_check_circ:

```
PAT-DELINQ check_circ_1_a
PAT-DELINQ check_circ_1_b
PAT-DELINQ check_circ_1_c_a
```

Note

The check routines (col. 2 of tab_check_circ) can be only routines that check the patron global (Z303) and patron local (Z305) record

Libraries that do not use the patron's Home Library (Z303-HOME-LIBRARY), should not be defined in PAT-DELINQ checks that are related to the patron's local record (Z305).

The PAT-DELINQ procedure check is consulted when the patron submits a new ILL request via Web OPAC or when the staff user submits a new ILL request on behalf of the patron via the ILL GUI.

Web OPAC ILL request - If the request is blocked, the patron gets a block message and cannot proceed with the ILL request. He can apply to a staff user to clarify the issue.

ILL GUI request – If the request is blocked, the staff user is prompted with a warning message: Patron has delinquencies. Continue Yes/No. The staff user can decide to abort the ILL request (selecting: 'No') or proceed with the ILL flow (selecting: 'Yes').

8 Setting Up Non-working Days of Potential Suppliers

You can define the-non working days of potential suppliers in tab_ill_supp_close_days. This table is consulted when calculating the Expiry Date.

tab_ill_supp_close_days has the following structure:

COL-1 - ILL Supplier (or "####################" for all Potential Suppliers)

COL-2 - Date (for example, 20070101 or ####0101 or ####08## or ########)

COL-3 - Day of Week (00 = Sunday, 06 = Saturday)

COL-4 - Status (Default is Closed; C=Closed; O=Open, used to define exceptions)

When the table does not exist or when it remains empty, the Expiry Date is calculated as if all days are working days.

9 Change Request Status

Staff users can update the ILL request's status manually, by clicking the Change Status button in the borrowing and lending requests list. This is valid for both borrowing and lending requests.

The Change Status button works according to the definitions set in the following tables:

./alephe/tab/tab_ill_change_status_out – for borrowing (outgoing) requests.

./alephe/tab/tab_ill_change_status_in – for lending (incoming) requests.

The above tables define from which status code (Col.1) the Change Status action is allowed. The status codes to which the request can be changed are listed in Col.2.

Col.3 may be used to trigger some additional processing that will take place along with the status change. In tab_ill_change_status_out, this column may set with ill_update_out_sup_status to trigger an update of the active Z411 record with the same status as the Z410 record.

.

Example from tab_ill_change_status_out:

```
!1   2           3
!!!-!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
NEW CA
```

```
NEW DEL
EXP CA
EXP NEW
EXP DEL
CLS DEL
LOW CA
LOW DEL
```

A list of status codes and their textual translation can be found in /<*ill library*>/tab/pc_tab_exp_field.eng.

Relevant sections:

BOR-REQ-STATUS – list of borrowing requests statuses and their textual translation.

LEND-REQ-STATUS – list of lending requests statuses.

In the event that you want to update the same request status to more than one status, define more than one line for the same status. See the above example of tab_ill_change_status_: status LOW (Locally Owned) can be update to CA (Canceled) or to DEL (Deleted).

Note

Changing a request's status manually triggers a log transaction that is displayed in the GUI ILL's Request Log but no further actions or procedures. Therefore, the setup of tab_ill_change_status_out and tab_ill_change_status_in must be done with extra care to avoid a conflict with the request handling workflow (for example, a lending request that has just been loaned out to requester (status Sent on Loan - SL) should not be changed to deleted (code: DEL).

10 Configuring ILL Cash Transactions (tab18.eng)

You can determine how much a patron will be charged for different transactions by configuring the ADM library table, tab18.eng.

Col. 1 of tab.18.eng is the transaction number. Following are the transaction numbers that are relevant for ILL:

0011
Borrowing Requests- The charge that will immediately be placed in the patron's cash transactions list for placing an ILL request.

0015
Borrowing Requests - This transaction determines whether or not the patron will be debited when an item is received by the library to fulfill the request. The sum (Col.5) is not active. Only col. 4, Activate the transaction Yes or No, is taken into account. The sum is filled by the user when registering the material's arrival. The calculation is based on the <*ill library*> table, ill_bor_charge

0016
Lending Requests - This transaction determines whether or not the ILL requester will be debited by the library when supplying the requested material. The sum (Col.5) is not active. Only col. 4, Activate the transaction Yes or No, is taken into account. The sum is filled in by the user when supplying the material. The calculation is based on the <*ill library*> table: ill_lend_charge

Col. 2 of tab18.eng is the sublibrary code. For the ILL transaction (0011, 0015 and 0016): You can charge different rates depending on the home library of the ILL requester. For system that do no feed in 'home library' in the patron's record: set the ADM library code in this col 2 (for example, MSL50). Other option is to set the 'match-all' value: #####.

For more information about tab18.eng, see the *System Librarian's Guide -Circulation*.

Example:

```
0011 ##### ## ## Y 10.00      L ILL request
0015 ##### ## ## Y 0           L ILL material arrival
0016 ##### ## ## Y 0           L Library charge for incoming ILL
request
```

11 Borrowing - Charging Patrons for Arrival of ILL Material

To determine how much a patron will be charged when an ILL is received from a supplier in order to satisfy a request, configure the ILL library's ill_bor_charge table (./XXX40/tab/ill_bor_charge).

This default amount that is set by ill_bor_charge is displayed in the GUI ILL's Borrowing - Receive form and it can be modified by the staff user. The charge will be added to the patron's cash transactions list as soon as the "receive" procedure is completed. This table works in conjunction with the ADM Library table: tab18.eng transaction 0015.

Column	Purpose	Example	Description
1	ILL Unit	ILL_LAW	The borrowing ILL unit that handle the request for the patron.
2	Patron Status code	01 or any other relevant code or ##	You can charge patrons of different statuses different rates. This is the patron status in the patron's home library local patron record. For a system that does not use a home library: the patron's ADM local record is used. If no such local record exists, the patron's ALEPH local record is used.
3	Supplier Status	DO (for Domestic) or any other value or ##	You can charge different rates depending on the status of the supplier. This status is the two-character code that is entered in the Partner Status field of the supplier record. For the relevant value, see the ILL-PARTNER-STATUS section of /xxx40/tab/

Column	Purpose	Example	Description
			pc_tab_exp_field.eng.
4	Page Size	A3, A4, and so on.	You can charge different rates depending on the size of the paper that is supplied when satisfying photocopy requests. For relevant values, see the PAGE-SIZE section of /xxx40/tab/ pc_tab_exp_field.eng.
5	Media Type	C	Different rates can be charged depending on whether the supplied item is a loan (code: L) or a photocopy (code: C) or electronic (code: E) . If "L" or "E" is specified in this field, fields 8 and 9 are irrelevant.
6	Fixed charge 1	4.5	The library can decide that a basic handling rate should be charged in order to cover all of the request's expenditure.
7	Fixed charge 2	0	To charge an additional amount for the supply of a request that you want to identify separately from the amount specified in column 6, enter it here.
8	Fixed Charge 3	0	As column 7.
9	Start counting after page X	10	This field is only used if column 5 is set to C - Copy. If the library decides that, for example, only the first 10 pages of any supplied copy is accounted for in the fixed charges, subsequent pages may be individually charged for by filling in columns 9 and10. This column (column 9) specifies the number of pages that are included in the charge (in this case the value would be 10). To charge for ALL pages, set the value here to 0000.
10	Fee per Page	0.5	This field is only used if column 4 is set to C - Copy. This column is used in combination with column 9 and specifies how much should be charged for each individual page of a supplied photocopy.

12 Lending -Charging ILL requester for Supplying ILL Material

To determine the parameters for charging the ILL requester for a lending request when the 'ship' (supply) action is submitted, configure the ILL library's ill_lend_charge table (./XXX40/tab/ill_bor_charge).

The default amount that is set by ill_lend_charge is displayed in the GUI ILL's Lending Ship form and it can be modified by the staff user. The charge will be added to the ILL requester's cash transactions list as soon as the "ship" procedure is completed. This table works in conjunction with the ADM Library table: tab18.eng transaction 0016.

Column	Purpose	Example	Description
1	ILL Unit	ILL_LAW or ##	The lending ILL unit that handle the lending request.
2	Requester status	FO or ##	The requesting ILL partner's status.
3	Requester code	HVDLAW or ##	The requester is the library or institution that applies to your library, asking for ILL material. Enter in this field the partner code of the requester (which is same as the requester patron ID).
4	Level of service	< or #	The requested Level of Service. You can charge different rates depending on the 'rush priority' and required service. For the relevant value, see the LEVEL-OF-SERVICE section of /xxx40/tab/ pc_tab_exp_field.eng.
5	Supplied Media type	C or #	Different rates can be charged depending on whether the supplied item is a loan (code: L) or a photocopy (code: C) or electronic (code: E). If "L" or "E" is specified in this field, fields 8 and 9 are irrelevant.
6	Fixed charge 1	4.5	The library may decide that a basic handling rate should be charged in order to cover all of the request's expenditure.
7	Fixed charge 2	0	To charge an additional amount for the supply of a request that you want to identify separately from the amount specified in column 5, you may enter it here.

Column	Purpose	Example	Description
8	Start counting after page X	10	This field is only used if column 4 is set to C - Copy. If the library decides that, for example, only the first 10 pages of any supplied copy is accounted for in the fixed charges, subsequent pages may be individually charged for by filling in columns 9 and 10. This column (7) specifies the number of pages that are included in the charge (in this case the value would be 10). To charge for ALL pages, set the value here to 0000.
9	Fee per Page	0.5	This field is only used if column 4 is set to C - Copy. This column is used in combination with column 7 and specifies how much should be charged for each individual page of a supplied photocopy.

13 Patron's Copyright Agreement for Borrowing Requests

Patrons can be required to ‘sign’ a copyright declaration when they submit a new ILL borrowing request. A request that has not been marked as 'copyright signed' when the table’s configuration shows that it is a mandatory issue for the supplier (to whom the request is sent), will be blocked upon the Send' action.

13.1 Determining copyright policy (tab47.eng)

Tab47.eng of the ILL library’s tab47.eng table `(./xxx40/tab/tab47.eng)` determines the default copyright policy for the patrons’ Copyright Agreement and Copyright Block.

For each combination of supplier code (Col. 1) and requested material type (Col.4), the definition in Col. 2 determines if a patron's copyright agreement is required, and the setting in Col. 3 determines if the dispatch of the request to supplier is blocked in the event that the copyright agreement is not signed.

If a staff user tries to send a supplier's request for which copyright agreement is mandatory but has not yet been declared, he will receive the following block message: "Copyright agreement is missing".

The table structure:

Column 1: Supplier Code

Column 2: specifies whether copyright agreement is needed. Possible values: y (yes) or n (no).

Column 3: specifies whether the request will be blocked if a copyright agreement is not provided. Possible values: y (yes) or n (no).

Column 4: The borrowing request media code. The codes must be one of the followings: PRINTED, COPY, MICROFORM, VIDEO, AUDIO, MR, OTHER with one of the following prefix of L-, C- or E-. For example: L-PRINTED, L-VIDEO, E-MR, C-PRINTED.

Column 5: Textual description for Col. 4

Example of `tab47.lng` setup:

```
!!!!!!!!!!!!!!!!!-!---!----!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
UEA                Y    Y     C-PRINTED          Book (photo)
```

In this example, an ILL borrowing photocopy book request from supplier UEA requires the patron's copyright agreement, and if it has not been provided, the ILL request will be blocked. In such a case, the request cannot be sent to the supplier UEA until the copyright agreement is signed and the Copyright Signed field of the 'ILL Borrowing Request Details' is marked as 'checked'.

13.2 Producing a Copyright Agreement Letter

A copyright agreement letter can be produced either from the GUI ILL module or from the Web OPAC ILL request.

Copyright Letter from GUI ILL: The copyright letter can be sent to the patron by clicking the **Print** button in the Borrowing Requests List. The patron is required to sign the letter.

The letter is sent by e-mail to the patron and as soon as it is returned signed, the Copyright Signed check box of the ILL request can be manually selected.

Copyright Letter from Web OPAC: The patron can sign the copyright agreement via the Copyright Declaration on the Web OPAC ILL Request. The Copyright Restrictions can be part of the ILL request form. The relevant HTML page is

./wwwf_eng/new-ill-r-copyright and the following two parameters are part of the copyright agreement setup:

COPYRIGHT_MANDATORY and UPDATE_COPYRIGHT_LETTER.

Example from the HTML page: www_f_eng/new-ill-r-copyright:

```
<input type=hidden name="COPYRIGHT_MANDATORY" value="Y">

<input type=hidden name="UPDATE_COPYRIGHT_LETTER" value="Y">
```

The **COPYRIGHT_MANDATORY** parameter indicates whether signing the copyright agreement is mandatory for non-returnable requests. Non-returnable requests are requests whose 'Preferred Media' field is set to "Copy" or "Electronic" (for example, C-PRINTED, E-MR). A returnable request is not checked against this parameter and can be registered without signing the copyright agreement (returnable

requests are requests whose 'Preferred Media' field is set to "Loan". for example, L-PRINTED).

If COPYRIGHT_MANDATORY is set to Y, it is impossible to place an ILL Copy or Electronic request without selecting the relevant check box. Once the transaction has been sent, the patron gets an ILL request confirmation (the HTML page www_f_eng/new-ill-req-conf-info) that contains the ILL request information and the Copyright Declaration text.

If COPYRIGHT_MANDATORY is set to N, the patron can sign the agreement, but it is not mandatory for returnable and non-returnable requests.

The UPDATE_COPYRIGHT_LETTER parameter indicates whether the database (Z410-COPYRIGHT-LETTER and its related Z411-AGREEMENT-SIGNED) will be updated as a result of signing or not signing the Web copyright Agreement. If UPDATE_COPYRIGHT_LETTER is set to Y, and the patron signs the copyright agreement, Z410-COPYRIGHT-LETTER is marked Y and all further related suppliers requests will have Copyright Signed field (Z411-AGREEMENT-SIGNED) set as Y.

If UPDATE_COPYRIGHT_LETTER is set to N, both Z410-COPYRIGHT-LETTER and Z411-AGREEMENT-SIGNED will not be updated (they will be left blank). Whether or not the copyright agreement has been signed, the database (Z410 and Z411) will not be affected.

14 The Borrowing Request's Format and Media

The ILL document (Z00 of the XXX40 doc) of a borrowing request includes an expand 592 field that indicate the bibliographic format of the request. For example, MN for monograph, SE for serial, MP for map, and so on.

The 592 field is created based on the ILL library table: ill_format_type (./xxx40/tab/ill_format_type) and expand program: `expand_doc_type`.

For each format (592 field of the ILL document), there is a list of possible media that can be assigned to it when creating a patron borrowing request (Z410-MEDIA). For example, L-PRINTED for loan of a book, E-MR for electronic copy of machine-readable, and so on.

The list of possible media per format is derived from the ILL library tab_ill_preferred_media table (./xxx40/tab/ tab_ill_preferred_media).

14.1 Building the 592 expand field (ill_format_type and expand_doc_type program)

The ILL library ill_format_type table, and the expand `expand_doc_type` program are used to build the 592 expand field of the ILL document (monograph, serial, and so on.)

The ILL system uses the content of the 592 field for determining ItemIdItemType (IllRequst ISO parameter) and for analyzing the possible media for each format. (see section 14.2 Defining media per format (tab_ill_preferred_media) on page 50).

The 592 field is an expand field created by the expand_doc_type program.

For this purpose, set the following line in the ILL library table: tab_expand.

(./xxx40./tab/tab_expand):

```
ILL-FORMAT expand_doc_type                ill_format_type
```

The above line of the expand_doc_type must be also defined in the ./tab/tab_expand table of the libraries from which the ILL request is created. For example:

- If the ILL request is based on a title found in the BIB XXX01 library, then expand_doc_type should be defined in ./xxx01/tab/tab_expand.
- If the ILL request is based on a title found in an external database (using Z39.50 gate), then expand_doc_type should be defined in ./extnn/tab/tab_expand.

Use the expand_doc_z410 and expand_doc_z416 to enable a word search anywhere in the record, for example, seraching by a request number. In this case, you may need to set up tab11_word.

To find a lending request by item call number by expanding the item details to the ILL bibliographic record of the lending request, use expand_doc_ill_z30. This routine enables you to index the call number of the item as a word index and to search for it in the Lending Bar. It creates (in the ILL bibliographic record) a field with code "Z30-2" if the item is type "Issue" and code "Z30-1" if the item is a different type (such as expand_doc_bib_z30). Add this expand for Word Indexing, after the line with expand_doc_z416. Make sure that the expand_doc_bib_z30, tab00.lng, tab11_word and, tab_word_breaking tables all define the call number.

This program and the ill_format_type table create a 592 field that is based on the content of various fields present in the record from which the ILL request is created (for example, a combination of the LDR and the 008 field).

Codes for MN (monograph) and SE (serials) must be defined in the table.

Multiple 592 fields can be created in the ILL document (in the event of multiple matching of record content with table definitions); the system takes the first occurrence of 592; therefore, the order of lines in the table is significant.

Example of ill_format_type setup:

```
592   MC Microform              007   F00-01     EQUAL      h
592   VD Videorecording         007   F00-01     EQUAL      v
592   MO Motion picture         007   F00-01     EQUAL      m
592   SR Sound recording        007   F00-01     EQUAL      s
592   MU Music                  LDR   F06-01     EQUAL      [c,d,i,j]
592   SC Score                  LDR   F06-01     EQUAL      [c,d]
592   SC Score                  006   F00-01     EQUAL      [c,d]
592   SR Sound recording        LDR   F06-01     EQUAL      [i,j]
592   SR Sound recording        006   F00-01     EQUAL      [i,j]
592   MP Map                    LDR   F06-01     EQUAL      [e,f]
592   MP Map                    006   F00-01     EQUAL      [e,f]
592   CM Computer file          LDR   F06-01     EQUAL      m
592   CM Computer file          006   F00-01     EQUAL      m
592   ER Electronic resource    007   F00-01     EQUAL      c
592   ER Electronic resource    85641            EXIST
592   ER Electronic resource    85640            EXIST
592   ER Electronic resource    856## 2          EQUAL      http
```

```
592   FI Film                    LDR   F06-01     EQUAL      g
                                 008   F33-01     EQUAL      m
592   FI Film                    006   F00-01     EQUAL      g
                                 006   F16-01     EQUAL      m
592   FI Film                    007   F00-01     EQUAL      m
592   FI Film                    007   F00-01     EQUAL      m
592   VD Video Recording         007   F00-01     EQUAL      v
592   VD Video Recording         LDR   F06-01     EQUAL      g
                                 008   F33-01     EQUAL      v
592   VD Video Recording         006   F00-01     EQUAL      g
                                 006   F16-01     EQUAL      v
592   GV Govt.Doc.               LDR   F06-01     EQUAL      [a,m]
                                 008   F28-01     N-EQUAL    ^
592   GV Govt.Doc.               006   F00-01     EQUAL      a
                                 006   F11-01     N-EQUAL    ^
592   SE Serial                  LDR   F07-01     EQUAL      s
592   SE Serial                  006   F00-01     EQUAL      s
592   MN Monograph               LDR   F06-01     EQUAL      a
                                 LDR   F07-01     EQUAL      m
592   CF Conference              LDR   F06-01     EQUAL      a
                                 008   F29-01     EQUAL      1
592   CF Conference              111##              EXIST
592   CF Conference              711##              EXIST
592   CF Conference              811##              EXIST
592   CF Conference              006   F00-01     EQUAL      a
                                 006   F12-01     EQUAL      1
592   UK Unknown format          245##              EXIST
```

Note

The structure of ill_format_type is similar to the XXX01 BIB library table: tab_type_config.eng.

Logically, the ill_format_type table must be present in the library from which a patron chooses a record for creating an ILL request (XXX01 or EXT01, EXT02, and so on). Therefore, there should be a path_convert line to redirect ill_format_type from the XXX01 BIB library and EXTnn libraries to the XXX40 ILL library.

Blank ILL requests (requests that are not created based on a found title but by filling in all BIB info), are assigned with a default 592 as follows: a book and non-serials request gets MN (monograph); a serials request gets SE (serial)

14.2 Defining media per format (tab_ill_preferred_media)

To define possible media that can be requested for each format (592 field of the ILL document), configure the ILL library tab_ill_preferred_media table (./xxx40/tab/tab_ill_preferred_media).

The list of preferred media is displayed in ILL borrowing request application forms; in the 'Preferred Media' field of the Web OPAC ILL request forms; and in the 'Requested Media' field of the GUI ILL borrowing request forms. The initiator of the ILL request (a patron via Web OPAC or a staff user via the GUI ILL) must select one of the offered media appearing in the drop-down list.

The possible media codes are as follows: PRINTED, COPY, MICROFORM, VIDEO, AUDIO, MR, OTHER. Each of the above media must be assigned with prefix of "L-" (for Loan), "C- " (for Copy) or" E-" (for electronic).

Examples of media code and possible usage:

- L-PRINTED (loan of a monograph)
- L-VIDEO (loan of a DVD)
- E-MR (electronic copy of a PC file)
- C-COPY (photocopy of serials)
- E-COPY (electronic copy of serials)

The textual translation of the media codes appears in the ILL library pc_tab_exp_field.eng table: (./xxx40/tab/pc_tab_exp_field.eng). The relevant entry is ILL-REQUEST-MEDIA.

Col. 2 through 16 of tab_ill_preferred media lists the possible media codes for the format 592 code defined in Col. 1.

The codes that are used in Col. 2 through 16 must be one of the following: PRINTED, COPY, MICROFORM, VIDEO, AUDIO, MR, OTHER (use these only with the prefix L for loan, C for copy and E for electronic).

Example of tab_ill_preferred_media setup

```
MN L-PRINTED       C-PRINTED       L-COPY          C-COPY       E-COPY
SE C-COPY          C-MR            E-COPY          L-MICROFORM
## L-PRINTED       C-PRINTED       L-COPY          C-COPY       E-COPY
   L-AUDIO         C-AUDIO         L-VIDEO         C-VIDEO      E-COPY
   L-OTHER         C-OTHER
```

15 Loading a Default Roster of Suppliers

In order to *Locate* potential suppliers for a borrowing request you must set the Roster of Potential Suppliers for each ILL Unit via the GUI ILL, Administration tab, Roster of Suppliers (see section 2.8 Creating a roster of potential suppliers (for borrowing requests) on page 25).

There is also an option to load a default 'roster' per ILL unit by setting up tab_z701 and running the GUI ILL 'Load Potential Suppliers Default' (ill-52) service. The load will be performed only if the 'roster' of the ILL Unit is empty.

15.1 tab_z701

tab_z701 can be *Locate*d either in the ILL Library tab directory (./xxx40/tab/tab_z701) OR in the alephe tab directory (./alephe/tab/tab_z701).

In the event that there is no tab_z701 file in the ILL Library path (or it is empty), the system will use the wide-default table of the alephe path..

tab_z701 structure

Col. 1 – Request Media: defines the media type code of the request, for example, L-PRINTED. It can be one of the following: PRINTED, COPY, MICROFORM,

VIDEO, AUDIO, MR, OTHER. Use these only with a prefix of L-, C- or E. In addition, you can define the ANYMEDIA and NOMEDIA codes (with no prefix).

Col 2 – Level: defines the group within the media.

Col.3 – Randomize Y/N flag: "N" indicates that the suppliers within the level are addressed in the order in which they are listed; "Y" indicates that the suppliers are addressed in random order. Define this parameter only for the first line per media + level.

Col.4 - Supplier *Locate* Base: The base that will be used for a BIB search within the supplier database. You may leave this field empty and a default *Locate* Base will be taken from the partner record of the supplier.

Col.5 – Partner Code: define the partner code that will be used as a supplier.

Col. 6 – Average Supply days: define the average number of days it takes a supplier to fill in the request. You may leave this field empty and a default Average Supply Days parameter will be taken from the partner record of the supplier.

Col.7 – Expiry Days: defines the number of days after which the request expires (relevant only for ISO ILL requests).

Col. 8 – Return Delay: defines the number of days it takes for you to return the requested items to the supplier. You can leave this field empty and a default Return Delay parameter will be taken from the partner record of the supplier.

Sample of tab_z701:

```
! 1                          2   3      4                              5
6  7    8
!!!!!!!!!!!!!!!!!!!-!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-
!!!-!!!-!!!
L-PRINTED            01 Y                    MED
L-PRINTED            01                      LAW

L-PRINTED            02 N                    OCLC
L-PRINTED            02                      GRT
L-PRINTED            02                      LAW_LT
L-PRINTED            99                      LAW_LT

C-COPY               01 Y                    OCLC
C-COPY               01                      BLDSC

C-COPY               02 N                    ILL_HYL
C-COPY               02                      ILL_MED
C-COPY               02                      LAW_LT
C-COPY               99                      LAW_LT

E-MR                 01 Y                    OCLC
E-MR                 01                      BLDSC
```

E-MR	02 N	ILL_HYL
E-MR	02	ILL_MED
E-MR	02	LAW_LT
E-MR	99	LAW_LT
ANYMEDIA	01 Y	ILL_LAW
ANYMEDIA	01	ILL_MED
ANYMEDIA	01	TLCD

15.2 Loading the default roster of potential suppliers

In order to load into your ILL Unit, the default 'roster of potential suppliers' that is defined in tab_z701, you must run the Service: "Load Potential Supplier Default" (ill-52).

Log into the GUI ILL as a ILL Unit staff user, click Services, then General, then Load Potential Supplier Default (ill-52).

When the load is run, and the system detects that your ILL unit code is defined as a partner code in tab_z701 (Col.5), the specific row will not be loaded, so that your ILL unit will not be assigned as a supplier of itself.

The load will be performed only if the 'roster' of the ILL Unit is empty; otherwise, you will be prompted by a block message: "Service can not be run. Potential supplier list already exist".

16 Handling BLDSC Borrowing Requests

In order to enable the dispatch of ILL borrowing requests to the BLDSC (British Library Document Supply Centre (http://www.bl.uk/services/document/dsc.html)) in a TX format; you must carry out the following steps:

- Setting up the BLDSC as a partner and define it as a potential supplier (pre-requisites)
- Setting up Customer ID and keyword list (pre-requisites)
- Assigning the BLDSC to a NEW borrowing request and setting up keywords and Customer ID.
- Sending the request to the BLDSC
- (Optional) - Resend request to the BLDSC for RENEW, CHASER or any other purpose.

16.1 Setting up the BLDSC as an External Partner and a Potential Supplier

As with all other partners that are used as external suppliers, the BLDSC must be set as an external partner (via the GUI ILL, Administration tab, Partners List).

In the Partner record's Communication tab, in the 'Protocol Type field, select 'British Library Format' (BL). There is no need to fill in the 'Server Address' and 'Institution Symbol' fields. All other fields have no special relevancy to the BLDSC setup.

In order to enable the usage of the BLDSC as a supplier, you must set it in the 'Roster of Potential Suppliers' (via the GUI ILL, Administration tab, Potential Suppliers list).

For more information, see section 2.8 Creating a roster of potential suppliers (for borrowing requests) on page 25.

16.2 Setting up Customer ID and Keywords

Customer ID: In order to send a request to the BLDSC (a partner with BL in its 'Protocol Type' field), you must set a BLDSC customer ID for each ILL unit. To do so, in the GUI ILL, click the Administration tab, then the Customer ID node. This invokes the Customer ID List.

Make sure that you select the BLDSC in the Customer ID's supplier field.

To allow sending of the request vi e-mail (ARTeMail), set the BLDSC e-mail address in the 'Supplier Email ' field.

Keywords: Define keywords in the ILL Library table: pc_tab_exp_field (./xxx40/tab/pc_tab_exp_field.eng). Relevant entry: BL-KEYWORDS.

Partial example:

BL-KEYWORDS	L Copyright required	RZ
BL-KEYWORDS	L Worldwide search required	WWS
BL-KEYWORDS	L Photocopy only required	PHOTO
BL-KEYWORDS	L Loan only	LOAN
BL-KEYWORDS	L Microform not acceptable	MICNO

The BLDSC: ADD (Alternative Document Delivery) and SED (Secure Electronic Delivery) services are also supported via the Keywords mechanism. For more info, see section 17 The BLDSC Services: ADD and SED Services on page 59.

16.3 Assigning the BLDSC to a New Request

As in the case of all other suppliers, the BLDSC supplier can be assigned to a new request by one of the following ways: adding the BLDSC in manual mode (clicking the **Add Supplier** button) OR using the *Locate* function (**Locate** button or running UE_19).

- **Clicking the Add Supplier button from the Borrowing Request expand**. Upon selecting the BLDSC as a supplier (partner with BL in its 'Protocol Type' field), the keywords and the customer ID will be available for you to fill in. When the Add Supplier action is completed, the patron request status will become: Waiting for Process (WAP), the BL request to supplier will become: New-British Library (NEB). At this point the request is ready to be sent to the BLDSC (see next paragraph: Sending the Requests to the BLDSC).
- **Activating the Locate action either by**:
  - Using the '*Locate*' button from the Borrowing Requests list.

OR

- Using the 'Unmediated Locate' (the ILL library daemon: UE_19)

Upon assigning the BLDSC as a supplier (whether by the **Locate** button or by the UE_9), click the **Modify** button in the Borrowing Request expand to set the BL keywords and customer ID. When the **Send** button is clicked, the BL request becomes Ready for BL (NEB). At this point, the request is ready to be sent to the BLDSC (see section 16.4 Sending the new request to the BLDSC below).

Note

You can use the ILL library daemon, UE_19, for locating the BLSDC as a supplier but not for sending the request out to the BLDSC.

16.4 Sending the new request to the BLDSC

In order to send all 'Ready for BL' (NEB) requests to the BLDSC, run the ILL Send Requests to the British Library' (ill-75) borrowing service. This service searches for 'BL format' requests to suppliers whose status is NEB (Ready for BL) and enables you to send an ART (Automated Request Transmission) to the British Library (ARTTel or ARTEmail).

Upon running this service, the request status become: Sent to Supplier (SV) and the action is recorded in the Request Log as Request Sent or Re-Sent to the BL (log transaction number 82).

The templates that are used for this purpose reside in the ./xxx40/form_eng directory.

The convention for the file name is: ill-print-bldsc-<media-type>-<format>-00.

The templates for this function are located in the ILL library's (XXX40) form_<lng> directory with the text string "bldsc".

For each request media and format, there is a template for the ART message.

The request format can be, for example: SE (serial) MN (monograph) MR (machine-readable) or any other code.

The request format code is determined by ALEPH tag 592 subfield a (592$$a) of the ILL library document. In the event that the ILL document contains more than a single 592 field, the first appearance of 592 is considered as the valid one.

The request media type is determined by the 'Requested media' field of the patron's request. Requests for a 'loan' (also known as a 'returnable request') are represented by the media code "l". Requests for 'copy' and 'electronic (also known as non-returnable) are represented by the media code "c".

For example:

- For format SE (Serial) which is a loan request, the form ill-print-bldsc-l-se-00 is used.
- For format SE non-returnable request (copy or electronic), the form ill-print-bldsc-c-se-00 is used.
- For format CF (Conference) which is a loan request, the form ill-print-bldsc-l-cf-00 is used.

- For format CF non-returnable request (copy or electronic), the form ill-print-bldsc-c-cf-00 is used.
- For formats that has no matching forms, the forms ill-print-bldsc-l-other-00 and ill-print-bldsc-c-other-00, are used.
- Note that the 00 extension of the BLDSC templates is hard coded and cannot be modified by the user.

Note

Type "l" refers to all returnable requests (loan) and type "c" refers to all non-returnable requests (electronic and copy). There are no templates for "e" as electronic type is considered as "c" (non-returnable).

16.5 The BLDSC Template Format

The ART file that is sent to the BLDSC must be in the following structure.

==========

Your Customer Code	87-9999
Your ART Password	*******
4 blank lines	
1st request	
TX line, Message Keyword Code(s)	TX PHOTO MICNO
Bibliographic details	Canadian Veterinary Journal
(max. request length is 11 lines)	2003 VOL 35 PT 2 PP 111-
(max. line length is 40 characters)	Bovine neonatal ... Bryan, LA
Shelfmark, if known	3046.10000
4 blank lines	
2nd request	
TX line, Message Keyword Code(s)	TX LOAN
Bibliographic details	Immunology: a comparative approach
	Turner, RJ (ed)
	2003
ISBN quoted, if known	0471944009
4 blank lines	
End of file code	NNNN

Followed by a single return-key stroke

==========

Following are the ALEPH placeholder ($ signs) that are available for use in BLDSC TX forms templates:

- $001 - TX + request number (in template: `TX$00138)` OR $015 – TX + Supplier Request Number (in template: `TX$01538`)
- $002 - through $005 – title
- $007 - place of publication (MARC 21 tag 260 subfield a)
- $008 - publisher (260-b)
- $009 - year of publication (260-c)
- $010 - requested pages (ISS-p)
- $011 - call no.
- $012 – supplier reference number
- $014 - ISBN/ISSN
- $016 - BIB library
- $017 through $020 - author
- $021 - report no.
- $022 - edition (250)
- $023 – year (ISS-y)
- $024 - customer ID
- $025 - password
- $026 through $028 - article title (74002)
- $030 - keyword
- $031 through $033 - article author or additional author
- $035 - series (440)
- $036 - source (590)
- $040 thorough $044 – end use physical or electronic address (ADD and SED service)
- $051 – volume (ISS-v)
- $052 – issue (ISS-i)
- $053 – month (ISS-m)

Example of a template:

ill-print-bldsc-c-se-00

```
S1
## - EMAIL_ADDRESS $05000
```

```
## - PLAIN,L01
$02400
$02500
+
+
+
+
S2
TX$00138
- $03040
- $00240
- $00340
- $00440
- $02310     VOL $05110      $05210
- $01440
- $02640
- $02740
- $02840
- $03140
- $03240
- $00720               $00820
- $01140
- $03540
- $03640
+
+
+
+
S3
:DELIVER ABOVE ITEM TO:
- $04033
- $04133
- $04233
- $04333
- $04433
+
+
+
+
S4
```

NNNN

The first three digits after the $ sign comprise the system data placeholder and the last two digits indicate the maximum length.

For example:

```
- $00720      $00820
```

This line will print Place of publication ($007) for 20 characters and Year of publication ($008) for the next 20 characters.

Refer to this document (http://www.bl.uk/services/document/pdf_files/artguide.pdf) for more information about sending ART requests.

16.6 Resending requests to BLDSC

After the initial request is sent to the BLDSC, there is an option to resend the request. For example:

- After the initial request is sent to BLDSC you might want to send a chaser (claim) to the BL (BL Keyword: CHASER).
- After material has arrived from BLDSC, you might want to renew the loan periods (BL Keyword: RENEW)

To resend a request to BL, click the **Modify** button in the Borrowing Request expand, set the relevant keywords and click the **Resend BL** button.

This will cause the request to supplier to become: Ready for BL (NEB). The Patron request status will stay as is.

Note

The Resend BL button appears only for a BL request which already sent to the BLDSC.

The next step is to run the 'Send Requests to the British Library' (ill-75)' service. The batch service will resend the BLDSC request with the updated keywords.

When this service is run, the supplier request status will revert to its previous status (as before the Resend BL action) and the action is recorded in the Request Log as transaction 82: Request Sent or Re-Sent to the BL. Among other data, the log includes the sent keywords so that the user can know that RENEWAL/CHASER/Any other update has been sent to the BL.

17 The BLDSC Services: ADD and SED Services

ADD (Alternative Document Delivery) is a British Library address service that enables requested items to be sent to an address that is different from the registered customer address.

SED (Secure Electronic Delivery) is a British Library address service that enables requested items to be sent to the patron’s e-mail.

You can use the ADD service to request direct delivery of material to the patron's physical address (ADD) or to his e-mail address (ADD combined with SED).

Both services can be integrated with borrowing requests which are sent to the British Library by ARTel or ARTemail.

In order to support the above features, ensure that the following lines appears in the <*ill library*> table: pc_tab_exp_field.eng (./xxx40/tab/ pc_tab_exp_field.eng):

BL-KEYWORDS	L Alternative Document delivery	ADD
BL-KEYWORDS	L Secure Electronic Delivery	SED99
BL-KEYWORDS	L Secure Electronic Delivery 2 hour delivery	SED99*2*
BL-KEYWORDS	L Secure Electronic Delivery 24 hour delivery	SED99*24*

17.1 Activating the ADD Service

In the process of assigning a borrowing request to the BLDSC, the Keywords 1 and the Keywords 2 fields are available. If a staff user wants the requested item to be sent directly to the patron (and not to the library), he must select ADD as one of the keywords.

When the 'Send Requests to the British Library' (ill-75) service is run, each request which has ADD as one of its keywords will have the patron's address quoted below. For example:

```
CUSTID
PASSWD
+
+
+
+
TXAB12345 Y LOAN
Bleak House
Dickens, Charles
+
+
+
+
: DELIVER ABOVE ITEMS TO:
Mr P. Thrower
Brick Lane
Hayes
Middlesex
+
+
+
+
```

NNNN

Note

The ADD keyword is a British Library service but not a BL official keyword. The ADD will not be placed in the keyword section of the British Library ART transmission. It is an ALEPH internal keyword code whose purpose is to indicate that the patron's address should be displayed with the requested item in the ART transmission to the British Library. If the user wants the requested item to be sent

directly to the patron (and not to the library) he must select ADD as one of the keywords.

17.2 Activating the SED Service

In the process of assigning a borrowing request to the BLDSC, the Keywords 1 and the Keywords 2 fields are available. If a staff user wants the requested item to be sent to the library email account, he should select SED keyword (for example, SED99).

If a staff user wants to the requested item to be sent directly to the patron's e-mail address, he should select ADD and SED keywords.

When the 'Send Requests to the British Library' (ill-75) service is run, each request which has ADD and SED as its keywords will have the patron's e-mail address as quoted below.

For example:

```
CUSTID
PASSWD
+
+
+
+
TXAB12345 Y COPY SED99
Bleak House
Dickens, Charles
Chapter XI
+
+
+
+
: DELIVER ABOVE ITEMS TO:
throwep@cla.ac.uk

+
+
+
+
NNNN
```

18 Handling SLNP Borrowing and Lending Requests

In order to enable the dispatching and receiving of ILL borrowing and lending requests with your partner libraries by SLNP (for detailed instructions, see *ALEPH500 – SLNP Interface (ILL2)*), you must carry out the following steps (mandatory):

- Setting up the central ILL server as a partner
- Setting up your partner libraries to work with SLNP
- Setting up your items as available for ILL
- Configuring your local SLNP server and then starting it (util w / 3 / 7 / 5)

18.1 Setting up the Central ILL Server as an External Partner

Although the central ILL server does not actually supply the requested material, it is temporarily used as an external supplier in the borrowing requests, and must therefore be set as an external partner (via the GUI ILL, Administration tab, Partners List).

In the Partner record's Communication tab, in the 'Protocol Type field, select 'SLNP Format' (SLNP). There is no need to fill in the 'Server Address' and 'Institution Symbol' fields. All other fields have no special relevancy to the SLNP setup.

18.2 Setting up the Partner Libraries to Work with SLNP

In order to send and receive requests from partner libraries via SLNP, the libraries must be setup as follows:

1. In the Partner record's Communication tab, in the 'Protocol Type field, select 'SLNP Format' (SLNP). There is no need to fill in the 'Server Address' and 'Institution Symbol' fields. All other fields have no special relevancy to the SLNP setup.
2. Make sure the libraries exist as patron with the library code (Sigel) as patron ID or barcode.
3. Define the sigel of the library in alephe/tab/tab_sigel

18.3 Setting Up Items as Available for ILL

Use xxx50/tab/tab_ill_item_status to define, depending on item status and item process status, which items are available for ILL. You may differentiate between loan and photocopying or allow both.

18.4 Configuring your local SLNP server and starting it

In contrast to all other suppliers, the ZFL supplier cannot be assigned to a new request manually or using the Locate function. The supplier is automatically assigned by the local SLNP server when creating a new borrowing request.

The configuration of the local SLNP server is done in xxx40/tab/tab_slnp.conf. Here, you must specify the host address and port of the central ILL server with which it will communicate (for a description of all parameters, see *ALEPH500 – SLNP Interface (ILL2)*).

The SLNP server may be started manually (UTIL W/3/7/5) or automatically (aleph_start.private: parameter "SLNP_STARTUP_LIBS").

19 APDU Exceptions

The APDU Exceptions list displays all ISO ILL APDU messages received by the ILL server that for various reasons could not be processed by the ILL server daemon (ue-17); therefore the APDU messages are not reflected in the ILL borrowing or lending requests.

Each ILL APDU message that is received by the library's ILL server becomes an ILLSV (ILL server) record. All ILLSV records are immediately picked up by the ue-17 daemon that consistently runs in the ILL Server library. Usually, most of the ILLSV records are successfully handled and the ILL database is updated accordingly. Only a small proportion of the ILLSV records will not be handled; in this event, the APDU Exceptions interface can be used as a monitor for the outstanding APDUs.

Staff users with the required permissions can access the APDU Exceptions list via the ILL GUI by clicking the Administration tab, then the APDU Exceptions List node. The APDU Exceptions interface allows staff users to manipulate the ILLSV records and to resend them to the ue-17 daemon for further handling.

Note

The 'APDU Exceptions' permission is related to the ILL server (ILLSV) library. The ILLSV library is the shared repository for all APDU messages received by the ILL server and it used in common to all ILL libraries and units of the ALEPH installation. It means that a staff user that accesses the APDU Exceptions list can see and update APDUs that do not necessarily belong to his ILL library/Unit; bear this in mind when assigning the APDU Exceptions privilege to staff users.

Technically, the system analyzes the ILLSV records that are created in the ILLSV library as a result of receiving of APDUs.

The APDU Exceptions List displays all records whose status field (tag 466$$a of the ILLSV record) indicates that they have not been processed.

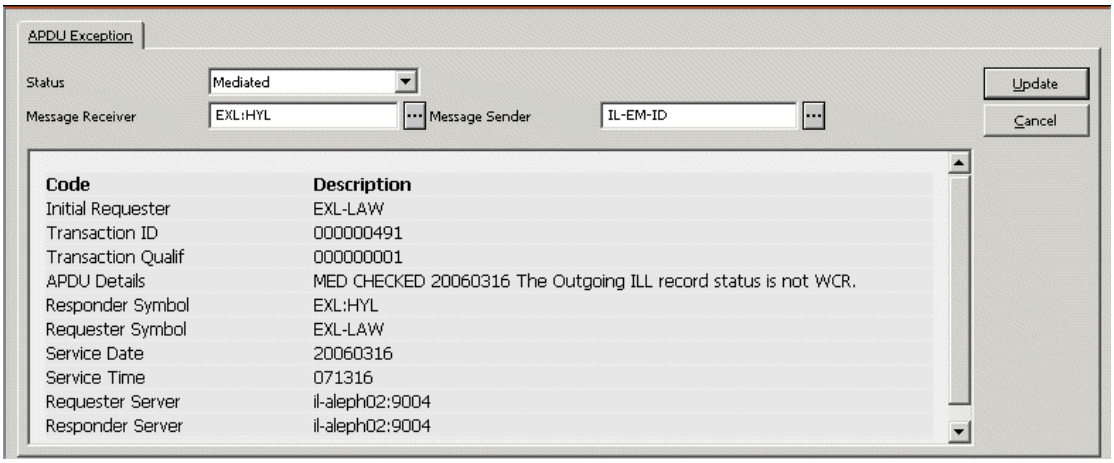
Statuses indicating that an APDU has not been handled are:

- Unrecognized (UNR) – An APDU that has been received from an unknown sender, that is, no ILL partner record (Z700) has been found at the ALEPH installation.
- Unsupported (UNS) – An APDU received by your server with a type that is not supported, so the daemon is not capable of processing the message. For example: APDU of the 'Forward-Notification' type.
- Mediated (MED) -The APDU contains data that cannot be processed or is missing crucial data. For example, an APDU of type RSMSG (Request Submission Message) contains an unfamiliar patron barcode.
- Failed (FLD) - ue-17 failed to process the APDU.

For each entry in the list, the following information is displayed:

Doc. No (ILLSV record number), Title, Author, APDU Type (IllRequest, Shipped, Received, CheckedIn, and so on), Service Date/Time.

The lower pane shows expanded details of the highlighted entry in the APDU Exception list and allows records to be updated.



You may want to update the ILLSV record for one of the following reasons:

- To change details that might cause the ue-17 daemon not to handle the APDU in order to resend the ILLSV record to the daemon.
- To dismiss an entry from the APDU Exception category.

Information Fields

Most of the fields are for information only with no option to update. The display of the ILLSV record field is according to the setup of the table: ./illsv/tab/edit_doc_999.lng.

The HTML page is taken from ./illsv/pc_display<lng>/apdu-info

Active Fields

The staff user can update the 'Status', 'Message Receiver' and 'Message Responder' fields.

Updating Statuses

The Status field shows the current status of the ILLSV record. This status can be one of the statuses that are assigned to an ILLSV record as a result of a failure in the process: Unsupported, Unrecognized, Mediated or Failed.

The drop-down list allows you to update the status to one of the following statuses: Close or New.

- Close – The system automatically moves an ILLSV record status to 'close' upon completing handling the APDU message. If a record is changed manually to 'Close' (by using this option), the system will not refer to this record again. The daemon will not attempt to handle the record again and the entry of the ILLSV record is dismissed from the APDU Exceptions category.
- New – The system automatically assigns 'New' status to each recently-created ILLSV record and it communicates to the daemon that ILLSV record has not been handled yet and it should be processed in order to integrate the APDU within the ILL data. If a record is changed manually to 'New' (by using this option), the system will execute another attempt to analyze the record and to integrate it within the ILL data. The user should change the status to New only if action has been taken in order to correct the problem that caused the failure message. After changing the record status to New, the ue-17 daemon picks the

'New' record and attempt again to handle it. This also dismisses the record from the APDU Exceptions category.

Changing the 'Message Receiver' and 'Message Sender' fields
These fields enable you to change the code (institution symbol) of the receiver/sender in the ILLSV record.

The drop-down menu lists all the ILL partners (supplier and requesters) that are set in your ALEPH installation.

You can change the message receiver and message sender code (library symbol), for example, when an APDU IllRequest is received from an unknown requester that is not registered as an ILL partner (or wrongly written in the partner files). An ILLSV record is created as an 'Unrecognized'. If the user decides to handle the request, he can register the partner and then update the Message Sender using the Change Message Sender field. In order to re-send the IllRequest to the daemon, the status must be changed from Unrecognized to New.

20 ILL Temporary Items

An ILL temporary item is automatically created by the system when the staff user registers the arrival of ILL material which should be returned to the supplier (the "Receive-Returnable" action in the ILL Borrowing workflow).

The ILL temporary item is linked to an ADM document record that also created upon the Receive-Returnable action is completed.

The ADM record (XXX50 doc) is linked to the ILL document (XXX40 doc) that was created when the borrowing request has been initiated.

The item record is discarded upon sending the material back to the supplier (the Return action).

When receiving material, it is possible to define the number of temporary items that are created.

The circulation note is mandatory for multiple items. It is recommended to enter a description such as "Item 1 out of 12 related items".

Supplying the item to the patron:

The item can be immediately loaned to the patron (upon completing the Receive action) OR wait in the 'hold shelf' for the patron (see Setting up handling of ILL Items (tab27) on page 69).

In the event of an immediate loan, the request status is updated to LOP-Loaned to Patron.

In the event of a "hold shelf", the request status is updated to LON-Loaned to Library. The request status will be updated to LOP - Loaned to Patron only when the patron approaches to pick his ILL material. The loan to patron is performed via the Circulation GUI. The due date is determined according to the Request Return Date (specified by the supplier) minus the days that the library requires to return the material back to the supplier. The Return Delay parameter is displayed in the Potential Supplier tab or the Partners Info tab.

In the case of multiple items, the status of the ILL request is updated only after all the temporary items are loaned.

20.1 Circulation Checks

The tab_check_circ ADM library table (./xxx50/tab/tab_check_circ) is consulted in the following ILL related events:

Loan checked-in material to patron - The relevant check procedure (Col.1 in tab_check_circ) is LOAN. The LOAN procedure check is performed whether the 'loan to patron' action is performed from the GUI ILL (immediate loan upon receiving the item from the supplier) or via the Circulation GUI (loan action). If the loan action is performed via the Circulation GUI, you can define the check_circ_12_a routine (in Col. 2 of tab_check_circ). The 12_a routine checks if there is a match between the original ILL patron and the patron trying to loan the ILL item.

Renew loan for an ILL material – The check_circ_13_a routine (Col. 2 of tab_check_circ) works with the RENEW and RENEW-GBL (Col.1 of tab_check_circ) actions. The 13_a routine checks if the renewed items are ILL items; if so, the renew action is enabled only for authorized ILL staff user (staff user with ILL Unit same as the ILL unit that attached to the ILL item).The reason for this check: ILL items renewal require supplier's confirmation and the renew action should not be allowed for patrons or non ILL staff. It is recommended to block the override option for this check. To do so, set error code 803 in ./xxx50/tab/ check_circ_override as follows:

```
0803 99 ILL item (Barcode $1) can be renewed only by authorized ILL
staff (renew requires supplier's confirmation).
```

Note

If staff is authorized according to the 13_a check, or if the check is not defined, then the renewal of ILL items sends a request for renewal to the lending library.

Only for the SLNP protocol is the loan extended without receiving permission from the lending library.

Returning the item:

The item can be returned by the patron via the ILL GUI or the Circulation GUI.

If the Return action is performed in the ILL GUI (Borrowing-Return button), the patron's loan is discharged and the ILL request is marked with one of the following statuses (depends on the supplier's protocol type):

- CLS - Closed (for non-ISO workflow. This is a final ILL request status)

OR

- RT-Returned by library (for ISO workflow. The ISO 'Checked-In' message is expected to be received from the supplier).

If the Return action in performed in the Circulation GUI, a PC file configuration of the circ.ini controls the system's behavior. The relevant flag in circ.ini is:

```
[Ill]
ReturnIll=
```

The value can be Y (Yes) or N (No). Default: N

If the flag is set to ReturnIll=Y, then the patron's loan is discharged and the ILL request is marked with one of the following statuses (depends on the supplier's protocol type):

- CLS - Closed (for non-ISO workflow. This is a final ILL request status)

OR

- RT-Returned by library (for ISO workflow. The ISO 'Checked-In' message is expected to be received from the supplier). Note that when there is a special case of transit sublibrary, the only type would be RTP.

If the flag is set to ReturnIll=N, then the patron's loan is discharged and the ILL request is updated with the status RTP - Returned by Patron (regardless of the supplier's protocol type). At this stage, the final return to the supplier should be performed via the ILL GUI, Borrowing - Return action.

The due hour of the loaned items is set to the closing time of the library and is taken from tab17 of the ADM library.

20.2 The ILL Item Values

The ILL temporary item gets the following values:

Item Barcode

The staff user can manually type in the barcode in the Receive-Returnable form. For easy follow-up, we suggest you use a request number with an "ILL-" prefix. For example: item barcode ILL-471 for ILL request 471. There is a system check for barcode uniqueness.

If no input has been typed in, the system will automatically generate a unique barcode (same as if a new item is created via the GUI ILL-Catalog-Items).

For automatically creating an ILL item barcode with the prefix ILL- defined the following line in tab_checksum (./xxx50/tab/tab_checksum):

```
CSCR-ILL-BARCODE                checksum_create_ill_barcode
```

When the barcode is generated for multiple items, the items' barcode is created with the suffix "-n". The barcode of the first item is not be changed.

For example, request number 561 has three related items. The following barcodes are generated:

- First item: ILL-561
- Second item: ILL-561-2
- Third item: ILL-561-3

Item Call Number

The item call number is hardcoded as follows:

Item Call Number Type – Depends on the value that is configured in the call_number_type variable of the pc_server_defaults file. The default value is 7.

Item Call Number – The ILL Request Number

Item Sublibrary

The item sublibrary is set as defined in the ILL-ITEM-SUB-LIBRARY flag of the ADM library's tab100 table (./xxx50/tab/tab100).

Possible values: 1 or 2.

1 = (default) use patron's home library if populated, otherwise use pickup location

2 = use pickup location if it is a valid "item" sublibrary, otherwise use patron's home library.

Item Process Status

The item process status is hard coded: IL.

Make sure that the "IL" code is defined in Col. 3 of the ADM library's tab15.eng table (./xxx50/tab/tab15.eng).

For example, for an ILL item with a sublibrary code of LAW, set the following line in tab15.<lng>:

```
LAW   ## IL L ILL Temp. Item                Y Y N N N N N Y Y 00
N A
```

Item Collection

The item collection is hard coded: ILL.

Make sure that the "ILL" code is defined in Col. 1 of the ADM library's tab40.eng table (./xxx50/tab/tab40.eng).

For example:

```
ILL   LAW   L ILL Material
```

Item status

The staff user selects the item status from a drop-down list within the Receive-Returnable Form.

The drop-down list is derived from the ILL library's pc_tab_exp_field.eng table (./xxx40/tab/pc_tab_exp_field.eng). Relevant entry: ILL-ITEM-STATUS.

```
ILL-ITEM-STATUS      L 14 Pickup Circulating            14
ILL-ITEM-STATUS      L 72 Home Loan                     72
ILL-ITEM-STATUS      L 73 Box Loan                      73
ILL-ITEM-STATUS      L 74 Reading Room Loan             74
ILL-ITEM-STATUS      L 75 Pickup - Loan to Pickup SL    75
ILL-ITEM-STATUS      L 76 Non-Circulating               76
```

The first entry in the list is used as the default item status.

Note

Each item status code defined in the ILL-ITEM-STATUS list must have a matching line in the ADM library's tab15.eng table (./xxx50/tab/tab15.eng).

The item status code is a trigger to the action that should be taken by the system when the Receive-Returnable action is completed. The possible actions are: item is assigned to the "hold shelf" or immediately loaned. The possible actions are set in the ADM library's tab27 table (./xxx50/tab/tab27).

Circulation note

As entered.

Note

When creating multiple items for a single request, all the created items are created in one temporary ADM record. All the created items have the same information in all fields, except for the Barcode and Note fields.

20.3 Setting up handling of ILL Items (tab27)

For ILL items, the setup of the ADM library's tab27 table (./xxx50/tab/tab27) defines whether the item is assigned to the hold shelf or immediately loaned.

When material is received from an ILL supplier, the manner in which ALEPH handles the item depends upon the target option as defined in Col. 5 of tab27.

Target Option Code	**Procedure**
01 - Hold Shelf	A slip (for insertion in the material before it goes on the hold shelf) and a collection notice (for the borrower) are printed, but the item is not automatically loaned. When the borrower collects the item, the loan is performed. The due date is calculated according to the request's 'expected return date'. (The "Expected Return Date" minus the "Return Delay" which is taken from the partner record of the supplier. If the Return Delay is zero, the value of setenv `ill_return_for_user` in `pc_server_defaults` is used instead).
02 - Home Delivery	A slip (for insertion in the material before it is posted to the borrower) is printed and the item is automatically loaned to the patron. The due date is calculated according to the request's 'Expected Return Date' and Return Delay parameter.
03 - Mailbox Delivery	A slip (for insertion in the material before it is dispatched to the mailbox) and a notice (for the patron) are printed. The item is automatically loaned to the patron.
04 - Reading Room	A slip (for insertion in the material before it is dispatched to the reading room) and a notice (for the patron) are printed. The item is automatically loaned to the pickup sublibrary The due date is calculated according to the request's 'Expected Return Date' and Return Delay parameter. From the pickup sublibrary, the item can be loaned to the patron.. The due date is calculated according to the request's 'Expected Return Date' and Return Delay parameter

Target Option Code	Procedure
05 – ILL	A slip (for insertion in the material before it goes on the hold shelf) and a collection notice (for the patron) are printed. The item is automatically loaned to the pickup sublibrary (there must be a 'patron' record with a Patron ID that corresponds to the pickup sublibrary code). The due date is calculated according to the request's 'Expected Return Date' and Return Delay parameter.

In the demo library setup of USM50, the following item statuses are dedicated to ILL items and the following explanation uses these statuses:

14 - ILL Pickup - matches target option 01

72 - ILL Home Loan - matches target option 02

73 - ILL Box Loan - matches target option 03

74 - Reading Room - matches target option 04

75 - Loan to pick up or ILL sublib. - matches target option 05

76 - ILL Non-circ.

All the above statuses are defined in ./usm50/tab/tab15.eng as well.

The matching setup in tab27 is as follows:

```
!   2   3    4   5
!-!!!!!-!!-!!!!!-!!
H ##### 73 ##### 03
H ##### 72 ##### 02
H ##### 75 ##### 05
H ##### 14 ##### 01
H ##### 74 ##### 04
```

20.4 Multi Items Related Configuration

- In the following printouts, use z30-barcode information instead of z411-item-barcode:

./<ILL_lib>/form_lng/ill-patron-letter-4.xsl
./<ILL_lib>/form_lng/ill-patron-letter-5.xsl
./<ILL Lib>\form_lng\ill-arrival-slip-new.xsl
./<ILL Lib>\form_lng\ill-patron-letter-a.xsl

- Add z30-barcode to ./<ill library>/pc_display_lng/bor-request-expand

- GUI \Circ\Tab\lng\Message.dat:
  Add:

MultiIll Multi Item ILL O I There are %d additional items that should also be loaned.

- GUI .\Ill\Tab\Lng\Message.dat:

  Add:
ReturnMultiItems ALEPH ILL OC Q This request has %d related items. Continue?

- GUI .\ill\tab\<lng>\window.dat:

  Add:
  MaterialType.Radio.ReturnableMulti Returnable &Multi

  ReceiveReturn.Tab.Screen0 Request
  ReceiveReturn.Tab.Screen1 Items
  ReceiveReturn.Btn.Add &Add
  ReceiveReturn.Btn.Delete &Delete
  ReceiveReturn.Btn.Modify &Modify
  ReceiveReturn.MultiItems.EmptyBC System Generated Barcode

  BulkReceiveRet.Tab.Screen1 Items
  BulkReceiveRet.Btn.Add &Add
  BulkReceiveRet.Btn.Delete &Delete
  BulkReceiveRet.Btn.Modify &Modify
  BulkReceiveRet.MultiItems.EmptyBC System Generated Barcode
  ReceiveMultiInput.Title Multiple Items
  ReceiveMultiInput.Btn.Add &Add
  ReceiveMultiInput.Static.Barcode Item Barcode
  ReceiveMultiInput.Static.Note Circulation Note

- GUI .\alephcom\tab\<lng>\tab_col.dat:

  Add:
  ILL_MULTI_ITEMS L Barcode 01 030 01 C01 barcode
  ILL_MULTI_ITEMS L Circulation Note 02 070 01 C01 note

21 Lending Library - Supplying an Item

When the lending library supplies an ILL material for 'loan' purposes (Ship-Returnable action), the regular circulation policy is consulted.

Loan due date
Staff users can fill in the 'Expected Return Date' in the Ship-Returnable GUI ILL Form. If no date has been typed in, the ADM library's tab16 table (./xxx50/tab/tab16) is consulted in order to determine the loan due date. The supplied item status, process status and local patron definitions of the ILL requester are taking into account.

Circulation Checks

The ADM library's tab_check_circ table, (./xxx50/tab/tab_check_circ), is consulted in order to check if the ILL requester can loan material from your ILL lending services. The relevant check procedure (Col.1 in tab_check_circ) is LOAN. You can define various checks over the item and/or the patron record of the ILL requester.

22 ILL Request Log

The ILL request log is displayed in the GUI ILL functional mode of Borrowing and Lending requests. The Request Log keeps track of various request-related activities that have occurred.

The ILL library's tab46.eng table, (./xxx40/tab/tab46.eng), determines which transactions are recorded in the ILL Request log.

22.1 tab46.eng Structure

Column	Purpose	Example	Description
1	Transaction Number	Hard coded	This number is used by the system and should not be changed.
2	Active/Non active log	Y	Setting this flag to Y - Yes or N - No determines whether this type of transaction should be logged. It is recommended to set all flags to Y.
3	Manual Log/Non-manual log	N	Setting this flag to Y – Yes or N – No determines whether this type of transaction can be manually added by a staff user (not automatically system-driven).
4	Alpha and Functional Description	ILL Request Created	Alpha code (L for Latin) and Textual description of the transaction. This text appears in the Description Column of the Request Log details.

Important

It is recommended to set Y in Col. 2 for all transactions, regardless of whether your library uses ISO or non-ISO flow.

It is recommended to set Y in Col.3 only for transactions 00, 48, and 49.

The text that is displayed in the "System Notes" column of the Request Log, is driven from `$alephe_root/error_lng/ill_logger2`.

22.2 Request Log Display – Sort Routines

The sort order and routine of the ILL Request Log are set in `./alephe/pc_server_defaults`.

The relevant lines are:

```
ill_user_z415_sort_routine
ill_user_z415_sort_order
```

ill_user_z415_sort_routine defines the sort routines:

00 – by the open date and hour of the transaction

01 – by sequence no. of the log

02 – by the log action date plus the log sequence. The latest action date will always be sorted on the top. If the log action date is zero, the system will use instead log open date, open hour and open seconds combined as a string

ill_user_z415_sort_order defines the sort order:

A – ascending

D – descending

Example:

```
setenv ill_user_z415_sort_routine     00
setenv ill_user_z415_sort_order       D
```

23 Request Numbers

23.1 Borrowing - Request Number

Each borrowing request that is created automatically receives a main number: Request Number. The request number is automatically derived from the last-doc-number system counter in the UTIL G/2 counter of the ILL library. This number can be displayed in the GUI-ILL-Borrowing-List of Requests. There is an option to search a request by its request number using the GUI-ILL-Borrowing Search bar: select the Request Number option from the drop-down menu and type in the relevant value.

23.2 Borrowing – Supplier Request Number

There is an option to have an additional request number for each request to supplier: Request to Supplier Number. This number is created upon generating request to supplier and it is automatically derived from the last-bor-supp-no system counter in the UTIL G/2 counter of the ILL library. This number can be displayed in the GUI-ILL-Borrowing-forth tab: Suppliers. There is an option to search a request by its request number using the GUI-ILL-Borrowing Search bar: select Additional Request Number from the drop-down menu and type in the relevant value.

23.3 Lending – Request Number

There is an option to assign a unique request number for each request to a supplier in the "Supplier Request Number" field. This number is created for each new request to a supplier and it is automatically generated from the last-bor-supp-no system counter

in the ILL library UTIL G / 2 counter. This number can be displayed in the Suppliers and Request Info tabs of the GUI-ILL-Borrowing module. It is possible to search for a request by its supplier request number using the GUI-ILL-Borrowing Search bar by selecting **Supplier Request Number** from the Search bar drop-down list and typing in the relevant value.

23.4 Lending – Requester's Request Number

For lending requests that are created by ISO-ILL request, the requester original request number is recorded in the Requester's Reference Number field. This number is as defined at the requester side and populated in the ISO lending request. This number can be displayed in the GUI-ILL-Lending-tab 1: Request Info (view only). There is an option to search a request by its requester reference number using the GUI-ILL-Lending Search bar: select Requester Reference No. option from the drop-down menu and type in the relevant value.

24 ILL Configuration Tables

The ILL configuration tables are located in the ILL library's tab directory (./xxx40/tab) and in the alephe tab directory (./alephe/tab). One table (used for SLNP only) is located in the ADM library's tab directory (./xxx50/tab). The ILLSV tab directory is relevant only for ISO ILL usage.

24.1 ILL Library Tables (./xxx40/tab)

edit_paragraph.eng

Defines the formatting of the bibliographic information that is displayed in the new ILL request forms (Web OPAC form and GUI ILL form). The relevant section (Col.1 of edit_paragraph) is : 002.

Note: When submitting a new ILL request based on an existing title, the format is defined in ./xxx01/tab/edit_paragraph.lng.

form_description

This table defines the subject line for an Email when a print form is sent by Email.

ill_bor_charge

Borrowing ILL charges; determines the default amounts for charging patrons when ILL material is received from a supplier. Works in conjunction with transaction 0015 of ./xxx50/tab/tab18.eng.

ill_format_type

Defines the 592 format fields of an ILL document. Works in conjunction with expand_doc_type (See section 14 The Borrowing Request's Format and Media on page 48).

ill_formats_map.eng

For ISO ILL : This table translates the ISO IllRequest.ItemId.ItemType code (1, 2, 3) for an incoming ILL request to a textual description:

The tag is mapped to field 592 of the ILLSV record and the ILL document.

The table should read:

```
!-!!!!!!!!!!!!!!!!!!!!!!!!!
1 Monograph
2 Serial
3 Other
```

ill_lend_charge

Lending ILL charges; determines the default amounts for charging the ILL requester when ILL material is supplied. Works in conjunction with transaction 0016 of ./xxx50/tab/tab18.eng

ill_unit_group.<eng>

Defines the ILL unit group for the US copyright mode (see section 34 Copyright – US Mode on page 100).

pc_tab_exp_field.eng

Definition of GUI drop-down menus and textual description for system codes. See section 26 pc_tab_exp_field.eng on page 86.

tab00.eng

Defines field codes and names for the ILL documents indexes (WRD, ACC, and IND).

The WIT and WIU indexes must be defined. Equivalent values must be defined in ./xxx40/tab/tab11_word.

Sample:

```
H WIT   W-016      00       00       Borrowing/Lending
H WIU   W-017      00       00       ILL unit
```

In addition, to allow indexing of ILL History records, set a line for WHS index, as in the example below:

```
H WHS   W-041      00       00       ILL History
```

tab01.eng

Defines list of MARC record tags for the ILL document.

tab11_word

Defines word indexes for the ILL Document.

Among other indexes' codes, the Word indexes, WIT and WIU, must be defined for the ILL document.

Sample:

```
TYP##                                          00     WIT
```

```
UNT##                                                   00      WIU
```

In addition, to allow the ILL History procedure, set a Word index for ILLHS, as in the below example:

```
ILLHS                                                   03      WHS
```

tab46.eng

Defines transaction types for the ILL request log. See section 22 ILL Request Log on page 72 and the table's header.

tab47.eng

Determines the default copyright policy for the patron's Copyright Agreement and Copyright Block. See section 13 Patron's Copyright Agreement for Borrowing Requests on page 46.

tab50

Defines automatic processing for ILL Borrowing requests. Works in conjunction with the ILL library daemon: UE/19. See section 5 Setting up Unmediated Mode for ILL Units on page 35.

tab100

tab100 is the central configuration table for system-level, server-level and library-level variables. See section 25 tab100 - Related Entries for ILL on page 82.

tab_expand

List of ILL document expand programs.

Make sure that the following lines are defined in ./xxx40/tab/tab_expand:

```
 !  1              2                   3
 !!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
  U39-DOC    expand_doc_z410
  U39-DOC    expand_doc_z416
U39-DOC    expand_doc_bib_copyright_2
U39-DOC    expand_doc_bib_copyright_3


  FULL-ILL   expand_doc_z410
  FULL-ILL   expand_doc_z416
FULL-ILL   expand_doc_bib_copyright_2
FULL-ILL   expand_doc_bib_copyright_3
  FULL-ILL   expand_doc_extract


SEND-ILL   expand_doc_bib_copyright_2
  SEND-ILL   expand_doc_bib_copyright_3
```

WORD expand_doc_z410

WORD expand_doc_z416

WORD expand_doc_ill_z30

* ILL-FORMAT expand_doc_type ill_format_type

* The line of the expand_doc_type must be also defined in the ./tab/tab_expand table of the libraries from which the ILL request is created. For example:

If the ILL request is based on a title found in the BIB XXX01 library, then expand_doc_type should be defined in ./xxx01/tab/tab_expand.

If the ILL request is based on a title found in an external database (using Z39.50 gate), then expand_doc_type should be defined in ./extnn/tab/tab_expand

tab_filing

Defines filing procedures for ILL document tags. This will ensure that the ILL doc has clean BIB information. This is important for the 'Locate' function.

For ISO ILL: the filing will assure that the ItemId tags of the ISO message will not include extra characters such as {[% and so on. In this way, the lending library receives 'clean' BIB information. The filing procedures are important for both: ISO ILL and non-ISO ILL.

tab_ill_bib_info

For ISO ILL: this table defines the ILL document tags that build the ISO IllRequestItemId XML tags. Note that default values for building the ItemId tag are set in the code (for example, field tag 260##$$a is directed to ItemIdPlaceOfPublication). See table's header.

tab_ill_bib_key

For ISO ILL: This table enables you to set the IllRequest ISO tags SystemNo (SYSNO) and NationalBibliographNo (NBN) according to tags defined in the ILL document.

Cols. 1 through 3 define the well-known number, its code and type (for example, LC 01 SYSNO). Cols. 4 through 6 define the tag, subfield and filing procedure. Cols. 7 through 9 define the database tag, subfield and filing procedures.

The whole set is used by the system for building the SYSNO or the NBN.

Col. 10 through 15 is the second priority set definition In the event that the first set (col. 4 through 9) has no matching values in the ILL document.

All fields defined in this table will be used when building the ILLSV document and will be included in the lending library's ILL document according to the definition of the corresponding tab_ill_code_keys table.

Example:

```
!1      2       3     4   5 6    7   8 9   10  11 12  13   14 15
!!-!!!!!!!!!!-!!!!!-!!!!!-!-!!-!!!!!-!-!!-!!!!!-!-!!-!!!!!-!-!!
01 LC          SYSNO 010   a
06 RLIN        NBN   035## a
09 ALEPH_SYS   SYSNO SID   c     SID   c
```

tab_ill_preferred_media

Defines possible media that can be requested for each format - 592 field of the ILL document (See section 14 The Borrowing Request's Format and Media on page 48)

tab_locate

Defines the Locate string query that is used in the ILL borrowing, ILL lending Locate, and Locally Own check functions. You may define a tiered-Locate search. See sections 3 ILL Locate Function on page 29, and 6 Setting up "Locally-owned" checks on page 39. See also the tab_locate table's header.

tab_slnp.conf

Configuration file for SLNP server.

tab_sublib_group.eng

This table allows you to define groups of sublibraries within the same ADM. The group code is listed in the GUI ILL's Partner record, Lending ILL Unit tab: Circ. Sublibraries. In a case where an ILL Unit represents more than 20 sublibraries, you must use a group code instead of individual sublibraries' codes. See also the tab_sublib_group.eng table's header.

tab_z701

Defines the default for the 'Roster of Potential Suppliers'. Works in conjunction with the Load Potential Supplier Default (ill-52) service. For a system-wide default, set the table in ./alephe/tab instead of ./xxx40/tab.

24.2 ADM Library Tables (./xxx50/tab)

tab_ill_item_status

Defines whether an item is available for ILL or not (used for SLNP only).

24.3 The alephe tables (./alephe/tab)

tab_ill_held_media_map.lng

For ISO ILL: The table translates the ISO IllRequest.ItemId.HeldMediumType value (1, 3,4,5,6,7) for an incoming ILL request to a textual description:

The tag is mapped to field 905 of the ILLSV record and the ILL document.

Example of ill_held_media_map.eng setup:

```
!        2
!-!!!!!!!!!!!!!!!!!!!!!!!!!
1 Printed
3 Microform
4 Film-or-Video
5 Audio
6 Machine-Readable
7 Other
```

library_relation

Defines the relationship between libraries by library type (see section 2.1 Setting up the ILL Library on page 12).

tab_base.eng

Defines logical bases that are used for the various Locate functions: borrowing Locate, Lending Locate and locally own check. See sections 3 ILL Locate Function on page 29, and 6 Setting up "Locally-owned" checks on page 39.

```
WID                                  WID-Main  Library          USM01
USM01 Y wsl=wid

HYL                                    HYL-East  Asian  Lib     USM01
USM01 Y wsl=hyl

LAW                    LAW-Law School       USM01              USM01
Y wsl=law
```

In addition, define lines for:

the ILL XXX40 Library

the XXX40 Borrowing database (XXX40-B)

the XXX40 Lending database (XXX40-L)

Example:

```
USM40                                          USMARC   ILL   (USM40)
USM40 N

USM40-B                                        USMARC   ILL   Borr-40
USM40 N wit=BORROW

USM40-L                                        USMARC   ILL   Lend-40
USM40 N wit=LEND
```

tab_ill_change_status_in

Defines the options to change lending request status. See section 9 Change Request Status on page 41.

tab_ill_change_status_out

Defines the options to change borrowing request status. See section 9 Change Request Status on page 41.

tab_ill_partner_conf

Only for ISO ILL. This table defines user name and password for ILL suppliers that demand those values (like OCLC). This tables replaces the Web ILL table: alephe/tab/ tab_supplier_ext (see table's header).

tab_ill_status_borrow

For the GUI ILL display of the Borrowing-Summary mode tree: defines the grouping for the 'Statuses' node. The textual description of the group codes and statuses is driven from ./xxx40/tab/pc_tab_exp_field.eng. Relevant entry: BOR-REQ-STATUS

tab_ill_status_lend

For the GUI ILL display of the Lending-Summary mode tree: define the grouping for the 'Statuses' node. The textual description of the group codes and statuses is driven from ./xxx40/tab/pc_tab_exp_field.eng. Relevant entry: LEND-REQ-STATUS.

tab_ill_tree_bor_msg

For the GUI ILL display of the Borrowing-Summary mode tree: defines the grouping for the 'Messages' node. The textual description of the messages codes is driven from ./xxx40/tab/pc_tab_exp_field.eng. Relevant entry: BOR-REQ-MSG

tab_ill_tree_lend_msg

For the GUI ILL display of the Lending-Summary mode tree: define the grouping for the 'Messages' node. The textual description of the messages codes is driven from ./xxx40/tab/pc_tab_exp_field.eng. Relevant entry: LEND-REQ-MSG.

tab_sigel

Concordance for library "sigel" and sub-library code resp. ILL Unit code (used by SLNP only).

tab_ill_supp_close_days

Define the non-working days of the system’s ILL suppliers, which may be either internal ILL units or external suppliers.

24.4 Aleph_start - ILL Related Entries

ill_borrower_status

This switch is used in ISO mode, when a lending request is sent to an external partner, and there is no patron record in the system. The system creates a patron record automatically and assigns it the status which is specified in the switch.

24.5 The ILLSV tables (./illsv/tab)

The illsv/tab directory is relevant only for a library that uses ISO ILL.

tab00.eng

The STS must be defined as an index code in tab00.eng. The following must be set:

```
H   STS   IND     11 00     0000       Process Status
```

tab11_ind

The STS index must be defined in tab11_ind for building the index for the Process Status field of the ILLSV records (field 466).

The following must be set:

```
466##    STS   ab
```

tab_ill_code_description

This table maps some of the APDU message's XML tags to numeric tags. When ue_17 process the ISO ILL messages, the programs use some of these numeric tags. Accordingly, you can add but not delete or override lines from this table.

Partial example:

```
46501 *.TransactionId.TransactionGroupQualifier
46502 *.TransactionId.TransactionQualifier
471   *.ResponderId.PersonOrInstitutionSymbol.Institution
472   *.RequesterId.PersonOrInstitutionSymbol.Institution
478   *.RequesterId.NameOfPersonOrInstitution.Institution
479   *.ResponderId.NameOfPersonOrInstitution.Institution
474   *.ServiceDateTime.ThisService.Date
475   *.ServiceDateTime.ThisService.Time
476   *.ServiceDateTime.OriginalService.Date
477   *.ServiceDateTime.OriginalService.Time
74002 IllRequest.ItemId.TitleOfArticle
70012 IllRequest.ItemId.AuthorOfArticle
245   IllRequest.ItemId.Title
100   IllRequest.ItemId.Author
592   IllRequest.ItemId.ItemType
```

tab_ill_codes_key

For ISO ILL: This table builds the relevant tags for SYSNO and NBN in the lending library ILL document. The SYSNO and NBN are derived from the IllRequest message and are used to set the defined tag (for example, 010).

Cols. 1 through 3 define the well-known number, its code and type (for example, LC 01 SYSNO). Cols. 4 through 7 define the ILL document tag and the subfields that will contain the NBN or SYSNO information, the number, database ID and database type.

Example:

```
!1      2       3     4   5 6 7
!!-!!!!!!!!!!-!!!!!-!!!!!-!-!-!
01 LC          SYSNO 010   a b c
06 RLIN        NBN   035## a b c
09 ALEPH_SID   SYSNO SID## a b c
```

tab_tag_text

This table defines the default LDR field for the ILLSV records. The following must be set:

```
LDR IL ^^^^^nam^^22^^^^^^a^4500
```

25 tab100 - Related Entries for ILL

`tab100` is the central configuration table for system-level, server-level and library-level variables. A few lines of the table are shown below:

ADM-ILL-PREFIX

Possible values: Y (Default) or N.

This flag directs the system for the method in which the patrons for the ILL partners are detected.

For multi ADM environment that has defined in $alephe_tab/tab100 USER-SHARING = "N"; set ADM-ILL-PREFIX to "Y".

For single ADM environment that has defined in $alephe_tab/tab100 USER-SHARING = "N"; set ADM-ILL-PREFIX to "N"

If your system is set to USER-SHARING = "Y", ADM-ILL-PREFIX flag is not consulted by the system.

For a multi ADM environment that uses non-shared-users, the patron record for the ILL partners must be defined with the library's ADM code as part of the patron ID.

For example:

If the ADM library ABC50 is defined as non-shared-users (TAB100-USER-SHARING=N), the patron ID of the ILL partners records must be defined with a prefix of ABC50.

For partner code MEDIL, define a patron record with ID: ABC50-MEDIL (not MEDIL).The maximum the partner’s code can be 6 is characters long.

Setting ADM-ILL-PREFIX =Y will point the system to look for a patron ID record with a ADM library code set as a prefix.

COPYRIGHT-MODE

Define in alephe/tab/tab100

Values: 0, 1, 2. Default: 0

Defines which national copyright mode in the ILL Borrowing request is tracked.

0 – No definition (for Europe)

1 – US copyright mode

2 – Australian copyright mode

DOC-TYPE-ILL

Define in ./xxx40/tab/tab100

Values: Y and N. Default: N

Y- indicates that the library is of type "ILL"

ILL-CHECK-DUP

For a system-wide default, define in alephe/tab/tab100. For a library default, define in ./xxx40/tab/tab100

Values: 0, 1, 2. Default: 0

For a Web OPAC ILL request, define the routine for checking duplicate active requests that has been submitted by the same patron.

0 - no duplication check

1 - Reject duplicated requests: a duplication check is performed and in the event that a 'match' active request is located for the same patron, the request is blocked and the patron is prompted by a 'Duplicate Request' block message.

2 - Accept duplicate requests: a duplication check is performed and in the event that a 'match' active request is located for the same patron, an ILL request with status: 'New Staff Review' is created.

ILL-ITEM-SUB-LIBRARY

Define in the ADM Library: ./xxx50/tab/tab100

Values: 1, 2. Default: 1

Define the Item sublibrary that is assigned to an ILL temporary item (an item that is created when returnable ILL material is checked in.)

1 - use patron home library if populated, otherwise use pickup location.

2 - pickup location if it is a valid "item" sublibrary, otherwise use the patron’s home library.

MULTI-ILL-UNIT-SELECT

Should be defined in ./xxx40/tab/tab100.

This variable controls the option of assigning an alternate ILL unit for a borrowing request that is registered via the Web OPAC:

Values: Y and N. Default: N

N = The ILL unit of the patron is used as default (default).

Y = A list will be displayed including all ILL units that belong to the patron's ADM.

P = The patron will be given a pickup locations list that is a joint list of all of the pickup locations of all the ILL units that belong to the patron's ADM.

When submitting the request, the system will direct the ILL request to the ILL unit that supplies the selected pickup location to the patron.

The pickup location list that is displayed in the ILL request form represents the possible pickup locations of the selected ILL Unit (not the patron original ILL Unit but the selected ILL Unit). Upon submitting the form, the ILL request is assigned to the selected ILL Unit.

Note: The “Y” and “P” options are available only for Patrons entering a request in the Web OPAC. Staff users submitting a request via the GUI ILL will not have the option of selecting an alternate ILL unit; therefore, the patron's assigned unit will be used (even in cases where MULTI-ILL-UNIT-SELECT=Y).

PAT-CANCEL-ILL

This flag is relevant only for ISO borrowing requests (requests which were sent to a supplier using ISO ILL Request message).

For a system-wide default, define it in alephe/tab/tab100. For a library default, define it in ./xxx40/tab/tab100.

Values: Y and N. Default: N

Defines the patron’s ability to cancel ILL borrowing requests (via the Web OPAC’s Library Card - ILL Active and Total Requests) which have already been sent to the supplier.

Y – 'Sent to supplier' request can be canceled by patron

N- 'Sent to supplier' request cannot be canceled by patron (the **Cancel** button will not be available. The action can be done only by a staff user via the GUI ILL)

PAT-RENEW-ILL

For a system-wide default, define this setting in alephe/tab/tab100. For a library default, define this setting in ./xxx40/tab/tab100

Values Y and N. Default: N

This variable defines whether the patron is allowed to renew an ILL loaned item outside of the ILL module.

Y – ILL loaned material can be renewed outside of the ILL module.

N- ILL loaned material cannot be renewed outside of the ILL module.

PATRON-LOCAL-OWN

For a system-wide default, define this setting in alephe/tab/tab100. For a library default, define this setting in ./xxx40/tab/tab100.

Values: 0 and 1. Default: 0

Defines the system’s behavior in the event that a patron submits a Web OPAC ILL request that has been marked as 'Locally owned'.

0 - locally-owned requests are blocked.

1 – locally-owned requests continue to be processed and they get the status 'Locally Owned' (LOW).

SHIP-TO-PICKUP

This flag is relevant only for ILL units that are defined as ISO partners (their partner's protocol type is ISO)

Should be defined in ./xxx40/tab/tab100.

This variable controls the option of selecting an alternate delivery address:

Values: Y and N. Default: N

N = The delivery address quoted in the APDU is the ILL unit address.

Y = The delivery address quoted in the APDU is the pickup location address. The ship to address is displayed in the Lending request details and can be printed out in the lending slip.

SHIP-TO-PICKUP-ADDR

This flag is relevant when SHIP-TO-PICKUP is set to Y.

Should be defined in ./xxx40/tab/tab100.

This variable controls the option of filling in the address information in the APDU tags:

Values: Y and N. Default: N

Y- The address from tab_sub_library_address.lng marked as of type "X" populates the <DeliveryAddress> ILL tag as follows:

Line 1 from the address -> name-of-person-or-institution (150 chars)

Line 2 from the address -> street-and-number (60 chars)

Line 3 from the address -> city (19 chars)

Line 4 from the address -> country (20 chars)

Line 5 from the address -> postal code

Line 6 from the address -> Telecom-service-address (Fax)

Line 7 from the address -> Telecom-service-address (Email)

Line 8 from the address -> Telecom-service-address (Ariel (FTP))

N = Address populates the <ExtendedDeliveryAddress> ILL tag.

SUPPRESS-PATRON-INFO

Values: Y and N. Default: N

This variable controls the option to suppress the patron information from being sent via the ILL Request APDU.

Y = Patron info will not be included in the IllRequest APDU except for patron's email (on the outgoing APDU) when the requested media is electronic.

N = Patron info will be populated in the relevant ILLRequest ISO tags.

E = Patron information will not be included in the IllRequest APDU. The ILL Unit's shipping e-mail address is sent (not the patron's email). If there is no shipping e-mail, the ILL Unit's general e-mail is used.

26 pc_tab_exp_field.eng

The pc_tab_exp_field.eng table of the ILL library (./xxx40/tab/pc_tab_exp_field.eng) determines some of the values of the GUI ILL drop-down menus. It also enables textual description for the system codes.

Structure of pc_tab_exp_field.eng

Column	Purpose	Example	Description
1	Internal Command	BUDGET-TYPE	Internal Reference for the menu code
2	Alpha	L	
3	Description	Regular	A free text description of the entry, as it will appear in the drop-down list.
4	Code	REG	The code to which your selection will be mapped by the system.

Example:

```
ILL-PARTNER-STATUS       L Foreign                                FO
ILL-PARTNER-STATUS       L Domestic                               DO
ILL-PARTNER-STATUS       L Active                                 AC
```

Internal commands for the menu codes

The following is a list of menu codes (Col. 1 of pc_tab_exp_field.<lng) that you may want to update by adding or deleting lines.

Internal Command	GUI ILL window	Notes
BORROWING-SEARCH	Borrowing-search bar	You can add BIB search values for the ILL library document. For example: for WTI word index, add line with code BIB-WTI. Make sure that the WTI word index is supported in /xxx40/tab/tab11_word and that indexes have been run for the new word index.
LENDING-SEARCH	Lending-search bar	You can add BIB search values for the ILL library document. For example, for the WTI word index, add a line with the code BIB-WTI. Make sure that the WTI word index is supported in ./xxx40/tab/tab11_word and indexes have been run for the new word index.
ILL-BOR-PRINT	Borrowing-List of Requests-Print button	You may want to add new templates that can be sent out to the patron. To do so, create a new XSL file in the ./xxx40/form_eng directory with the file name ill-patron-letter-<x>; x stands for the suffix letter that is

Internal Command	GUI ILL window	Notes
		used for the new template. The new templates's suffix should be used as the code for the pc_tab_exp-field menu (Col.4).
ILL-ITEM-STATUS	Borrowing-Receive Returnable Form: Item status	List of item statuses for the ILL temporarily items. All codes must be supported in ./xxx50/tab/tab15. First entry will be used as a default.
ILL-*LOCATE*-LIBRARY	Administration-Partners-tab 2 Communication-Locate Library field	The ADM library code (for example, KDS50, KSR50) for partners with Locate Type ALEPH
ILL-LOCATE-BASE	Partners and Potential Suppliers Expand Pane.	Using this section is optional and may be used to set a short list of Locate Bases. If this section is not populated, the system will list all of the tab_base.lng bases.
BL-KEYWORDS	BLDSC Keywords	Update this list with the relevant BLDSC keywords
LEND-ITEM-SORT-TYPE	Lending-Request expand-tab 4: Items-Sort	The sort codes must be defined in ITEM section of ./xxx50/tab/tab_z30_sort
ILL-PARTNER-STATUS	Administration-Partners-tab 1: Partner Info-Status field	You can add additional partner status for information. Note that the code NA (Not active) has a functional effect: The partner can not be used as a supplier.
PAGE-SIZE	Receive-Non-Returnable Form-Type of Page field	

27 Lending Request - Setting up Captions for Volume and Issue

It is possible to configure the prefixes of the volume and issue on the lending request form.

The following lines can be configured in `$alephe_root/error_lng/global_ill`:

```
1030 0000 L iss.
1031 0000 L vol.
```

28 Print Templates

There are two types of print templates:

- GUI templates which are produced by various flow actions, for example, a lending request pickup slip.
- ILL Services - batch and report templates.

28.1 GUI ILL Templates

The names of the ILL template files are as follows (where nn is a number identifying the version):

GUI Templates

Template File Name	**Purpose**	**print.ini function name**
Ill-arrival-slip-new	Borrowing – slip with the request details, to be placed in the item	ArrivalLetter
Ill-patron-letter-i	Borrowing request info. slip (**Print** button)	PatronLetter
Ill-patron-letter-5	Borrowing –letter to patron- recall (**Print** button)	PatronLetter
Ill-patron-letter-0	Borrowing – letter to patron- 'more info.' (**Print** button)	PatronLetter
Ill-patron-letter-3	Borrowing – letter to patron- copyright letter (**Print** button)	PatronLetter
Ill-patron-letter-c	Borrowing – letter to patron- cancellation (**Print** button)	PatronLetter
Ill-patron-letter-2	Borrowing – letter to patron- 'title is owned by library' notification (**Print** button)	PatronLetter
Ill-patron-letter-a	Borrowing – letter to patron- Request arrival notification.	ArrivalLetter
Ill-patron-letter-4	Borrowing – letter to patron- Overdue (**Print** button)	PatronLetter

Template File Name	Purpose	print.ini function name
ill-patron-letter-X (X stands for any one character that can be set by the system librarian)	Borrowing – letter to patron- (**Print** button) - can be set by system librarian. Add the "X" to ILL-BOR-PRINT menu of ./XXX40/tab/ pc_tab_exp_field.eng	PatronLetter
Ill-send-request-letter	Borrowing – Non-ISO request letter to supplier (**Send** button)	SendLetter
ill-return-letter-00	Borrowing – Non-ISO letter to supplier-Return Notification to Supplier	ReturnLetter
ill-bor-letter-c	Borrowing – Non-ISO letter to supplier claim (**Response** button)	BorClaim
ill-bor-letter-m	Borrowing – Non-ISO letter to supplier - general message (**Response** button)	BorMsg
ill-bor-letter-l	Borrowing – Non-ISO letter to supplier - ILL item is lost (**Response** button)	BorLost
ill-bor-letter-x	Borrowing – Non-ISO letter to supplier - cancellation of request (**Response** button)	BorCancel
ill-bor-letter-r	Borrowing – Non-ISO letter to supplier - renew ILL item (**Response** button)	BorRenew
ill-bor-letter-d	Borrowing – Non-ISO letter to supplier - ILL item is damaged (**Response** button)	BorDamage
ill-check-in-letter	Check-In - Notification to Requester - ILL loaned item has been returned and checked in.	CheckInLetter

Template File Name	Purpose	print.ini function name
Ill-bor-receive-letter-00	Borrowing – Non-ISO letter to supplier-Arrival Notification to Supplier	ArivalLetter – for returnable items CopyArrivalLetter – for non-returnable items. Optional. If it does not exist, ArrivalLetter is used.
ill-print-lend-request	Lending request info. slip (**Print** button)	LendRequest
ill-lend-item-list	Lending – items for supply - pickup slips (**Print** button of Items tab)	LendItemList
ill-lend-letter-m	Lending – Non-ISO letter to requester - general message (**Response** button)	LendMessage
ill-lend-letter-r	Lending – Non-ISO letter to requester - recall (**Response** button)	LendRecall
ill-lend-letter-a	Lending – Non-ISO letter to requester - renew answer (**Response** button)	LendRenewAnswer
ill-lend-letter-o	Lending – Non-ISO letter to requester - overdue (**Response** button)	LendOverdue
ill-answer-3	Lending – Non-ISO 'unfilled' letter to requester - (**Unfilled** button)	LendUnfilled
ill-answer-2	Lending – Non-ISO 'retry' letter to requester - (**Response** button)	LendRetry
ill-answer-6	Lending – Non-ISO 'hold placed' letter to requester - (**Response** button)	LendHold
ill-answer-5	Lending – Non-ISO 'will supply' letter to requester-(**Response** button)	LendSupply

Template File Name	Purpose	print.ini function name
ill-lend-supply-slip	Lending – 'supply' slip to be placed in the item (Ship action for 'loan' and 'copy')	ShipReturn ShipNonReturn
ill-lend-supply-letter	Lending – Non-ISO – electronic 'supply' notification to the requesting library (Ship action for 'electronic' / bulk and one-off)	ShipElectronic
ill-ship-electronic	Lending – ISO – electronic 'supply' e-mail notification to the patron of the borrowing side (Ship action for 'electronic' / bulk and one-off)	ElectronicShip

28.2 GUI ILL Services Templates

Template	Purpose
Borrowing Statistics report ill-out-statistic-report	Generated by ill-63
Borrowing – patrons' requests report ill-request-report	Generated by ill-74
Borrowing – static request report ill-static-status-req-report	Generated by ill-66
Borrowing – Outstanding Requests of former patrons Outstanding-req-of-former-bor	Generated by ill-67
Borrowing – claim letters (non-ISO) ill-claim-letter	Generated by ill-73
Borrowing – claim report ill-claim-report	Generated by ill-73
Borrowing – TX format requests to British Library ill-print-bldsc-<media-type>. For example, ill-print-bldsc-l-printed OR ill-print-bldsc-c-	Generated by ill-75

Template	Purpose
copy.	
Borrowing – hold shelf list ill-hold-shelf-list	Generated by ill-76
Lending – Statistics report ill-statistic-report	Generated by ill-64
Overdue-apdu-report-body AND overdue-apdu-report-total Lending – report and summary for overdue ISO messages sent to requesters	Generated by ill-68
Lending – request report ill-lend-requests-report	Generated by ill-65
Copyright – Payment report for the USA ILL copyright mode copyright-payment-report	Generated by ill-61
Copyright – Payment statistics report for the USA copyright mode copyright-statistic-report	Generated by ill-62
Archiving – patron link removal report patron-removal-ids	Generated by ill-70
Archiving – generate borrowing history ill-bor-history-report	Generated by ill-88
Archiving – generate lending history ill-len-history-report	Generated by ill-89

29 Defining ILL Column Headings (pc_tab_col.lng)

pc_tab_col.lng defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (USM01) table `pc_tab_col.lng`. For more information about `pc_tab_col.lng`, see the *ALEPH Staff User's Guide - General* chapter - *Using and Customizing Your GUI*.

The following is a list of the ILL windows which use pc_tab_col.lng for formatting data, and their identifiers (Column 1 in pc_tab_col.lng):

Unique Code	**ILL GUI Windows**
PC_ILL_PARTNER_LIST	Administration – Partners List
PC_PARTNERS_LIST	Borrowing – Request Expand –Suppliers tab
PC_ILL_B_LIST	Borrowing – List of Requests
PC_ILL_L_LIST	Lending – List of Requests
PC_ROSTER_LIST	Administration – Potential Suppliers List
PC_ILL_*LOCATE*_PREV	Borrowing – *Locate* action – Preview
PC_ILL_*LOCATE*_PREV_L	Lending – *Locate* action – Preview
PC_ILL_MLC	Lending – *Locate* action – Preview – Multi *Locate* preview
PC_ILL_LOG	Borrowing and Lending – Request Log
PC_MSG_LIST	Borrowing and Lending – Request expand – 5th tab: Messages
PC_ILL_APDU_LIST	Administration – APDU Exceptions Lost
PC_ILL_MSG_LIST	Borrowing and Lending – Summary Mode –Messages pane
PC_ILL_STATUS_GROUP	Borrowing and Lending – Summary mode: main pane
PC_ILL_ITEM_LIST	Lending – Request expand – Items tab
PC_ILL_CUSTOMER_ID	Administration – Customer ID

30 pc_server_defaults - ILL Related Entries

`pc_server_defaults` is an ALEPH configuration table located under `./alephe_root`. This configuration table defines the default values for the PC server. The relevant entries for GUI ILL in `pc_server_defaults` are:

call_number_type

This variable defines the call number type that is assigned to the temporary items that are created when an ILL borrowing request arrival is registered. If it is not defined, the default value is set to 7.

ill_expected_return_date

This variable defines the default value for the expected return date of the borrowing request.

If the Borrowing-Receive-Returnable window is open and there is no other information as to the date on which the supplier expects to get the item back, then the number of days defined in ill_expected_return_date is used to calculate the expected return date.

For example, if the setup is:

```
setenv ill_expected_return_date          028
```

The Expected Return Date field will display current date + 28 days.

ill_return_for_user

Possible value: up to a three-digit value can be entered here. This numeric value determines how many days it takes to return a loaned item to its ILL supplier. This value is activated only if the 'Return Delay' field in the partner's supplier record and in the Roster of the Supplier is zero. This parameter is taken into account when calculating the Due Date of an ILL loaned item.

For example, when the setup is:

```
setenv  ill_return_for_user   003
```

The Due Date of the ILL loaned Item will be three days before the date on which the ILL supplier expects to receive the item back (as defined in the 'Expected Return Date' field of the GUI ILL – Borrowing - Receive-Returnable Form.

ill_user_z415_sort_routine and ill_user_z415_sort_order

Define sort routines and order for the Request Log display in GUI ILL.

Sorting routines (ill_user_z415_sort_routine):

00 - by the open date and hour of the transaction

01 - by sequence no. of the log

02 - by the log action date plus the log sequence. The latest action date will always be sorted on the top. If the log action date is zero, the system will use instead: log open date, open hour and open seconds combined as a string

Sorting order (ill_user_z415_sort_order):

A – ascending

D - descending

Example:

```
setenv ill_user_z415_sort_routine     00
setenv ill_user_z415_sort_order       D
```

31 www_server.conf - ILL-related Entries

`www_server.conf` is an ALEPH configuration table located under `./alephe_root`. This configuration the defines the default values for the WWW server. The relevant entries for GUI ILL in `www_server.conf`:

ill_total_limit_period

This variable defines the date from which date the counting of the ILL borrowing request is started. The `ill_total_limit_period` format is MMDD (for example, 1020) is the 20th of October. The calculation is made from this variable date to the current date (the assumption is that the variable date is earlier than the current date). All GUI and Web OPAC ILL requests created between these dates are counted and compared to the patron's ILL total limit. The patron will be blocked if he has exceeded its limit within this period. In the Web OPAC library card, only ILL requests that are within the time limit frame are displayed.

To ignore this calculation, either delete this variable or enter 0000.

Example:

```
setenv ill_total_limit_period           1020
```

This means that the limit period starts on the 20th of October. The patron's ILL total limit (as defined in the patron global record) is set to 15. The patron submits an ILL request (from Web OPAC or GUI ILL) on the first of January. The system counts all requests from the 20th of October until the first of January (current date). If there are more than 20 requests, the patron will be blocked by the system: "Exceeded Limit on ILL Requests".

32 PC setup - ill.ini file

The `ill.ini` file defines settings for the ILL client. This chapter presents and explains the `ill.ini` settings.

Explanations of other setting definitions, relating to more than one module, can be found in the **.Ini Files* chapter of the *General* module of the *ALEPH Staff User's Guide*.

32.1 [General]

```
[General]
EnableRefer=Y
```

EnableRefer

Y - In the GUI ILL – Lending – Functional - List of Requests, the **Refer** button is displayed. A library that uses the 'refer' feature (forwarding the request from Lending to Borrowing) should use Y.

N - In the GUI ILL-Lending-Functional-List of Requests, the **Refer** button is not displayed. Library that do not use the 'refer' feature should set N.

DefaultBorrowing=Y

This option enables skipping the dialog box that pops up when the ILL Request button is clicked in the Show node of the Search tab. The Borrowing Request Form is automatically opened up and no option will be presented to create a lending request.

32.2 [BorrowStatusTree]

```
[BorrowStatusTree]
DisplayMessages=Y
ExpandMessages=Y
ExpandStatuses=Y
OpenByDefault=Y
```

The [BorrowStatusTree] section defines default setting for the GUI ILL – Borrowing - Summary Mode - Left pane branches, Statuses and Messages.

DisplayMessages
Y– The GUI ILL Messages left pane node is displayed. A library that uses ISO ILL should define Y.

N – The GUI ILL Messages left pane node is not displayed. A library that does not use the ISO ILL at all should define N.

ExpandMessages
This parameter is valid only if DisplayMessages is set to Y.

Y – The GUI ILL Messages node is opened in an 'expand mode'.

N – The GUI ILL Messages node is not opened in an 'expand mode'.

ExpandStatuses=Y
Y – The Statuses branch is expanded.

N – The statuses branch is not expanded.

OpenByDefault=Y
Y – The status tree is opened by default when the Borrowing tab is clicked

N (default) – The functional tree is opened by default when the Borrowing tab is clicked

32.3 [LendStatusTree]

```
[LendStatusTree]
DisplayMessages=Y
ExpandMessages=Y
ExpandStatuses=Y
OpenByDefault=Y
```

The [LendStatusTree] section defines default settings for the GUI ILL-Lending-Summary Mode-Left pane branches, Statuses and Messages.

DisplayMessages
Y- the Messages left pane node is displayed. A library that uses ISO ILL should define Y.

N - the Messages left pane node is not displayed. Library the do not use the ISO ILL at all, should define: N

ExpandMessages
This parameter is valid only if DisplayMessages is set to Y

Y - the Messages branch is expanded.

N - the Messages branch is not expanded.

ExpandStatuses=Y
Y - the Statuses branch is expanded.

N - the statuses branch is not expanded

32.4 [AdminTree]

The [AdminTree] section defines settings for GUI ILL – Administration.

```
[AdminTree]
APDUExceptions=Y
CustomerID=Y
```

APDUExceptions
Y – In GUI ILL - Administration tab: the APDU Exceptions node is displayed. A library that uses ISO ILL should define: Y.

N – In GUI ILL - Administration tab: the APDU Exceptions node is not displayed. A library that does not use ISO ILL at all should define: N

CustomerID
Y – The Customer ID node is displayed. A library that needs to define Customer ID for their suppliers (like BLDSC) should set Y.

N – the Customer ID node is not displayed. A library that does not need to define Customer ID should set N.

32.5 [GetDefaultValue]

[GetDefaultValue]

RequestedMedia=Y

The [GetDefaultValue] determine if the 'Requested Media' for a new borrowing request will be filled with the first relevant value or will appear blank.

RequestedMedia

Y=the 'Requested Media' will have as a default the relevant media as definred at the ILL library table: tab_ill_preferred_media (./xxx40/tab/tab_ill_preferred_media).

N=the 'Requested Media' will be empty and the staff user will have to pick the relevant value from the drop-down list (the list is driven from the ILL library table: tab_ill_preferred_media). The value 'N' is used as a default.

NeedByDate=[number of days from current date]

The NeedByDate parameter sets the number of days from the current date which will be used to set the default ‘need by date’ parameter for outgoing requests.

32.6 AddSupplierMethod]

[AddSupplierMethod]

PartnerCodeSearch=Y

PartnerCodeSearch
PartnerCodeSearch=Y - new search functionality is used.

PartnerCodeSearch=N - old functionality is used.

The default value of this parameter is "N".

33 Copyright – Australian Mode

This chapter is relevant only for Australian copyright law implementation for ILL.

To set your ALEPH installation for the ILL Australian copyright mode, define the following line in ./alephe_tab/tab100:

```
COPYRIGHT-MODE=2
```

This flag will cause each ISO ILL request sent out to suppliers to contain the relevant Australian Copyright Statement in the Copyright Compliance tag.

The text for the statement is driven from the partner record of the borrowing ILL unit. Relevant field: Copyright Statement. This field can be up to 300 characters.

For example:

You may set the text:

"Complies with Section 49 and 50 of the Australian Copyright Act 1968" in the 'Copyright Statement' field of the partner record of the ILL Unit. Every ISO ILL request that will be sent out to suppliers from this borrowing ILL unit will contain this text in its CopyrightCompliance tag.

Note that the USA copyright mechanism (Z418) is disabled for COPYRIGHT-MODE = 2.

34 Copyright – US Mode

The chapter is relevant only for USA copyright law implementation for ILL.

This chapter contains information about:

- The initial creation of a copyright file for each new borrowing request for a 'non-returnable'.
- ILL unit policy for handling various copyright statuses.
- Control of the required copyright mode for ILL (US copyright law or other countries mode).
- The update of the copyright file upon request fulfillment/un-fulfillment.
- Interface for copyright management.
- The ability to manually update/recalculate the copyright status.
- Record of all copyright activities in the request's transaction log.
- Processing the copyright payment report.
- Copyright statistics report.

34.1 Prerequisites

Following are the pre-requisites set up for the US copyright mode:

34.1.1 ILL US Copyright Mode

To set your ALEPH installation for the ILL US copyright mode, define the following line in ./alephe_tab/tab100:

COPYRIGHT-MODE=1

34.1.2 Setting ILL Units Groups for Copyrights Procedures

A few ILL units can be grouped to perform the copyrights check procedure as if they are one, as long the group is composed of units that belong to the same ILL library.

For this purpose, set the relevant values in the ILL library table: ill_unit_group.eng (./xxx40/tab/ill_unit_group.eng).

```
!1                      2
!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
U15AR                GRP01
U15ED                GRP01
U15HS                GRP02
U15LA                GRP02
```

Col 1: ILL Unit code (20 characters)

Col 2: Group code (20 characters)

Set a line for each ILL unit that is not independent for the purpose of copyright check policy. No need to set a line for an 'independent' ILL unit.

34.1.3 ILL Unit/Group Copyright Policy

Copyright policy can be determined for each ILL Unit or Group.

For this purpose, set the relevant values in the ILL library table: copyright_policy.eng (./xxx40/tab/ copyright_policy.eng)

```
!1                   2  3  4  5  6  7      8
!!!!!!!!!!!!!!!!!!!!-!!-!!-!!-!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!>
GRP01                05 04 02 03 02 01 CR Group - U15ED and U15AR
GRP02                05 04 03 03 02 01 CR Group - U15HS and U15LA
U15GN                05 03 03 03 03 03 CR for U15GN
```

Col 1: ILL unit/group code (20 characters).

Col.2: Number of calendar years for checking publication date (2 digits).

Col.3: Maximum number of copyright free copies allowed (2 digits).

Col.4: Action code for staff initiate request for copyright status 11 (2 digits).

Col.5: Action code for staff initiate request for copyright status 21 (2 digits).

Col.6: Action code for patron initiate request for copyright status 11 (2 digits).

Col.7: Action code for patron initiate request for copyright status 21 (2 digits).

Col.8: Group name (50 characters).

For each ILL unit/group (col. 1), set the number of calendar years that are relevant for copyright check (col. 2) and the maximum number of requests per title (within the range of years) that are free of copyright fee (col. 3).

In the event that nothing else is specified for a group or ILL unit, the default value for the 'years' parameter (col. 2) is 5 and the default value for the 'number of free request' parameter is 4.

Cols. 4 through 7 define the actions that should be taken by the system when a borrowing request that is subjected to a copyright check is initiated by a staff user or by a patron.

The possible values (actions) are:

01 – Auto reject for a patron and a warring message for a staff.

02 – Mediated: a borrowing request is created under the 'New-Staff review' category.

03 –Unmediated: a borrowing request is created under the 'New' category.

You may define actions (01, 02 or 03) for each of the following cases:

Col. 4 – set the action for a request that is initiated by a staff user when the system detects that the predicted copyright status of the request is "11 - Less than maximum number of filled and outstanding requests – probably free".

In the event the table is missing or no value is defined in this column, the default value is: 03 (unmediated).

Col.5 – set the action for a request that is initiated by a staff user when the system detects that the predicted copyright status of the request is "21 More than maximum number of filled and outstanding requests – probably not free".

In the event that the table is missing or no value is defined in this column, the default value is: 02 (mediated).

Col.6 – set the action for a request that is initiated by a patron when the system detects that the predicted copyright status of the request is "11 - Less than maximum number of filled and outstanding requests – probably free".

In the event that the table is missing or no value is defined in this column, the default value is: 03 (unmediated).

Col.7 – set the action for a request that is initiated by a patron when the system detects that the predicted copyright status of the request is "21 More than maximum number of filled and outstanding requests – probably not free".

In the event that the table is missing or no value is defined in this column, the default value is: 02 (mediated).

Col.8 – set the group name for the code that is defined in Col.1.

34.2 US Copyright Workflow

34.2.1 General

Each borrowing request for a non-returnable item has a copyright record that holds its title, ISSN/ISBN, copyright status, payment information, and so on.

The copyright status indicates if this request is subject to a copyright fee or not. The copyright status is represented in the IllRequest APDU message as US:CCG or US:CCL.

The match between requests that have the same title will be done based either on ISSN or on title (if ISSN fails to match).

The system automatically performs a check/update of the copyright status in the following events:

- When a request is created
- When a request is filled/un-filled

Staff manual intervention is supported in the following events:

- Manual change upon staff user decision
- Recalculate action – staff user manually activate the system's copyright calculation.

In the lifecycle of a request, its copyright status can be updated (automatically or manually) a few times with one of the following statuses.

Intermediate (Initial) Statuses**:**

11 – Less than the maximum number of filled and outstanding requests – probably free

21 – More than maximum number of filled and outstanding requests – probably not free

Final statuses:

00 - Free of copyrights royalty fee (US:CCL)

10 - Subject to copyrights fee but under max-number therefore free (US:CCG)

20 - Subject to copyrights fee and above max-number therefore not free (US:CCG)

50 – Canceled

51 – Unfilled

The list of statuses and their textual description is located in the ILL library's pc_tab_exp_field.eng table: (./xxx40/tab/ pc_tab_exp_field.eng. Relevant entry: COPYRIGHT-STATUS

34.2.2 Initial Copyright Check

The exact copyright status of a request is accurately determined only once a request has been filled/un-filled but the system must set an initial intermediate status once the request has been created.

The intermediate status allows managing the requests and supervising or blocking requests that are potentially subject to payment, depending on the probability that such a payment will be needed after all.

Upon creation of a new request for non-returnable, one of the following copyright statuses is assigned to the request:

00 - Free of copyrights royalty fee (the publication date is older than the pre-defined number of year for copyright check).

11 – Less than maximum number of filled and outstanding requests – Probably free

21 – More than maximum number of filled and outstanding requests – Probably not free

34.2.3 Additional Copyright Checks

After the initial copyright check, the system re-evaluates the copyrights at the stage of the request's fulfillment/un-fulfillment.

Filled Request:

Upon arrival of material from the supplier the system accurately determine the copyright status for each request that is subjected to a copyright check.

The status can be set to one of the following:

00 - Free of copyrights royalty fee

10 - Under the maximum number of free copies

20 - Above the maximum number of free copies

Un-Filled Request:

In the event that a request is eventually not filled, the system sets the copyright status to one of the following:

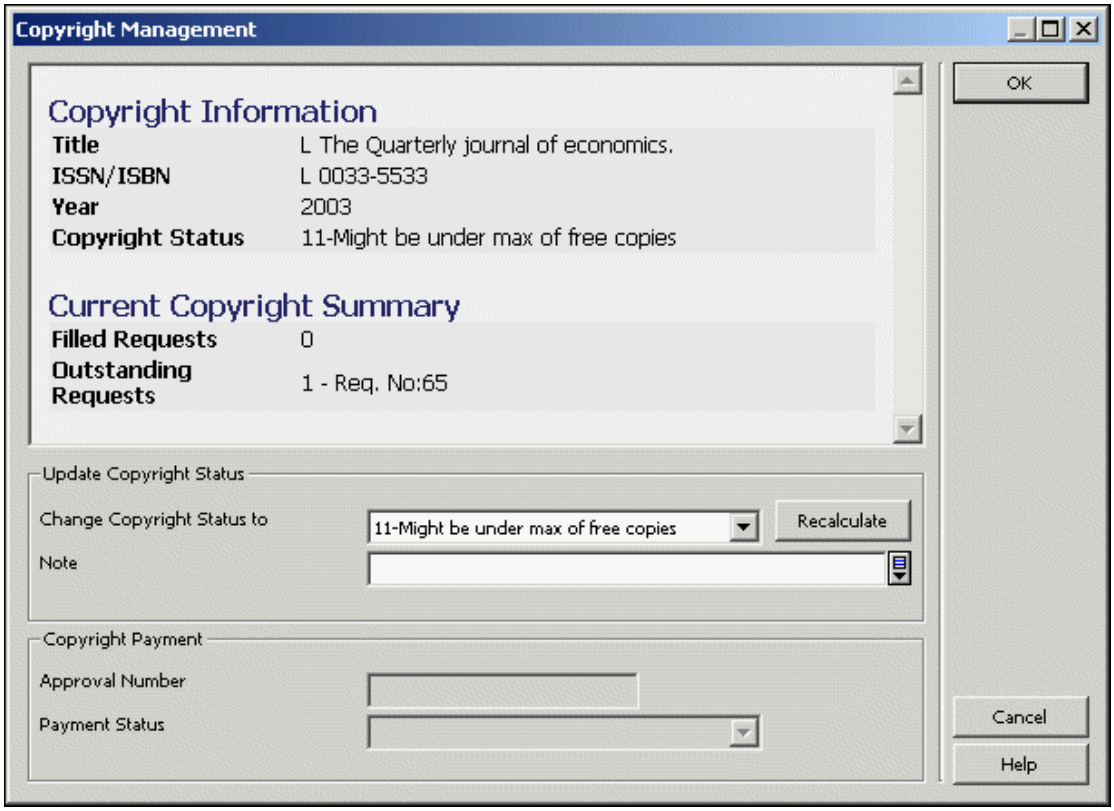
50 – Canceled/Expired/Deleted

51 – Unfilled

34.2.4 Copyright Management GUI ILL Interface

The GUI ILL Borrowing interface enables you to view the copyright status and the overall copyright summary of the title within the pre-defined number of years. In addition, there is an option to update/recalculate the copyright status.

To access the copyright management window, click the **Copyright** button from the Borrowing Request List. The **Copyright** button is enabled for each non-returnable requests (copy or electronic).



The Copyright Management window displays the following information:

Request information: title, ISSN/ISBN and year

Request latest copyright status.

Copyright Summary of the title within the pre-defined range of years. The summary shows the current number of outstanding and filled requests for the title within the pre-defined range of years. The counters include the current request. All related request numbers are listed. Note that this summary is updated each time the Copyright Management window is opened so it might happen for example that the copyright status shows "11-Might be under the max of free copies" but the counter of 'Outstanding Request' is already above the max of free copies.

The Copyright Management window allows manual update of the copyright status by a staff user in one of the following ways:

Change copyright status – the staff user can update the copyright status to one of the following final statuses:

00 – Free of copyrights fee

10 – Under max. number of free copies

20 – Above max. number of free copies

A request with a 'final' copyright status is not checked again by the system upon request fulfillment therefore, the manual change will not be overridden by the automatically system calculation.

To change the copyright status, in the Copyright Management window, select the required status from the pull-down list, add a free-text note if you want, and click **OK**. The manual update and the note (if exist) are written to the request log transaction.

Recalculate – The 'recalculate' action manually activates the system re-valuation of the copyright status. This action might be required since the system automatically calculated the copyright status only in two cases (request's creation and fulfillment/un-fulfillment) but the staff user may want to reflect the most updated copyright status in an intermediate step.

To do so, in the Copyright Management window, click **Recalculate**. The updated status will appear in the 'Change Copyright Status to' field. You may add a free-text note. If you want to proceed with the action, click **OK**. The manual update and the note (if present) are written to the request log transaction.

Note that at this stage, the Copyright Payment section of the Copyright Management form is disabled.

34.2.5 Request Log for Copyright Activity

Each update action of the copyrights status is written into the Request Log and can be viewed via the GUI ILL's Borrowing Request Log. This is especially important for manual copyright changes but also applies to changes made automatically by the system. In the event of a manual update/recalculate, the free note text (if present) also appears in the log.

The creation of the copyright log is dependent on the setup of the ILL Library table: tab46.eng (./xxx40/tab/tab46,eng). Relevant transaction (Col.1) for US copyright: 57 and 58.

In order to activate trans. 57 and 58, set Col. 2 to Y (as in the following example):

```
57 Y N LSet copyright status
58 Y N LManually update copyright status
```

34.3 Processing the Copyright Payment Report

The 'Copyright Payment' section is enabled in the Copyright Management window of each filled request with a copyright status of '20 -Above the maximum number of free copies'.

The 'Copyright Payment' section includes the following fields:

- Approval Number – displays the approval number that is assigned to the copyright file of the request
- Payment Status – shows the payment status of the request

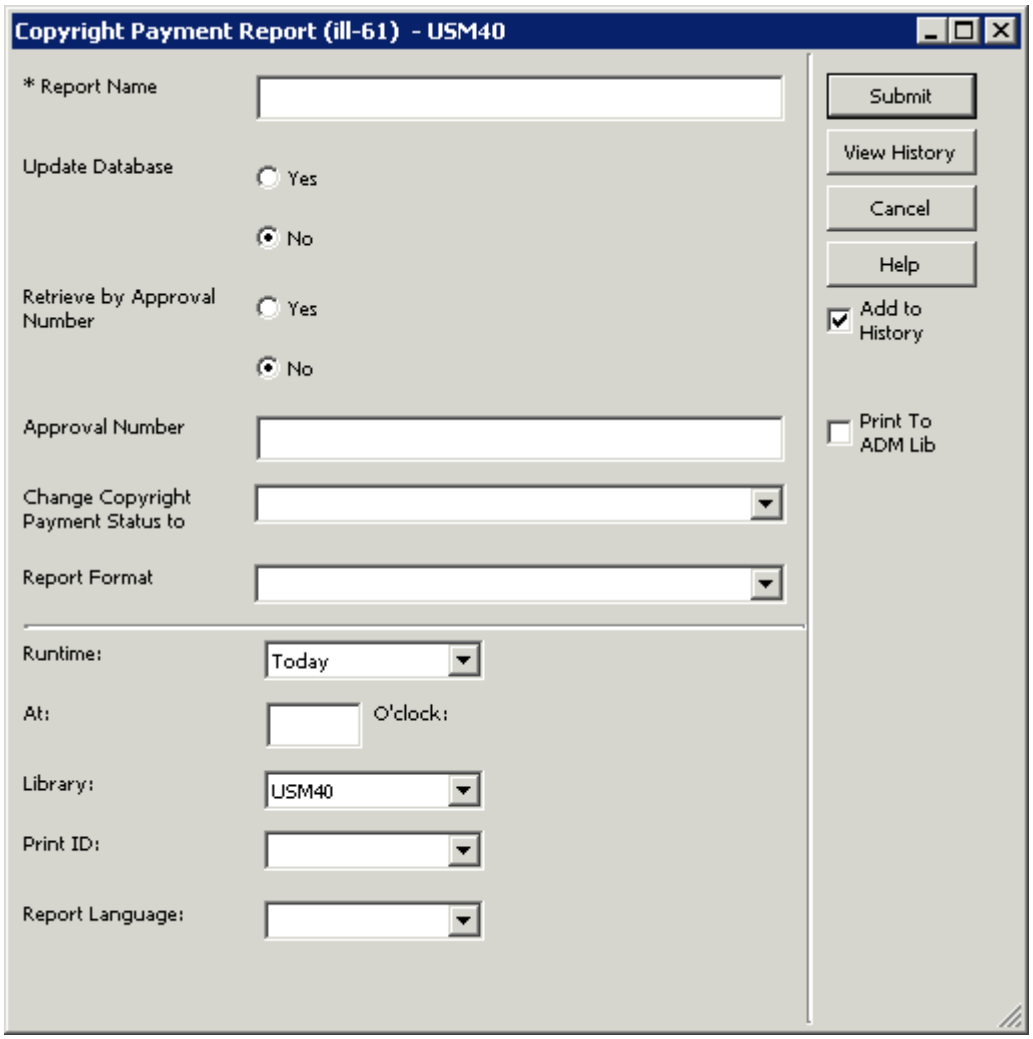
Upon fulfillment of a request, the initially assigned values for those fields are:

- Approval Number – left empty
- Payment Status - 'Not handled' (copyright payment procedure has not yet been concluded).

Both fields, Approval Number and Payment Status, can be processed by the Copyright Payment Report service (ill-61). There is also an option to manually update these fields.

34.3.1 The Copyright Payment Report (ill-61)

The copyright payment report is run for each ILL unit and activated by the GUI ILL "Copyright Payment Report" (ill-61) service.



This service performs the following two actions and produces an output report.

Action 1: Assigns a common approval number to all borrowing requests which are filled with a copyright status of 'Above the maximum number of free copies' (status code 20). An output report is produced, showing the details of all retrieved requests.

Action 2: Changes the copyright payment status of requests that are assigned to a common approval number. An output report is produced showing the details of all retrieved requests.

Following is an explanation of each action:

Action 1

This is the first phase of handling the copyright payments. It can be run in two modes: 'Update mode' and 'Preview mode'.

Update Mode: The 'Update mode' enables you to assign an approval number to all filled requests with a copyright status of 'Above the maximum number of free copies' and a copyright payments status of 'Not Handled'.

Upon submitting the batch, the copyright payment status of all retrieved requests is changed from 'Not Handled' to 'Reported'. The approval number is automatically derived from the last-z418-app-number system counter in the UTIL G/2 counter of the ILL library.

The 'Copyright Payment Report' printout (template file name: copyright-payment-report.xsl) lists all retrieved requests that are assigned to the 'Approval Number'.

In order to run action 1 in 'update mode', select the following batch parameters:

- Report Name - Type in a file name for the printout file.
- Update Database field - select Yes
- Retrieve by Approval Number field - select No
- Approval Number field - leave empty
- Change Copyright Payment Status to field - leave empty

Preview Mode: This mode enables you to produce a 'Copyright Payment Report' of all filled requests with a copyright status of 'Above the maximum number of free copies' whose copyright payments have not been yet handled. It does not assign an approval number, and does not change the payment copyright status to' 'Reported'.

You may want to run Action 1 in Preview mode in order to have a general overview of all predicted payments for copyright before actually starting to handle the copyright payments.

In order to run Action 1 in Preview mode, select No in the Update Database field. (All other fields should be set as in 'Update mode').

Action 2

This is the second phase of handling of the copyright payments report. It allows you to change the copyright payment status of requests that are assigned to a common approval number (using Action 1 – Update mode). You may use this action in order to indicate that copyright fees for requests that belong to the same approval number group are paid (change to status: Paid) or authorized for payment (change to status: Payment Authorization Given). An output report is produced showing the details of all retrieved requests (template file name: copyright-payment-report.xsl).

In order to run Action 2, select the following batch parameters:

- Report Name - Type in a file name for the printout file.
- Update Database field - select Yes
- Retrieve by Approval Number field - select Yes
- Approval Number field - type the approval number of the group of requests whose copyright payment status you want to update.
- Change Copyright Payment Status to field - Select the copyright payment status that you want to assign to the requests that belong to the approval number group.

The copyright payment status can be one of the following:

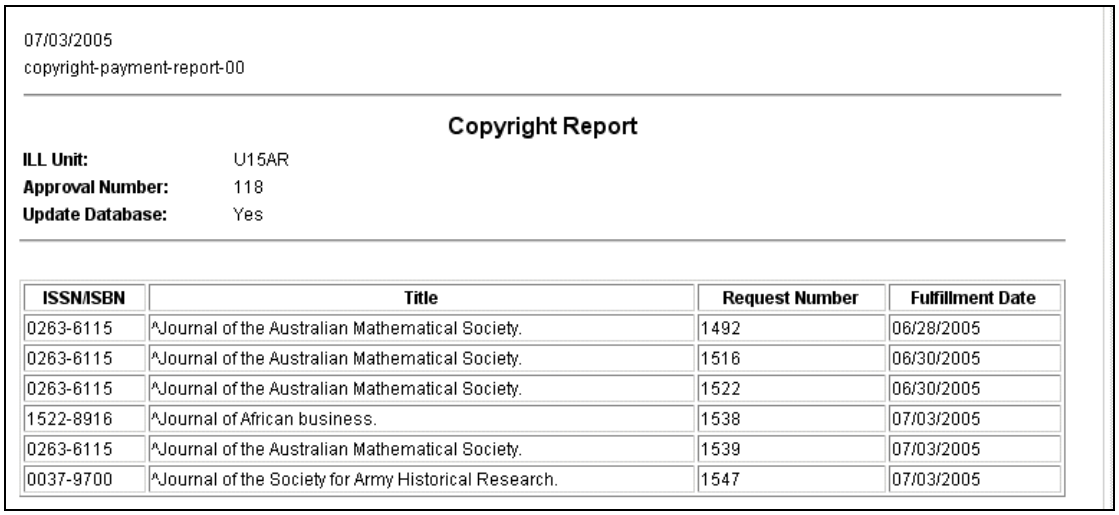
- Payment Authorization Given - This status indicates that the copyright fees for the approval number group are already approved.
- Paid - This status indicates that the copyright fee has already been paid.

- Dismissed - This status indicates that requests are dismissed from copyright payment handling. It also removes the approval number from all the retrieved requests.
- Not handled - This status indicates that the staff user currently does not want to handle the copyright payment of the retrieved requests. They are re-selected when Action 1 of ill-61 is run. It also removes the approval number from all the retrieved requests.

Note that additional copyright payment statuses can be added upon setting the following files: ./alephe/pc_b_eng/p-ill-61.xml (the interface of the ill-61service) **and** the COPYRIGHT-PAY-STATUS menu of the ILL library's pc_tab_exp_field.<eng> table:

Following are example outputs of the 'Copyright Payment Report service' (ill-61).

Sample 1: Action 1 – Update mode



07/03/2005
copyright-payment-report-00

Copyright Report

ILL Unit: U15AR
Approval Number: 118
Update Database: Yes

ISSN/ISBN	Title	Request Number	Fulfillment Date
0263-6115	^Journal of the Australian Mathematical Society.	1492	06/28/2005
0263-6115	^Journal of the Australian Mathematical Society.	1516	06/30/2005
0263-6115	^Journal of the Australian Mathematical Society.	1522	06/30/2005
1522-8916	^Journal of African business.	1538	07/03/2005
0263-6115	^Journal of the Australian Mathematical Society.	1539	07/03/2005
0037-9700	^Journal of the Society for Army Historical Research.	1547	07/03/2005

The above example report ran in 'Action 1 - Update mode'.

The batch parameters were:

- Update Database field - Yes
- Retrieve by Approval Number - No
- Approval Number - left blank
- Change Copyright Payment Status to – left blank

As a result of the batch run, approval number 118 has been assigned to all retrieved requests and their 'copyright payment status' has been updated to 'Reported'.

34.3.2 Manual Control of Payment Status

At this stage, based on the ill-61 report results, the staff user may manually change (update the approval number and/or the 'copyright payment status') the reported requests.

For example, the above report includes request number 1538. To update the 'Copyright Payment' information of the request, enter the 'Copyright Management' window of ILL request 1538 (click the **Copyright** button).

The Copyright Payment section shows the approval number and the payment status assigned as a result of ill-61 running.

If you want to exclude this request from the copyright payment procedure, you may manually change the payment status to 'Dismissed' or 'Not handled'.

- Dismissed - This status dismisses the request from copyright payment handling. It also removes the approval number from the request.
- Not handled - This status indicates that the staff user currently does not want to handle the copyright payment of the retrieved requests and they will be picked again when action 1 of ill-61 is run. It also removes the approval number from the request.

Sample 2: Action 2

07/03/2005
copyright-payment-report-00

Copyright Report

ILL Unit: U15AR
Approval Number: 118
Update Database: Yes
Copyright Payment Status: Payment Authorization Given

ISSN/ISBN	Title	Request Number	Fulfillment Date
0263-6115	^Journal of the Australian Mathematical Society.	1492	06/28/2005
0263-6115	^Journal of the Australian Mathematical Society.	1516	06/30/2005
0263-6115	^Journal of the Australian Mathematical Society.	1522	06/30/2005
0263-6115	^Journal of the Australian Mathematical Society.	1539	07/03/2005
0037-9700	^Journal of the Society for Army Historical Research.	1547	07/03/2005

The above sample report ran on 'Action 2 (change payment status for approval number group).

The batch parameters were:

Update Database field – Yes

Retrieve by Approval Number - Yes

Approval Number – 118

Change Copyright Payment Status to – Payment Authorization Given

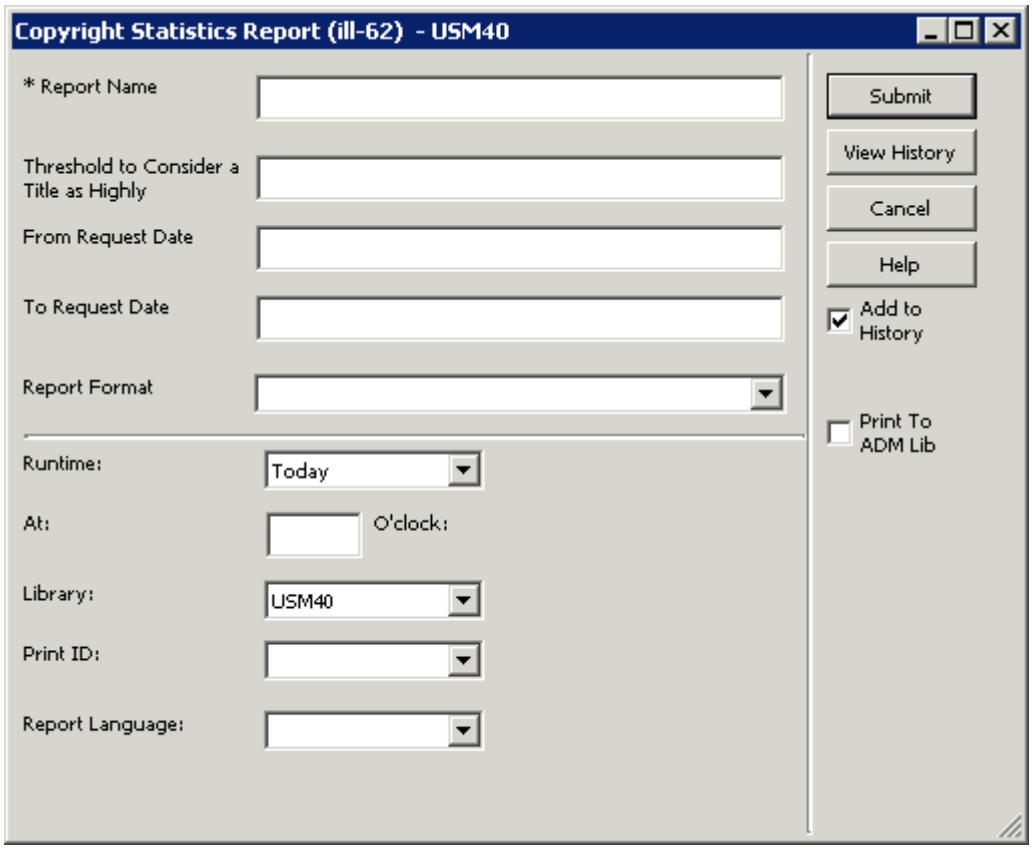
As a result of the batch run, all requests assigned to approval number 118 (not including request number 1538 that has been manually dismissed) were assigned as 'Payment Authorization Given'.

Later on, if you want to declare all of group 118 as 'Paid', you can run ill-61 with the same batch parameters except that 'Change Copyright Payment Status to' should be 'Paid'.

34.4 Copyright - Statistics Report

The GUI ILL service 'copyright – statistics report' (ill-62) detects highly requested titles, and therefore should be candidates for acquisition.

It runs over all filled request that their copyright status is either '10 - Under the maximum number of free copies' or '20 - Above the maximum number of free copies' and produces a report that shows the number of request per title and the number of different requesters (patrons) that requested each title (template file name: copyright-statistic-report.xsl).



The report's input parameters are:

- Report Name - Type in a file name for the printout file.
- Threshold to Consider a Title as Highly Requested – the report includes all titles that have been requested according to the number of times defined in this parameter (inclusive). If this field is left empty, the value 1 is considered as the threshold (that is, all titles that have been requested at least once).
- From/To Request Dates – enter a range of dates for the request's submission dates. The date format is: YYYYMMDD.

Sample output of p-ill-62:

07/04/2005
copyright-statistic-report-00

Copyright Statistic Report

ILL Unit: U15AR
Threshold: 3
From Request Date: 06/01/2005
To Request Date: 06/30/2005

ISSN/ISBN	Title	No of Requests	No of different Requesters
0263-6115	^Journal of the Australian Mathematical Society.	9	2
0037-9700	^Journal of the Society for Army Historical Research.	4	3
1522-8916	^Journal of African business.	4	1

35 Mapping Fields in ILL Documents

Every lending ISO ILL request has a matching document in the ILL Library. The document includes two types of information:

- Full list of the incoming PDU tags. The information is stored in fields for which:
  - The field code is ILL.
  - Subfield $$a is the tag name.
  - Tag $$b is the tags information.

For example:

ILL L $$aIllRequest.ItemId.Author$$bHill, George Birkbeck Norman,

ILL L $$aIllRequest.ItemId.Title$$bHarvard college :

- Bibliographic information, stored bty default in MARC21 format. For example, the author and title information in the above example is stored as:

100 L $$aHill, George Birkbeck Norman,

245 L $$aHarvard college :$$bby an Oxonian /

The following describes how to:

- Map information that is stored in ILL fields into MARC21 fields to enable display of the fields' information.
- Change the bibliographic information, which is by default in MARC21 form, into other MARC types.

When the ILL Library document is created, the document is fixed using the ILL-L section in the tab_fix table of the ILL Library. Manipulating the document contents may be done in the following manner:

1. Set the following line in the tab_fix table of the ILL Library:

   ILL-IL fix_doc_do_file_08 fix_ill

2. Set the fix_ill script in the /tab/import directory of the ILL Library. The script is used to change field codes and achieve the above listed goals.

For example, consider an example in which you would like to display the maximum cost information that is received in the ISO ILL Request PDU.

a) Set the fix_ill script as follows:

```
1ILL##   CHANGE-FIRST-IND-MATCH  #,0,$$aIllRequest.CostInfoType.MaximumCost.MonetaryValue
1 ILL     COPY-FIELD             COS
1 ILL0    DELETE-FIELD
1       COS     REPLACE-STRING         $$aIllRequest.CostInfoType.MaximumCost.MonetaryValue$$b,$$a
```

This will cause a ILL Library document field such as:

```
ILL   L $$aIllRequest.CostInfoType.MaximumCost.MonetaryValue$$b17.00
```

to become

```
COS   L $$a17.00
```

b) You can now add the COS field for display in the GUI, by configuring the bib_format and edit_paragraph tables. For example:

1. Set the bib_info table in the ILL Library data_tab to set bib_format of the lending requests. For example:

   ```
   lend-request-expand            300
   ```

2. Set the edit_paragraph in the ILL Library data_tab as follows:

   edit_paragraph:

   ```
   !1    2   3     4          5      6
   !!!-!!!!!-!-!!!!!!!!!!-!!!!!!!!!!-!
   300 1#### D             .^         A
   300 245## D
   ```

```
                    300 COS    D Max Cost:
```

Another example is if you want to map the document's bibliographic information from the default MARC21 into MAB. In this case set the fix_ill script to do the mapping. The following will map the title, which is field 245 in MARC21, into the equivalent 331 MAB field.

```
1 245##             CHANGE-FIELD                    331
```

The following, will change the 260 $$b MARC21 Publisher field into the equivalent 412 $$a MAB field:

```
1 260##             COPY-FIELD                      412
1 412##             DELETE-SUBFIELD                 a
1 412##             DELETE-SUBFIELD                 c
1 412##             CHANGE-SUBFIELD                 b a
```

The following is the default MARC21 mapping of incoming ILL Requests:

MARC Field	ISO PDU Tag
020	IllRequest.ItemId.ISBN
022	IllRequest.ItemId.ISSN
100	IllRequest.ItemId.Author
245	IllRequest.ItemId.Title
250	IllRequest.ItemId.Edition
260	IllRequest.ItemId.PlaceOfPublication
300	IllRequest.ItemId.Pagination
440	IllRequest.ItemId.SeriesTitleNumber
503	IllRequest.ItemId.SystemNo.External.IPIGSystemNo.IPIGNo.Other.PersonOrInstitutionSymbol.Institution
503	IllRequest.ItemId.SystemNo.External.IPIGSystemNo.IPIGNo.WellKnownSystem
504	IllRequest.ItemId.SystemNo.External.IPIGSystemNo.IPIGNo.DatabaseId
505	IllRequest.ItemId.SystemNo.External.IPIGSystemNo.IPIGNo.Number
590	IllRequest.ItemId.VerificationReferenceSorc
592	IllRequest.ItemId.ItemType
70012	IllRequest.ItemId.AuthorOfArticle
74002	IllRequest.ItemId.TitleOfArticle
852	IllRequest.ItemId.CallNumber
900	IllRequest.ItemId.PublicationDateOfComp
901	IllRequest.ItemId.VolumeIssue
902	IllRequest.ItemId.SubTitle
903	IllRequest.ItemId.Publisher
904	IllRequest.ItemId.PublicationDate

```
 905    IllRequest.ItemId.HeldMediumType
906    IllRequest.ItemId.PublicationDate
```

Ex Libris
Aleph

System Librarian's Guide - Items

Version 23

Ex Libris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 LIBRARIES 5

2 COLLECTIONS 5

3 COLUMN HEADINGS 5

4 DROP-DOWN MENUS 6

5 HOLDINGS RECORD 8

6 CALL NUMBER FIELD 8

6.1 Copying the Call Number from the BIB Record to the Item Record 12

6.2 Automatic Assignment of the Call Number Type Field 13

7 ITEM HISTORY RECORD 13

7.1 Setting up Item History Record 13

7.2 Viewing Item History in GUI 14

8 SORT ORDER FOR LIST OF ITEMS 15

9 CLIENT SETUP (CATALOG.INI) 16

9.1 Print.ini 17

10 ITEMS TABLES 17

11 ITEMS-RELATED ENTRIES IN TAB100 20

12 ITEM STATISTIC OPTIONS 24

13 PRINT TEMPLATES 25

13.1 Available Templates 25

14 ITEMS VALUES OF PC_SERVER_DEFAULTS 26

15 CHECKSUM OF ITEM BARCODE 27

16 COMPLETE VOLUME REPORT 28

17 ITEMS VALUES OF ALEPH_START 28

18 ITEM CALL NUMBER – AUTHOR NUMBER AND CHRONOLOGY NUMBER 29

18.1 The Special "852 Form" (Call Number Form - Type 6) 30

The Special "852 Form" (Call Number Form - Type 7) 30

18.2 tab_author_num.conf 33

18.3 Author Number and Chronology Number: Methods and Algorithms 36

18.3.1 Author Number by Cutter Three Figure Method 36
18.3.2 Author Number by Cutter Three Figure Without Caret Mark Method 36
18.3.3 Author Number by Cutter Sanborn Method 37
18.3.4 Author Number by Lee Jai Chul Method 37
18.3.5 Author Number by Elrod 37
18.3.6 Author Number by Jang Il Sei 39
18.3.7 Chronology Number Algorithm 40

18.4 Other Related Tables 42

19 BARCODE CHECKSUM ROUTINES FOR KEIO 43

19.1 Item Barcode Algorithm - checksum_valid_item_bar_keio 43

19.2 Patron Barcode Algorithm - checksum_valid_bor_id_keio 48

20 ITEM CHANGES ALERT 50

21 MULTIPLE ITEM ADDITION ALERT 51

1 Libraries

To define the administrative libraries to which the user may connect in the Items function, follow these steps:

1. Define the ADM libraries related to the BIB library in the `library_relation` table of the `$alephe_root/tab` directory.
2. Go to the CATALOG/TAB directory of the client and open the `catalog.ini` file. In the [ADM Library] section define the default ADM library.
3. From the ALEPH menu, select the appropriate ADM library from the "Select ADM Library" submenu.

2 Collections

In order to define Collections, edit `tab40` using UTIL G/4/40 in an ADM library. Following is a sample from the table:

```
! 1     2   3     4
!!!!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!>

REF   UEDUC L Reference
SHL   ##### L Short Loan
WIDEN ##### L Widener Collection
GEN   ##### L General Collection
```

Column 1 is the code identifying the collection. Enter up to five characters.
Column 2 is the sublibrary to which the collection belongs. You may use the # character as a wildcard. In the above example, ##### means that the collection is common to all sublibraries.
Column 3 is the code for the character set of the menu choice. Insert L for Latin.
Column 4 is the name of the collection as it will appear in the menu in the GUI.

3 Column Headings

`pc_tab_col.lng` defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (USM01) table `pc_tab_col.lng` using UTIL I/9 or the ALEPHADM module.

The following is a list of the Items panes which use `pc_tab_col.lng` for formatting data and their identifiers (Column 1 in `pc_tab_col.lng`).

Identifier	**ITEMS GUI PANES**
PC_ITEM_ITEM_LIST	Items List Pane
PC_COM_HOLD_H_LIST	All Items History/Hold Requests
PC_COM_HOLD_H_LIST	Item History/Hold Request

Identifier	ITEMS GUI PANES
PC_COM_HOLD_H_LIST_1	Item Circulation History - Hold Request List in ACQ and CAT if ITEM-BOR-INFO-CIRC-ONLY=Y
PC_COM_LOAN_H_LIST	All Items History/Loans
PC_COM_LOAN_H_LIST	Item History/Loans
PC_COM_LOAN_H_LIST_1	Item History - List of Loans in ACQ and CAT if ITEM-BOR-INFO-CIRC-ONLY=Y
PC_ALL_ITEM_HISTORY	All Items History/Item Changes
PC_ITEM_HISTORY	Item History/Changes
PC_ITEM_BIND	Items List for Binding
PC_ITEM_HOL_LIST	Item Information/Hol Links

4 Drop-down Menus

You may add or delete choices for some of the menus of the Item Form by editing the following ADM library tables:

1. `pc_tab_exp_field_extended.lng` using UTIL L/1.
2. `pc_tab_exp_field.lng` using UTIL L/2.
3. ALEPH ADM tables:
    - tab40.lng
    - tab25.lng
    - tab15.lng

pc_tab_exp_field_extended.lng
The drop-down menus defined here are for Item Inventory Counter, Item Call No, Item 2nd Call No., Item Call No.'s prefix and suffix.

The following list shows the `pc_tab_exp_field_extended.lng` entries for drop-down menus which appear in various forms in the Items function.

In contrast to `pc_tab_exp_field.lng`, all menus that are defined in `pc_tab_exp_field_extended.lng` are sublibrary sensitive (Column 2).

pc_tab_exp_field_extended.lng entries	GUI Fields
INVENTORY-NUMBER	Inventory Number
ITEM-LOCATION	Call Number
ITEM-LOCATION-2	2nd Call Number

pc_tab_exp_field_extended.lng entries	GUI Fields
LOCATION-GR	Call Number/2nd Call Number
LOCATION-PREFIX	Call Number/2nd Call Number
LOCATION-SUFFIX	Call Number/2nd Call Number
LOCATION-SOURCE	Call Number/2nd Call Number
LOCATION-USIG	2nd Call Number/2nd Call Number
ITEM-PART-1	Call Number/2nd Call Number – Author Number (special 852 Form - types 6 and 7)

pc_tab_exp_field.lng

The drop-down menus defined here are for Item Call No. Type, Item Process Status, Item List Sort Type, Item Gap Indicator, and Item Statistic, etc.

pc_tab_exp_field.lng entries	GUI Fields
ITEM-LOCATION-TYPE	Call Number Type
ITEM-PROCESS-STATUS	Item Process Status
ITEM-SORT-TYPE	Items List/Sort Options
ITEM-GAP-INDICATOR	GAP Indicator
ITEM-STATISTIC	Statistic
ITEM-STATISTIC-1	Statistic
ITEM-STATISTIC-2	Statistic
ITEM-STATISTIC-3	Statistic
ITEM-STATISTIC-4	Statistic
ITEM-DEPOSITORY-ID	Remote Storage ID
ITEM-PART-2	Call Number/2nd Call Number – Chronology Number (special 852 Form -type 6)
ITEM-PART-1-TAG	Call Number/2nd Call Number – Author Number field-Alternative BIB tags (special 852 Form - types 6 and 7)

ALEPHADM Tables

tab40.lng - used to define the drop-down menu of Item Collection.

tab25.lng - used to define the drop-down menu of Material Type.

tab15.lng - used to define the drop-down menu of Item Status.

5 Holdings Record

You can determine whether or not a Holdings Record controls the location field of the item record. To do so, follow these steps:

1. Edit tab100 in your Administrative library. In that table, the relevant line is:

   ```
   X852-ITEM-OVERRIDE=Y
   ```

   If you want the Holdings Record to control the location information, enter *Y* in column 2 as shown above. If you do not want the Holdings Record control, enter *N*, instead.

2. Edit `library_relation`. Make sure that a line similar to the following line appears in that table:

   ```
   HOL USM50 USM60
   ```

   You should replace "USM50" with the name of your ADMinstrative library. and "USM60" with the name of your HOLdings library.

6 Call Number Field

This discussion refers to the "Call No." and "Second Call No." fields of the Item Form (and the Subscription Form in the Serials module). The following explanations are relevant if you have decided NOT to use the holdings record for control of the call number. It is up to you, the System Librarian, to determine how these fields "behave." You can enable your staff to assign a call number in each field by typing in text, choosing from a drop-down menu, filling in a form, or activating a counter by typing **?** and the code of the counter.

To define which method will be used by your staff, go to the ALEPHCOM/TAB directory and open the ALEPHCOM.INI file. Go to the section labeled `[ExpandField]`. An example of this section is shown below:

```
[ExpandField]
ItemLocation1Field=4
ItemLocation2Field=3
SerialLocation1Field=4
SerialLocation2Field=3
```

Use "ItemLocation..." for the Items function and "SerialLocation..." for the Serials function. You may enter *1*, *2*, *3*, *4*, *5*, 6 or *7* to the right side of the equal (=) sign. The meaning of each value is listed below:

- **1** - a drop-down menu will be available whose choices you must define in `pc_tab_exp_field_extended.lng` (UTIL L/1 in an Administrative library) under the IDs ITEM-LOCATION and ITEM-LOCATION-2. The choices may include counters that have been defined using UTIL G/2.

- **2** - the user must type a call number in the field, or activate a counter by typing **?** and the code of the counter that is defined in UTIL G/2.

- **3** - for use with closed stacks (where the library "counts" rather than "classifies" items in order to assign the item's call number`). The user must fill in the Build Call No. Field form to "build" the call number from three components known (in German nomenclature) as NR (number), USIG (prefix/suffix) and GR (format or size). The choices available for the GR and USIG fields in the Build Call Number Field form are defined in `pc_tab_exp_field_extended.lng`. You must define the rules for combining the three components in `tab_acc_location` (in the tab directory of an Administrative library). In addition, this option activates a counter that you must define using UTIL G/2.

- **4** - by clicking the right-arrow in the Call Number field, the user accesses a browse list of call numbers. He can select a call number from the list. The selected call number may then be modified. To support this option, there is no need to set up an Index of call numbers; the system automatically does this, if the ADM library includes Z311 in its `file_list`.

  This option 4 and the Z311 record have additional functionality for automatic assignment of sequential call numbers. This is dependent on assigning call number type "i" when creating an item record. The Z311 is used to register the last sequence number assigned for a sublibrary+prefix, so that the next highest number can automatically be assigned when an item record is created. Prefixes do not have to be set up beforehand; when a new sublibrary+prefix combination i is detected, a new Oracle record is created.

  So, if the call number type is set to "i" and if the user enters a prefix code followed by "?" (for example, ABD ?) in the call number field, the system assigns the next number in sequence. The sequence is set according to sublibrary+prefix. (NOTE: a suffix may be added, but it is not taken into account when assigning a sequence number).

  - If the user enters a prefix code followed by a sequence number, and the sequence number is higher than the Oracle sequence number, this higher number is assigned to the Oracle record, and sequence numbers proceed from there.

  - If the user enters a prefix code followed by a sequence number, and the sequence number is lower than the Oracle sequence number, the Oracle record is not updated.

  This functionality is supported by tab_z311 of the ADM library and tab100 flag: INVENTORY-NUMBER=Z311. tab_z311 is used to define the z311 programs used by the system for the creation of the Z311 counters for the z30-call-no, z30-call-no-2 and z30-inventory-number fields.

The following is a sample of the table:

```
!  1                                    2     3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
z30-call-no                             0 create_z30_location_i
z30-call-no-2                           0 create_z30_location_i
z30-inventory-number                    1 create_z30_inv_number
```

Key to the tab_z311 table:

- **Column 1 - Item field name**
  Available for: z30-call-no, z30-call-2-no, z30-inventory-number.

- **Column 2 - Z311 Record type;**

  - 0 - location

  - 1 - inventory-number

- **Column 3: Program name**

- **5** - the user may define the call number by filling in a form that has standard 852 subfields. The form also includes a Browse button to select a call number from a Browse List of Call Numbers. The user can copy a line from the Browse list and the information will be copied into the form, in the proper subfields.

- 6 - the user may define the call number by filling in a special 852 form. This form includes the option to calculate Author number and Chronology Number based on various methods. For example, for Latin material, Cutter Sanborn and Cutter Three Figure methods and for Korean material (Hangul and Hanja), Lee Jai Chul, Jang Il Sei, and Elrod methods. The form also includes a **Browse** button to select a call number from the **Browse List of Call Numbers**. The user can copy a line from the Browse list and the information is copied into the form in the proper subfields. For more details, see the Item Call Number- Author Number and Chronology Number section of the A*leph 20.01 System librarian's Guide – Items*.

- 7 - the user may define the call number by filling in a special 852 form. This form includes the option to calculate Author number based on various methods. For example, for Latin material, Cutter Sanborn and Cutter Three Figure methods and for Korean material (Hangul and Hanja), Lee Jai Chul, Elrod, and Jang Il Sei methods. The form also includes a **Browse** button to select a call number from the Browse List of Call Numbers. The user can copy a line from the Browse list and the information is copied into the form in the proper subfields. For more details, see the **Item Call Number- Author Number and Chronology Number** section of the *Aleph 20 System Librarian's Guide – Items*.

Defining Counters With UTIL G/2

Counters may be defined for options 1 and 2 above using any string of characters. However, the counters that may be defined for option 3 (NR/USIG/GR) follow very specific guidelines.

Three types of counters may be defined for option 3 and you may have an unlimited number of each type:

- Sublibrary plus GR (format) (examples of counter IDs are "USM1-2", "USM1-4")
- Sublibrary only (for example, "USM1")
- GR (format) only (e.g, "2")

When the librarian uses the Build Call No. Field form to enter the GR (format) and the USIG (prefix/suffix), the system checks the counters in UTIL G/2 to determine which counter should be used for assigning the NR (number). The system first searches for a counter that is Sublibrary+GR. If one is not available, the system will then search for a counter that is defined for the Sublibrary. If that is not found, the system will search for a counter defined for the GR (format) only.

The rules for combining the three components (NR/USIG/GR) for display in the location field are defined in `tab_acc_location` (in the tab directory of an Administrative library). This table also defines the delimiters that should be used when displaying the three components. However, in case the librarian does not provide information about the GR (format) or USIG (prefix/suffix), the system will not display irrelevant delimiters. Unnecessary delimiters that may not be displayed are:

- hyphen -
- slash /
- bracket ()
- period .

6.1 Copying the Call Number from the BIB Record to the Item Record

It is possible to copy the call number from the 090$$a subfield of the BIB record to the call number type field of the Item record. This is done by setting the value "i" in the call number type field.

In order to activate this functionality, set tab_Z311 with the create_z30_call_no_i_bib_090 routine and tab100 flag:: INVENTORY-NUMBER=Z311.

The following is an example setup in ./usm50/tab/tab_z311:

```
!  1                                  2     3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
z30-call-no                           0 create_z30_call_no_i_bib_090
```

In the following cases, the call number is not copied from BIB 090$$a to the Item record:

- BIB 905$$a does not exists.
- The linked BIB record contains a 940 field.
- The Item record is linked to a HOL record.
- The item is already populated with a call number.

The call number type field with value “i” is relevant for item creation and update workflows, such as: Cataloging, Acquisitions, Serials, and services.

6.2 Automatic Assignment of the Call Number Type Field

You can set the Item and Subscription forms to be populated automatically with the “i” value in the call number type field. Configure the following in ./alephcom/tab/alephcom.ini:

```
[ItemForm]
CallNoType1=i
```

After setting this configuration, the call number type field is automatically populated with the value “i” in the Item and Subscription forms of the Aleph interface. This causes the call number to be copied from the BIB record to the item record when you click the Add or Update button. If the values of the Item and Subscription forms are saved as the default using the Save Defaults or Subs Default buttons and CallNoType is set with value “i” in ./alephcom/tab/alephcom.ini, the default value does not override the call number type field setting and it remains set to “i”.

7 Item History Record

A log file of all changes made to an item record can be created and viewed in the GUI-Item List-History window.

7.1 Setting up Item History Record

If your library is interested in having a log file of changes made to an item record, set the value to "Y" in the CREATE-Z30H flag of ./xxx50/tab/tab100 and define the fields that when updated will trigger the creation of an Item History record in ./xxx50/tab/tab_item_history.lng .

tab_item_history.lng list the actions and items fields that create an Item History log record. The following is a list of possible values that can be set in column 1 of tab_item_history.lng:

- ITEM-LOCATION
- ITEM-CALL-NO-2
- ITEM-COLLECTION
- ITEM-SUB-LIBRARY
- ITEM-PROCESS-STATUS
- ITEM-BARCODE
- BIND
- DELETE

- ITEM-STATUS
- SHELF-REPORT-NUMBER
- ITEM-MATERIAL
- ITEM-DESCRIPTION
- ITEM-CIRC-NOTE
- ITEM-INT-NOTE
- ITEM-OPAC-NOTE
- ITEM-LOCATION-TYPE
- ITEM-CALL-NO-2-TYPE
- ITEM-COPY
- ITEM-ENUM-A
- ITEM-ENUM-B
- ALL-NON-CIRC
- ALL-FIELDS

Use "ALL-FIELDS" in order to register a change in any other field. When using ALL-FIELDS, list it last in tab_item_history.lng.
If ALL-FIELDS is listed before the specific field, changing a listed field will register both "ALL-FIELD" and the specific field.

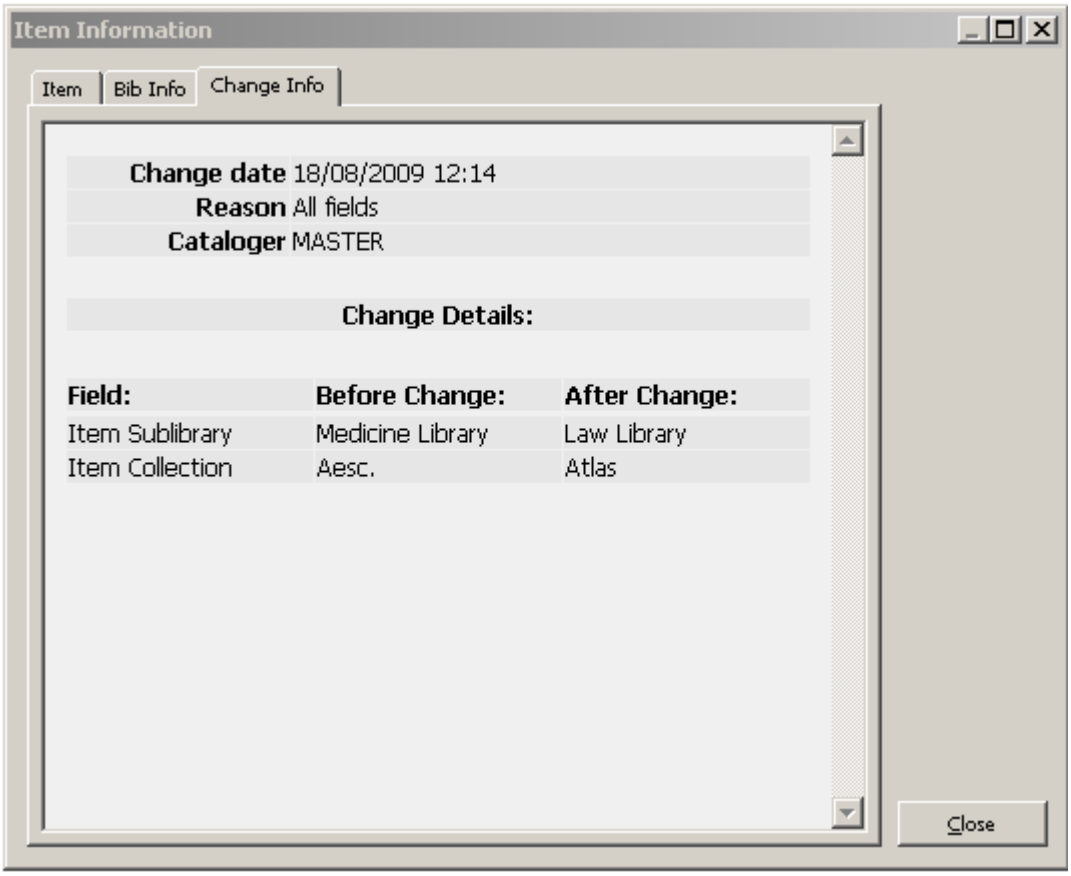
Use "ALL-NON-CIRC" in order to register all changes except item (Z30) changes caused as a result of the following item circulation activities:

z30-date-last-return
z30-hour-last-return
z30-ip-last-return,
z30-no-loans
z30-maintenance-count

When using ALL-NON-CIRC, list it last in tab_item_history.lng.
Make sure to NOT set both "ALL-FIELDS" and "ALL-NON-CIRC" in the table.
If ALL-NON-CIRC is listed before the specific field, changing a listed field will register both "ALL-NON-CIRC" and the specific field.

7.2 Viewing Item History in GUI

Item history actions can be viewed in GUI-Item list-History-Item Information (available in the following modules: Cataloging, Circulation, and Acquisitions/Serials).



The item information history fields are set in HTML display which is defined in the server file: ./xxx50/pc_display_lng/item-history-expand.

The Change Info tab can include item information "before and after" the change (to do this; refer to the sample demo file ./usm50/pc_display_lng/item-history-expand).

NOTES:

1) The "Change Details" section includes ALL changed field. This means that each updated field will be reported, regardless of whether it triggers an item history record or not.

2) If you set ALL-NON-CIRC in ./xxx50/tab/tab_item_history.lng and you are not interested in viewing the item's circulation related changes, edit item-history-expand to NOT include a call to the following fields:
z30h-date-last-return
z30h-hour-last-return
z30h-ip-last-return,
z30h-no-loans
z30h-maintenance-count

8 Sort Order for List of Items

The BIB library's `tab_z30_sort` table sets the sort order for a list of items in Web OPAC and GUI Search. The ADMinistrative library's `tab_z30_sort` table sets the order for a list of items in ADM functions. The order is set separately for the various

modules and functions. The table sets both the sort order and whether the order is ascending or descending, separately for "ISSUE" items and other items. Note: If the sort routine uses chronological or enumeration fields and these fields are empty, the item's description field (z30-description) will be used instead. See tab_z30_sort header for information about the various sorting types that can be set in columns 3 and 5.

The sort order options available to the librarian are displayed in a drop-down list in the panes displaying: Item List, Check-In List and Group List. The text in the drop-down list and the link to the relevant sort option in tab_z30_sort is defined in the following menus of the `pc_tab_exp_field.lng` table of the Administrative library:

ITEM-SORT-TYPE
For the Item List displayed by selecting the Item List root, or the List of Items root in the Cataloging, Acquisitions/Serials and Circulation modules.

SERIAL-CHK-SORT-TYPE
For items displayed in the Check-In List in the Acquisitions/Serials module.

SERIAL-GRP-SORT-TYPE
For items displayed in the Group List in the Acquisitions/Serials module.

SERIAL-SORT-TYPE
For the Items List of a particular subscription displayed by selecting the Items node under the Subscription root in the Acquisitions/Serials module.

The last sort option selected by the user is saved, and serves as the default the next time the list is displayed.

UTIL F/2/23 (of the ADM library) can be used to check item sort. This utility enables you to view the Z30 text according to which the item is sorted.

Enter the sort type as defined in tab_z30_sort - 00,01,02,03,...
Enter item key (doc number + sequence).
The result is the type of item (issue/non-issue) and the sort text of this item.

For example: to view sorting routine 07 of ADM record 55675 sequence 30:

```
Enter values as defined in Col. 3 or 5 of tab_z30_sort: 07
Enter z30 key (number+seq) (Q to quit) : 000055675000030
IN   -000055675000030-
<<<connect directly no listener
TYPE -ISSUE-
OUT  -3 UEDUC GEN 000000001 0000011 2005-
```

9 Client Setup (catalog.ini)

The `catalog.ini` file includes settings for the Items function.

[ADM Library]

This section determines the default ADM library of the connected library.

For example:
`usm01=usm50`

[**CatalogItemBar**]
`FgColorDescript`
This setting determines the color of the text displayed in the item bar.

[**ItemTree**]
`BKColor`
This setting determines the color of the Item Tab.

Note: Additional entries in the *ItemTree* must not be modified.

9.1 Print.ini

The `print.ini` file of the Catalog/Tab directory is used to determine which form is used for printing labels. The forms are determined in the ItemLabel and ItemLabelPrint lines as follows:

```
ItemLabel       00 N P label.prn
ItemLabelPrint  00 N P itemlbl.prn
```

10 Items Tables

LIBRARY TABLES

1. pc_tab_exp_field_extended.lng

`pc_tab_exp_field_extended.lng` defines the Item form drop-down menus (in Items, Acquisitions/Serials and Cataloging). This table is similar to `pc_tab_exp_field.lng` except that it has a column for a sublibrary (you may use #) which allows for sensitivity to sublibraries.

2. tab_export_03

This table is used in the ADMinistrative library to set the data of the item records for the barcode labels' output file. When constructing the `tab_export_03` table, you may combine fields. For example, in the table below, the sublibrary, a comma, and the collection together will take up only 11 spaces in the export file and will be cut off if too long so that the title/author will always start at position 34 of the output file.

```
!1             2                     3                 4
5
!!-!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!-!!!!!!!!!!
20 Z30-BARCODE
01 " "
00 Z30-SUB-LIBRARY
00 ","
11 Z30-COLLECTION
01 " "
00 245##              a          2####              a
00 "/"
```

```
40 1####              a          245##              b
01 " "
00 PST##              bchij
```

The above table construction will produce an output file (in table format) like this:

1579-10 UEDUC,GEN Computers and literacy/

XEDUC GEN LC149.5 C65 1985

2385-10 UEDUC,GEN Human sexuality. --/McCary, James Leslie.

XEDUC GEN ZQ 21 M123 1967

2592-10 ULINC,GEN Natur und Sittlichkeit bei Fichte //Boehm, Max Hildebert,

XLINC GEN B2849 E7 B6 1981

3. tab100

`tab100` is the central configuration table for system-level, server-level and library-level variables. See Items-related Entries in tab100 on page 20 and the `tab100` table header.

4. tab15.lng

Defines the item statuses and various parameters connected to these statuses.

5. tab25.lng

This table defines the list of possible values for Z30-MATERIAL (item material type). The list must also be defined in UTIL L/2, so that it will be available as a pull-down menu option for the registering item data.

6. tab40.lng

`tab40.lng` defines 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.

7. tab_filing_call_no

This table 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.

8. tab_item_history.lng

`tab_item_history.lng` defines in which cases a field change or an action will cause an item history record to be written. In addition, the table defines

the fields displayed in the Item History screen and their description. This information is accessible via the "History" button in the Items module.

9. tab_label.lng

`tab_label.lng` defines the prefixes that will be printed on the spine label. This table can contain up to 2000 lines.

10. tab_label_parse

`tab_label_parse` specifies the routines used to parse (split apart) the call number and item description when generating item labels. The definitions in this table are used to create the raw XML data. The relevant template should be checked to make sure it includes elements for printing this data.

11. tab_z30_sort

`tab_z30_sort` defines sort options for items. Different sort routines can be used in different parts of the system.

12. tab42

`tab42` controls automatic updating of the processing status field in the item record. Automatic updating is triggered by various actions or changes in the Acquisitions, Serials and Items modules. List of the various triggers (processing actions) that can affect the item process status can be found in the `tab42` header.

13. tab_checksum

`tab_checksum` is used for defining the checksum programs. The programs can be used for creation, validation and correction of item barcode, patron barcode, budget code prefix, vendor/supplier code prefix, and vendor/supplier additional code. The relevant programs for checksum of item barcode are:

1. codabar
2. patron/item barcode - site-specific UEA
3. patron/item barcode - site-specific Newcastle
4. patron/item barcode validation - site-specific Keio

14. tab_events.lng

1. `tab_events.lng` defines which transactions will write a record to the Z35 or Z69 Oracle tables.

   While tab_event.lng in the administrative library defines which transactions write a record to the Z35 Oracle table, the tab_event.lng table in the bibliographic library defines which transactions write a record to the Z35 Oracle table (types 10,11,12) or Z69 oracle table (types 20-35). While the

transactions in the Administrative library relate to Circulation, the transactions in the bibliographic library relate to Web OPAC and to Z39_server.

If the `tab_event.lng` table in the bibliographic library does not exist in the database then the events will not be created. This is not a problem.

15. **form_print_method**

    This table defines the print methods for XML-based printouts. The default method is XML_XSL.

16. **tab_z311**

    This table is used in the ADM library to define the Z311 programs used by the system for the creation of the Z311 counters for the z30-call-no, z30-call-no-2 and z30-inventory-number fields.

ALEPHE TABLES

1. **library_relation**

   `library_relation` defines the relation between libraries by library type.

2. **tab_sublibrary**

   `tab_sublibrary` is used to define the drop-down menu of the sublibrary field on the Item Information form.

11 Items-related Entries in tab100

This section details all the variables which are relevant to Items and set in tab100 of the Administrative library. Most of this information also appears elsewhere in this System Librarian chapter.

BIND-USE-DESCRIPTION

This variable is used to define what information ALEPH will return to external binding preparation software packages.

Possible values are: Y or N.

Y - Return Z30-DESCRIPTION in case Z30-CALL-NO is empty.

N - Always return the Z30-CALL-NO.

CHANGE-ITM-STS-LOAN

This variable enables the blocking of item status changes if the item is on loan. It is relevant for both online and offline item updates.

CHECK-DUP-CALL-NO

This variable enables a check on an item's call number to see if it is already in use.

Possible values are: Y or N.

N = Default. No Item Call Number duplication check is performed.

Y = Item Call Number duplication check is performed. When an item is created or updated with a call number (Z30-CALL-NO-TYPE and Z30-CALL-NO) that is already set for an item of a different ADM record, an error message is displayed and the item is blocked from being saved.

If the call number is already set for an item of the same ADM record, a warning message is displayed, but the action is not blocked.

ITEM-BARCODE-OR-CALL-NO
Disables or enables item retrieval by call number when using the Item Bar (Circulation and Cataloging GUIs).

Possible values are: Y or N.

Y - Item retrieval by call number is enabled (in the Item Bar in Circulation or Cataloging GUI).

N - Item retrieval by call number is disabled (i.e. there is an attempt to retrieve the item by barcode; if it fails, no further attempt is made to retrieve by call number).

The default value is Y.

ITEM-BOR-INFO-CIRC-ONLY

Determines whether or not patron information will be hidden when item information is displayed in the Catalog and Acquisitions modules.

Possible values are Y and N.

Y = Patron Information (such as Patron Name, ID) is only shown in the Circulation module, and not in the Acquisitions and Cataloging modules.

N = Patron Information (such as Patron Name, ID) is shown in all modules.

The default value is N.

ITEM-DATE-FROM-ORDER
Controls the automatic update of the arrival date of monograph orders.

The new flag has the following possible values: 0/1:

0 - Default. The arrival date is not recorded (old functionality).

1 - The arrival date is recorded in the item record (new functionality).This occurs only if the item's arrival date is not populated (Z30-ARRIVAL-DATE=00000000). If the item's arrival date is already populated, the item arrival date is not updated.

Notes:

- This functionality updates all order's related items regardless of the number of units ordered or the number of units arrived and is not related to the **Order completely arrived** checkbox in the Arrival Form or Order Form.

- If an Order arrival record is deleted, the item arrival date is not removed from the item record.

ITEM-FORM-STATISTIC
Determines whether or not the Statistic field is mandatory in the Item form used in the Catalog and ACQ/Serials modules.

Possible values are: Y or N.

N – Default. The item's statistic field (Z30-ITEM-STATISTIC) is not mandatory. An item record can be saved with null in the Statistic field.

Y – The item's Statistic field is mandatory. If the Statistic field is left blank when an item record is saved; a message is displayed requiring the user to populate the Statistic field.

INVENTORY-NUMBER - Z52 Z311 Z313
This variable defines how the item's inventory number is handled.

The default value is Z52.

Z52 - The inventory number is not mandatory and can be left blank. However, if it is used it can be assigned from a system counter. If ? followed by a Z52-sequence is entered (for example ?inventory-number or ?inventory-number-2), an inventory number is allocated by using the relevant Z52-sequence (for example, inventory-number inventory-number-2). If text is entered into this field, the text will be used as the inventory number.

Clicking the **Refresh** button in the Item Form (Cataloging / Items tab / Items List root / lower pane) can be used to allocate a new inventory number.

Z311 - The inventory number is not mandatory and can be left blank. If it is used then it is based on Z311 and on the table tab_z311.

Clicking the **Refresh** button in the Item Form can be used to allocate a new inventory number.

Z313 - The inventory number is not mandatory and can be left blank. If it is used then it is based on Z313 in the following manner :

If a mere question mark (?) is entered in the Inventory Number field, a Z313 record with Z313-USED = N, a blank Z313-SUB-LIBRARY and a blank Z313-SERIES is searched for. Once found, only the field Z313-INVENTORY-NUMBER is copied to Z30-INVENTORY-NUMBER.

If a question mark (?) followed by a series name is entered in the field Inventory Number, a Z313 record with Z313-USED = N, Z313-SUB-LIBRARY = Z30-SUB-LIBRARY and Z313-SERIES = <Series> is searched for. Once found, the combination Z313-SUB-LIBRARY + Z313-SERIES + Z313-INVENTORY-NUMBER is assigned to Z30-INVENTORY-NUMBER.

For example, if ?SER1 is entered in the field Inventory Number for an item of sublibrary UEDUC, and a Z313 record with Z313-USED = N, Z313-SUB-LIBRARY = UEDUC, Z313-SERIES = SER1 and Z313-INVENTORY-NUMBER = 000000095 is found, Z30-INVENTORY-NUMBER will receive the value "UEDUCSER1 000000095".

When clicking on the **Refresh** button in the Item Form (Cataloging / Items tab / Items List root / lower pane), an inventory number is not assigned, in order not to use a Z313 record in vain (for example if **Cancel** is used later on).

It is important to note that once an inventory number has been assigned to an item it can not be changed or reverted.

When an item that has a Z313 based inventory number is deleted, the behavior of the Z313 record depends on the contents of the item's internal note.

If one of the values "recover", "RECOVER" or "Recover" is found there, the Z313 record will be freed. This means that the Z313-USED field will revert to N and the item related fields will be cleared, making this Z313 record re-assignable to any other item.

If none of these values are found in the internal note field, the Z313 record will remain "in use", and only the withdrawal date that will be added to the record will mark that the record's item no longer exists. The Z313 will not be assignable to another item.

Note that in order to allow the correlation of the item record and its related Z313 inventory record, one of the following pairs of triggers should be defined in the ADM library file_list:

- Z313_TRIGGER_1 and Z313_TRIGGER

OR

- Z313_TRIGGER_1 and Z313_TRIGGER_2.

Make sure the triggers are enabled and created (using UTIL A/17/6/2 and UTIL A/17/6/1).

Z313_TRIGGER – Enables the updating of the inventory record each time item data is updated.

Z313_TRIGGER_2 - Directs the system to 'lock' inventory records whoser assign date is up to the defined 'lock' date (inclusive). The 'lock' date parameter (YYYYMMDD) is defined in column 3 of the ADM's library file_list (in the same row that Z313_TRIGGER_2 is set). Note that a library should decide which trigger to enable: Z313_TRIGGER or Z313_TRIGGER_2.

Z313_TRIGGER_1 – Handles the inventory record when the item is deleted. This action depends on the content of the item's internal note field. It supports the option to recover an inventory number of a deleted item or to record the deletion of the item within the inventory record by populating the inventory data, withdrawal date and withdrawal. Z313_TRIGGER_1 can be set with either one of the above triggers, Z313_TRIGGER or Z313_TRIGGER_2.

UPDATE-BARCODE

This variable is used to define whether the item barcode can be updated by the user in the GUI.

Y = Default. Enable barcode data update in GUI.

N = Disable barcode data update in GUI.

UPDATE-Z30-COPY

Determines how the system behaves when an item record is opened from a serials schedule record.

Possible values are: Y or N.

Y = Z16-COPY-SEQUENCE is copied to Z30-COPY-ID when an item record (Z30) is opened for the subscription.

N = Z16-COPY-ID is copied to Z30-COPY-ID when an item record (Z30) is opened for the subscription.

The default value is Y.

X852-ITEM-OVERRIDE
Determines whether or not to update the item location subfields from the 852 field of the linked HOL record.

Possible values are: Y or N.

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.

The default value is Y.

Z30-PRICE-FROM-ORDER
Determines how the estimated and invoiced acquisition price is updated from the item price field.

Possible values are: Y or N.

Y = the item price field is controlled by and automatically updated from the acquisition price (estimated and invoiced).

The default value is N.

12 Item Statistic Options

The Item Statistics information field is included in the General Information (2) tab of the Item Expand pane.

There are three options for entering data in the Item Statistic fields, depending on the client setup:

1. Simple Edit (default) - No arrow on field.
2. Regular - Opens `pc_tab_expand_field` menu.
3. Special Form - Opens a special four-part form, each part of which can have a `pc_tab_expand_field` menu (ITEM-STATISTIC-1 to ITEM-STATISTIC-4). The parts are separated by the delimiter defined in ItemStatisticDelimiter.

The available option is determined in your client's `alephcom/tab/alephcom.ini` file:

```
ItemStatisticField=3
SerialItemStatisticField=3
ItemStatisticDelimiter=/
```

ItemStatisticField refers to the Statistic field in the Item Form.

SerialItemStatisticField refers to the Item Statistic field in the Subscription Information Form.

The digit to the right of each line refers to the three options listed above.

The ItemStatisticDelimiter determines the delimiter that separates the parts of the special form (option 3).

There is an option to set the Item Statistic field (Z30-ITEM-STATISTIC) to be mandatory in the Item form used in the Catalog and ACQ/Serials modules.

To support this, set the following in ./<adm_library>/tab/ tab100:

ITEM-FORM-STATISTIC=Y

This flag determine whether or not item statistic is a mandatory field.

Possible values: Y/N

N – Default. The item's statistic field is not mandatory. An item record can be saved with null in the Statistic field.

Y – The item's Statistic field is mandatory. If the Statistic field is left blank when an item record is saved; a message is displayed requiring the user to populate the Statistic field.

13 Print Templates

13.1 Available Templates

The names of the Items template files are listed below (where nn is a number identifying the version).

Note: When nn is in the range 00 - 19, inside labels will be produced. When nn is in the range 20 - 99, spine labels will be produced.

Items Templates

Template	**Purpose**	**print.ini function name**
Copy Label item-copy-label-nn	Generated from the Cataloging GUI: Items tab/Items List pane, Label button or from the service b-item-03	ItemLabel
Copy Label item-copy-label-nn	Generated from the Cataloging GUI: Items Menu: Print Item Labels	ItemLabelPrint
Issue Label item-issue-label-nn	Generated from the Cataloging GUI/Items tab:Items List pane, Label button or from the service b-item-03	ItemLabel
Issue Label item-issue-label-nn	Generated from the Cataloging GUI: Items Menu: Print Item Labels	ItemLabelPrint

Template	Purpose	print.ini function name
Completed Volumes item-completed-volumes-nn	Completed volumes - ready for Binding	CompletedVolumsPrint

Note that libraries can share the same forms by adding a definition to the path_convert table under the library's tab directory. Your ADM library is probably linked to your BIB library:

```
$usm50_dev/usm50/form_eng      $usm01_dev/usm01/form_eng
```

Services Templates
Items Services

Service	Form
Report of Missing Items (ret-item-02)	missing-items-report.xsl
Shelf Reading Report (item-04)	shelf-reading-report.xsl
Shelf List (item-05)	shelf-list.xsl

14 Items Values of pc_server_defaults

`pc_server_defaults` is an ALEPH configuration table located under `$alephe_root`. It defines the default value for the pc server. Following are the relevant entries for GUI-Items:

default_lock_period
Locked item records are automatically unlocked after a period defined in this entry. The variable is defined in seconds.

Note: The value entered in setenv default_lock_period is shared also by Acquisitions, Circulation and ILL records.

item_label_restore
This entry works together with the GUI-Circulation utility Change Item Information. Possible values:

Y - When the user changes item information (using the Change Item Information utility in GUI-Circulation) and the Permanent Change field is not checked, as soon as the item label is printed, the original item information will be restored and it will replace the non-permanent information. The item label will also include the original item information.

N - The original item information will not be restored when the item label is printed.

serial_bind_period
The maximum number of days for serial issues to arrive after the Expected Arrival Date, after which a volume containing them will be considered complete to appear in the Completed Volume Report.

15 Checksum of Item Barcode

The ADM table, `tab_checksum`, is used for defining the checksum programs. The programs can be used for the creation, validation and correction of item barcodes, patron barcodes, budget code prefixes, vendor/supplier code prefixes and vendor/supplier additional-codes.

The relevant programs for checksum of item barcode are:

- checksum_create_z30_by_seq - for creating item barcode, using UTIL G/2 last-barcode-number

- checksum_create_z30_by_doc_no - for creating item barcode using Z30-REC-KEY
  ADM record number + item sequence number

- checksum_create_codabar - for 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 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_item_bar_keio and checksum_valid_bor_id_keio - site-specific procedures for an application in Keio-Japan, 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_correct_z30_bar_seri - site-specific procedure for an application in Korea, for validation of item barcodes. The following manipulation is done when barcodes are inserted:

  If the barcode is 12 characters length or more, no manipulation is performed.

  If it is less than 12 characters, it is processed according to the following algorithm before actual retrieval:

  (a) If the barcode starts with a digit, it is left-padded with zeros, e.g. "1234" --> "000000001234".

  (b) If the barcode starts with "C", "M" or "S" (see (c) below for "S" with specific prefixes) followed by digits, zeros are added between the alpha prefix and the digits, e.g. "M4567" --> "M00000004567".

  (c) If the barcode starts with one of the following prefixes - "S96", "S97", "S98" or "S99", zeros are added between the prefix and the remaining digits, e.g "S981123" --> "S98000001123".

  (d) Otherwise, no manipulation is performed.

The following example shows how to perform the UEA Item Barcode (valid) Checksum:

```
            1                          2
!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CSV-LOAN-ITEM-BARCODE        chksm_valid_item_barcode_uea
```

16 Complete Volume Report

A parameter defines a maximum number of days for issues to arrive after their Expected Date of Arrival (EDA), after which the volume containing them is considered complete despite their unavailability in the library, and appears in the report.

That parameter is in **alephe/pc_server_defaults**:

```
setenv serial_bind_period
```
The default is set at 50 days.

17 Items Values of aleph_start

aleph_start is an ALEPH configuration file located under $alephe_root. This configuration file defines default values when starting the system. Following is the relevant entry for the Items module:

hvd_call_no_type (site-specific)

This entry determines whether or not there will be a correction of the item's call number type "7" when the call number contains subfield 2 with special characters. This correction is not valid for the database but only for the call number filing procedure (in order to take the correct line from `tab_filing_call_no` and build `z30_call_number_key`). When hvd_call_no_type is set to Y: If the call number type is "7" and the call number field contains subfield 2 (selected uppercase) then the call number type will be overridden to "a" for filing purposes only. When hvd_call_no_type is set to N: the call number type will remain "7" in any case.

Note: The **hvd_call_no_type** parameter is a site-specific feature and is usually set to "N".

18 Item Call Number – Author Number and Chronology Number

This chapter refers to the **Call No.** and **Second Call No.** fields of the GUI-Cataloging-Item Form and their relevant setup and functionality for special "852 Form" (type 6).

The **Call No**. and **Second Call No.** fields can be opened in various modes/forms as explained in the Call Number Field section of *Aleph 20.01 System Librarian's Guide – Items*. To activate the special 852 Form that includes Author Number and Chronology Number, set the value 6 in ItemLocation1Field and or ItemLocation2Field parameters of the PC file 'alephcom.ini'

```
[ExpandField]
ItemLocation1Field=6
ItemLocation2Field=6
```

852 Form of type 6 allows the assigning of "Author Number" and "Chronology Number" for Item Call Number. This is in addition to the other Call number fields: Prefix, Call, Number, Suffix, and Source.

To activate the special 852 Form that includes Author Number (and not Chronology Number), set the value to 7 in the ItemLocation1Field and or the ItemLocation2Field parameters of the PC file 'alephcom.ini'

```
[ExpandField]
ItemLocation1Field=7
ItemLocation2Field=7
```

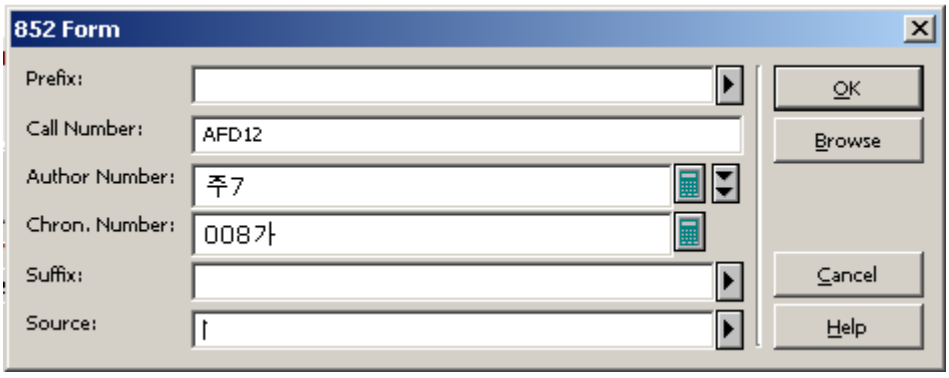
852 Form of type 7 allows the assigning of "Author Number" for Item Call Number. This is in addition to the other Call number fields: Prefix, Call, Number, Suffix, and Source.

- **Author Number Field** – Offers a list of methods according to which the Call Number's Author Number value (852$$i) can be calculated. The methods that are avail for Korean titles (Hangul and Hanja) are Lee Jai-Chul (various types), Elrod, and Jang Il Sei. The methods that are avail for non- Korean titles (e.g. Latin material) are Cutter Sanborn, Cutter Three-Figure, and Cutter Three Figure without Caret Mark.

- **Chronology Number** – Allows the option to calculate the Call Number's Chronology Number ($$i) that contains the year Number part and a sequence.

18.1 The Special "852 Form" (Call Number Form - Type 6)

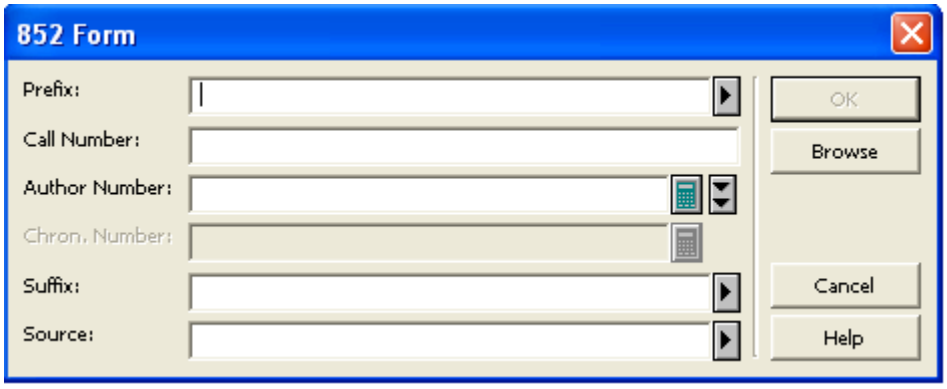
At GUI-Cataloging-Item Form; clicking the button next to the Item Call Number field displays the following form:



The 852 Form is one of the options for recording location (call number) information in the Item record. The six fields which make up this form are based on the MARC21 Standard for Holdings Records as follows (in brackets: tag subfield name in MARC21): Prefix, Call Number, Author Number, Chronology Number, Suffix, and Source.

The Special "852 Form" (Call Number Form - Type 7)

From GUI-Cataloging-Item Form, click the button next to the Item Call Number field to display the following form:



Note that this form is the same as form type 6, except of the Chronology Number field which is disabled.

The following is a description of each field of the 852 form:

Prefix

Call number prefix ($$k) can be manually assigned by the staff user. Users may select the relevant value from a drop-down list which is set in the LOCATION-PREFIX menu of ./xxx50/tab/pc_tab_exp_field_extended.lng.

Call Number
Item classification number ($$h). The Item Call Number can be automatically assigned by the system based on BIB information or manually assigned by the staff user. For more details, see tab_author_num.conf.

Author Number
The Call Number's Author Number part ($$i). This field is a combo-box containing the Author Number calculation methods. The content of the combo-box list is driven from ITEM-PART-1 menu of./xxx50/tab/pc_tab_exp_field_extended.lng.
When the 852 Special Form is opened, the **Author Number** field can be automatically generated by the system or manually assigned by the staff user. This behavior is depended on ./xxx01/tab/tab_author_num.conf (see details in "tab_author_num.conf" section).
If the Author number value is already filled-in, the user may re-select one of the methods. The recently calculated value overrides the current value.
Note that user may activate the button which is located to the right of the **Author Number** field and override the default system behavior which is defined in tab_author_num.conf. Clicking this button displays a list of BIB tags for calculating Author Number and can be used if the operator decides to use an alternative tag for calculation the **Author Number** field (not the tags defined in tab_author_num.conf).
The **Author Number** field may also be left blank. The following are the available Author Number Methods:

- **Lee Jai-Chul** – Lee Jai Chul method that can be applied for Korean material (Hangul and Hanja). The algorithm that is used to calculate the Lee Jai Chul author number consults the setup table: in ./xxx01/tab/tab_lee_jai_chul. The Lee Jai Chul method can have multiple entries in the Author Number combo-box (as defined in ITEM-PART-1 menu of ./xxx50/tab/pc_tab_exp_field_extended.lng).

- **Elrod** – Elrod method that can be applied for Korean material. For more details, see the Author Number by Elrod section.
- Jang Il Sei – Jang Il Sei method that can be applied for Korean material. For more details, see **Author** Number by Jang Il Sei on page 39.
- **Cutter Sanborn** – Cutter Sanborn method that can be applied for Latin and non-Korean material. The algorithm that is used to calculate the Cutter Sanborn author number consults the setup table: in ./xxx01/tab/tab_cutter.

- **Cutter Three-Figure** – Cutter Three Figure method that can be applied for Latin and non-Hangul/Hanja material. The algorithm that is used to calculate Cutter Three Figure author number consults the setup table in ./xxx01/tab/tab_cutter_three. Note that if the cutter-three-figure is calculated by the system (either by automatically assignment or by selecting the routine from the drop-down list) and the chronology number is automatically assigned by the system the system adds a ^ sign to the end of the cutter-three-figure value. This is done for display purposes so that a space is displayed between the two $$i fields (Author Number by cutter-three-figure and Chronology Number).
- **Cutter Three-Figure without Caret Mark –** the same as a regular cutter three figure, without a caret mark at the end of the field.

Chronology Number

The Call Number's Chronology Number part ($$i). This field is a combo-box containing one option, Chronology Number. The content of the combo-box list is driven from ITEM-PART-2 menu of./xxx50/tab/pc_tab_exp_field.lng. When the '852 Special Form' is opened, the 'Chronology Number' field can be automatically generated by the system or manually assigned by the staff user. This behavior is depended on ./xxx01/tab/tab_author_num.conf (see details in "tab_author_num.conf" section). If the Chronology Number value is already filled-in, the user may select the Chronology Number from the list; the value just calculated overrides the current value. The Chronology Number field may also be left blank. The **Chronology Number** field is cleared if one or more of the following 852 Form fields are changed, Prefix, Call Number, and Author Number. For technical reasons related to the Call Number sorting routine, the original year (the year value before calculating the chronology number) are recorded in $$9 of Z30-CALL-NO/ Z30-CALL-NO-2. If both fields **Author Number** and **Chronology Number** are filled-in, two $$i entries are created in the Z30-CALL-NO/ Z30-CALL-NO-2. For more details, see the section, Chronology Number Algorithm.

Suffix

Call number suffix ($$m). Can be manually assigned by the staff user. Users may select the relevant value from a drop-down list which is set in LOCATION-SUFFIX menu of ./xxx50/tab/pc_tab_exp_field_extended.lng.

Source

Call number source ($$2). Can be manually assigned by the staff user. Users may select the relevant value from a drop-down list which is set in LOCATION-SOURCE menu of ./xxx50/tab/pc_tab_exp_field_extended.lng.

NOTES:

Note 1:

When the 852 Form is opened for an item which has no value in the Call Number field (Z30-CALL-NO or Z30-CALL-NO-2 is empty), the system fills in values as defined in the relevant configuration settings (see tab_author_num.conf for configuration information).

When a filled 852 **Form** is opened, the system displays the current values without automatic recalculating the empty fields. If you want to update or re-calculate the values, type or select a relevant scheme. If the prefix, call number, or author number is changed, the chronology number is cleared.

Note 2:

To support the GUI drop-down list for the **Author Number** field, set the following in ITEM-PART-1 menu of ./xxx50/tab/ pc_tab_exp_field_extended.lng: ITEM-PART-1:

```
ITEM-PART-1          ##### L Lee Jai Chul 1          LEE-JAI-CHUL-1
ITEM-PART-1          ##### L Lee Jai Chul 2          LEE-JAI-CHUL-2
ITEM-PART-1          ##### L Lee Jai Chul 3          LEE-JAI-CHUL-3
ITEM-PART-1          ##### L Lee Jai Chul 4          LEE-JAI-CHUL-4
```

```
ITEM-PART-1          ##### L Lee Jai Chul 5            LEE-JAI-CHUL-5
ITEM-PART-1          ##### L Lee Jai Chul 6            LEE-JAI-CHUL-6
ITEM-PART-1          ##### L Lee Jai Chul 7            LEE-JAI-CHUL-7
ITEM-PART-1          ##### L Lee Jai Chul 8            LEE-JAI-CHUL-8
ITEM-PART-1          ##### L Elrod                     ELROD
ITEM-PART-1          ##### L Jang Il Sei               JANG-IL-SEI
ITEM-PART-1          ##### L Cutter Sanborn            CUT-SANBORN
ITEM-PART-1          ##### L Cutter Three Figure        CUT-THREE
ITEM-PART-1          ##### L Cutter Three without ^    CUT-THREE-CM
```

ITEM-PART-1 menu for: LEE-JAI-CHUL-<N> can have few entries depending on the procedure type that the library uses: LEE-JAI-CHUL 1 thru LEE-JAI-CHUL-8. Those types must be defined in ./xxx01/tab/tab_lee_jai_chul table.

Note 3:

To support the GUI drop-down list for 'Chronology Number' field, set the following in ITEM-PART-2 menu of ./xx50/tab/ pc_tab_exp_field.lng

```
ITEM-PART-2              L Chronology Number           CHRON
```

Note 4:

To support the GUI drop-down list of BIB tags which are used as an alternative tags for Author Number calculation, set ITEM-PART-1-TAG menu of ./xxx50/tab/ pc_tab_exp_field.eng. For example:

```
ITEM-PART-1-TAG          L 600                    600
ITEM-PART-1-TAG          L 610                    610
ITEM-PART-1-TAG          L 111                    111
```

18.2 tab_author_num.conf

The bib library tab_author_num.conf (./xxx01/tab/tab_author_num.conf) defines the library's policy for assigning the following Item Call Number fields of the special 852 Form: Item Cal Number, Author Number and Chronology Number.

Different policies can be set per sublibrary. This can be achieved by setting the sublibrary code as the extension of the table file name. For example, to define policy for sublibrary code “law", the following file name should be set: tab_author_num.conf.law.
The relevant table is called by the system according to the item's sublibrary code. If there is no table with the relevant sublibrary code OR the item is not assigned yet with sublibrary, the default table, tab_author_num.conf, is used.

Sample of tab_author_num.conf

```
[CALL-NO]
LAW = 090##,a,1,056##,a,1,082##,a,2
WID = 050##,a,2,052#1,a,1,090##,a,3
##### = 050##,a,2,052#1,a,1,090##,a,9

[AUTHOR-NAME]
fields = 100,110,111,245

[LATIN-TYPE]
type = 1

[HANGUL-TYPE]
type = 8

[COMMON-KOREAN]
common_words = ae40,c774,bc15,cd5c,c815,c870,ac15,c7a5,d55c,c724

[SPECIAL-LETTER-VALUE]
special_letter = 3131


[CHRON-TYPE]
type = 1
sequence = korean
```

[CALL-NO] section

Define the BIB fields that are applied in order to build the Call Number ($$i of Z30-CALL-NO / Z30-CALL-NO-2) and Call Number type (Z30-CALL-NO-TYPE/ Z30-CALL-NO-TYPE-2). For example:

WID = 050##,a,2,052#1,a,1,090##,a,9

“WID” is the item sublibrary code, BIB tag 050##$$a is a first candidate tag to populate $$i of the item call number. If 050##$$a is used for populating the call number, then the value "2" is set in the Call Number type. If the BIB record has no 050##$$a, the system will keep checking the next option: 052#1,a,1 and so on.

Note that in case you define of tab_author_num per sublibrary (e,g : tab_author_num.wid) or in case where all sublibrary use the same algorithm; you may set "match all" line. For example:

= 050##,a,2,052#1,a,1,090##,a,9

[AUTHOR-NAME] section

Define the BIB fields that are applied in order to determine whether the BIB record is Korean (Hangul or Hanja) or Else. For example:

AUTHOR-NAME = 100,110,111,700,710,711

The system first looks for 100, then for 110, and then 111, and so on, until it finds a matching field.

Note that user may override the BIB value defined in AUTHOR-NAME of tab_author_num.conf by activating the button which is located at the most right side of the Author Number field (at the "852 Form"). This button offers list of alternative BIB tags for calculating the Author Number.

[LATIN-TYPE] section

Use for generating Author Number field for non-Korean material (e.g.: Latin). For example:
type = 1

Values can be:
0 = Author Number of Item Part fields is displayed null (it is not automatically generated by the system). User may select the relevant Author Number method from the Author Number drop-down list or manually type in a value.
1= upon opening the "852 Form" the Author Number field is populated with Cutter Sanborn method. User may override the automatically filled value.
2= upon opening the "852 Form" the Author Number field is populated with Cutter three-figure method. User may override the automatically filled value.
3= upon opening the "852 Form" the Author Number field is populated with Cutter three-figure without caret mark method. You can override the automatically filled value.

[HANGUL-TYPE] section

Use for generating Author Number field for Korean (Hangul and Hanja) material. For example:
type = 4

Values can be:
0 = Author Number of Item Part fields is displayed null (it is not automatically generated by the system). User may select the relevant Author Number method from the Author Number drop-down list or manually type in a value.
1 thru 8 = upon opening the 852 Form, the **Author Number** field is populated according to the Lee Jai-Chul method according to the routine type as defined in ./xxx01/tab/tab_lee_jai_chul. User may override the automatically filled value by selecting other method from the combo-box list or by manually type in a value.
9 = upon opening the 852 Form, the Author Number field is populated with Elrod method. User may override the automatically filled value.
a= upon opening the 852 Form, the Author Number field is populated with Jang Il Sei method. User may override the automatically filled value.

[COMMON-KOREAN] section

Used by the system in case where the Author Number is assigned using one of the Lee Jai Chul types.
It is used to define the commonly used Korean names. For example:
common_words = ae40,c774,bc15,cd5c,c815,c870,ac15,c7a5,d55c,c724

[SPECIAL-LETTER-VALUE] section

Used by the system in case where the Author Number is assigned using Lee Jai Chul types 5 or 6 or 8. It is used to define the exception letter that should be handled differently. For example:

special_letter = 3131

[CHRON-TYPE] section

Used by the system to to define policy for assigning the Chronology Number. This section has two parameters: "sequence" and "type". For example:

type = 1
sequence = korean

The parameter **type** defines whether the library wishes the **Chronology** field to be automatically populated when the 852 Form is opened.
Types values can be "0" or "1".
0 – Chronology Number is displayed null (it is not automatically generated by the system). User may manually select the "Chronology Number" option from the drop-down list or type it in.
1 – Upon opening the 852 Form, the Chronology Number field is automatically populated based on the Chronology Number algorithm. User may override the automatically filled value.
The sequence parameter defines according to which sequence list the chronology number sequence part is set: Korean sequence list or Latin sequence list.
Values can be "korean" or "latin".

18.3 Author Number and Chronology Number: Methods and Algorithms

18.3.1 Author Number by Cutter Three Figure Method

The cutter three figures method is applied for non-Korean titles using the table: ./xxx01/tab/tab_cutter_three table.

To set the automatic assignment of cutter three figures to non-Korean titles, set the following in table ./xxx01/tab/ tab_author_num.conf:

[LATIN-TYPE]
type = 2

18.3.2 Author Number by Cutter Three Figure Without Caret Mark Method

The cutter three figures without caret mark method is applied for non-Korean titles using the table: ./xxx01/tab/tab_cutter_three table.

To set the automatic assignment of cutter three figure without caret mark to non-Korean titles, set the following in table ./xxx01/tab/ tab_author_num.conf:

[LATIN-TYPE]
type = 3

18.3.3 Author Number by Cutter Sanborn Method

The cutter sunburn method is applied for non-Korean titles using the table: ./xxx01/tab/tab_cutter.

To set the automatic assignment of cutter sanborn to non-Korean titles, set the following in table ./xxx01/tab/ tab_author_num.conf:

[LATIN-TYPE]
type = 1

18.3.4 Author Number by Lee Jai Chul Method

The Lee Jai Chul method is applied for Korean titles using the table: ./xxx01/tab/tab_lee_jai_chul.

Each library can define according to which of the eight existing mapping tables the Lee Jai Chul number will be automatically calculated. The definition is specified in HANGUL-TYPE section of the table ./xxx01/tab/ tab_author_num.conf. For example:

[HANGUL-TYPE]
type = 5
type 1 thru 8 are dedicated to Lee Jai Chul various types.

18.3.5 Author Number by Elrod

The Elrod method is applied to Korean titles, regardless of MARC formats KORMARC or MARC21 and material types Book, Serials, etc.
To set the automatic assignment of the Elrod method for Korean titles, set the following in table ./xxx01/tab/tab_author_num.conf:

[HANGUL-TYPE]
type = 9

Elrod author number consists of two parts delimitated by a single space: the Author Number part and the Title Number part. For example, if Author number part is XXX and Title number part is Y, the author number by Elrod is 'XXX Y'.

Author Number part of Elrod

Author Number part is up to three Korean syllables which are extracted from the $$a of the following author heading fields in the BIB record. If an author name is less than 3 syllables, the system extracts it as it is. If an author name consists of 2 Korean syllables only in length, just the 2 syllables should be extracted.

The BIB record is checked for Author heading fields as defined in AUTHOR-NAME section of tab_author_num.conf.
For example:
AUTHOR-NAME = 100,110,111,700,710,711
The system first looks for 100 and then for 110 and then 111, and so on, until it finds a matching field.

Example (the Capital letters represent Korean syllables):

When 100 and 700 fields exist
100 1^ $$aABCD → “ABC”
700 1^ $$aEFG

When an author name is less than 3 characters
100 1^ $$aAB → “AB”

When no IXX fields exist
710 1^ $$aABCD → “ABC”

(An example of the real Korean letters)
100 1^ $$a모현주 → “모현주”

If the relevant author name in the BIB record is in Chinese letters, the systems extracts three letters as an Author Number part from the AUTHOR-NAME field and then transliterate it to Korean letters. The transliteration from Chinese to Korean, uses the fix_doc program named “fix_doc_transliteration” with the program argument “HANJA_TO_HANGUL”.

If the Chinese author name is XXX and its translation into Korean is YYY; the result is YYY.

100 1^ $$a三浦綾子 → “삼포능”, instead of “三浦綾”

Title Number part of Elrod

Title number part is one Korean letter which is extracted from the 245 $$a in the BIB record.

Examples (the capital letters represent Korean syllables):

100 1^ $$aABCD
245 1^ $$aEFG /$$c by ABCD → “E”

An example of the Korean letters:
100 1^ $$a모현주
245 1^ $$a한국경제 /$$d 모현주 저 → “ 한”

If 245 $$a main title starts with a pre-title put in parentheses, the system ignores the pre-title with parentheses and then take the very next letter.

245 1^ $$a(ABC)DEFG → “D”

If 245 $$a main title is in Chinese letters, the system transliterates it to Korean letters.
If the Chinese main title starts with X and its translation into Korean is Y, the result is Y.

245 1^ $$a**韓**國經濟→“ 한”，instead of“ 韓”

245 1^ $$a(韓國)**經**濟→“ 경”，instead of“ 經”

If 245 $$a main title starts with in Roman or Numeric characters, the system takes it as it is. Staff users change it to the correct number.

Note:

Elrod algorithm uses a special entry in column 1 of tab_character_conversion_line: CALL_NO_ELROD. The library may define in column 5 of tab_character_conversion_line any character conversion table that should be used for CALL_NO_ELROD.

For example: It is possible to set a special character conversion table for CALL_NO_ELROD including the values from hanja_to_hangul and values from kana2hangul.

The following is an example of ./alephe/unicode/tab_character_conversion_line that defines cjk_to_hangul table for CALL_NO_ELROD:

```
CALL_NO_ELROD        ##### # line_utf2line_utf          cjk_to_hangul
```

The ./alephe/unicode/cjk_to_hangul table must be set to define the Unicode values to be used by CALL_NO_ELROD routine.

18.3.6 Author Number by Jang Il Sei

The Jang Il Sei method is applied to Korean titles, regardless of MARC formats KORMARC or MARC21 and material types Book, Serials, etc.
To set the automatic assignment of the Jang Il Sei method for Korean titles, set the following in table ./xxx01/tab/tab_author_num.conf:

```
[HANGUL-TYPE]
type = a
```

Jang Il Sei Author Number consists of two sequential parts: the Author Number part and the Title Number part. For example, if the Author Number part is XXX and Title Number part is Y, the Author Number by Jang Il Sei is 'XXXY'.

Author Number Part of Jang Il Sei
The Author Number part consists of the first Jamo (consonant) of the first syllable of the Author (defined in AUTHOR-NUMBER of `xxx01/tab/tab_author_num.conf`) and the corresponding numbers of the author. The system consults only the `xxx01/tab/ tab_jang_il_sei` table in order to calculate the two parts.

Title Number Part of Jang Il Sei
The Title Number part is taken from the first Jamo (consonant) of the first character (syllable) of the Title field(245$$a). The system consults `./alephe/unocode tab_hangul_syllable_to_letters` in order to break the Hangul syllable into two or three "atom" letters.

Note:

The Jang Il Sei algorithm uses a special entry in column 1 of tab_character_conversion_line: **CALL_NO_JANGILSEI**. The library may define in column 5 of tab_character_conversion_line any character conversion table that should be used for **CALL_NO_JANGILSEI**.

For example: It is possible to set a special character conversion table for **CALL_NO_JANGILSEI** including the values from hanja_to_hangul and values from kana2hangul.

The following is an example of ./alephe/unicode/tab_character_conversion_line that defines cjk_to_hangul table for **CALL_NO_JANGILSEI**:

```
CALL_NO_JANGILSEI      ##### # line_utf2line_utf
cjk_to_hangul
```

The ./alephe/unicode/cjk_to_hangul table must be set to define the Unicode values to be used by **CALL_NO_JANGILSEI** routine.

18.3.7 Chronology Number Algorithm

Chronology number consists of two parts: the Year Number part and the Sequence Number part. There is no space between the two parts.

Year Number Part of Chronology

Based on the publication year in the BIB record tag 008 positions 07-10 (or tag 100 positions 09-12 in UNIMARC), the system assigns the Year Number as follows.

- If the publication year is 1900 or before 1900, "00" are assigned.
  1900 → 00
  1854 → 00

- If the publication year is between 1901 and 1999, the last two numbers of the year are assigned.
  1901 → 01
  1902 → 02
  1970 → 70
  1999 → 99

- If the publication year is 2000 or after 2000, the last three numbers of the year are assigned.
  2000 → 000
  2001 → 001
  2007 → 007

Sequence Number Part of Chronology

The numbering for the sequence part of the Chronology Number can be differentiated depending on the library preferred sequence language counter, Latin or Korean. In order to determine to which sequence list to apply, the system consults the parameter "sequence" of the CHRON-TYPE section in tab_author_num.conf.
The parameter "sequence" can be "korean" (system applies to the Korean sequence list) or "latin" (system applies to the Latin sequence list).

In case where:
sequence = latin
The sequence number should be assigned in alphabetical order basically. The first turn is from a to z, the second turn is from za to zz, the third turn is from zza to zzz, and then the following tune is in the same way.

In case where:
sequence = korean
The sequence number is not based on the serial number. It is assigned according to the predefined order of Hangul syllables which consist of the Hangul consonant and the Hangul vowel only.

Examples of the predefined order of Hangul syllables:

가 [U+AC00]
갸 [U+AC38]
거 [U+AC70}
겨 [U+ACA8]
"omitted"
히 [U+D788]

Note that there are total of 140 Hangul syllables which are set in ./alephe/unicode/tab_hangul_syllable_seq

The first tune is based on the order of the above Hangul syllables. If the order is reached the last Hangul syllable, that is "히", the following sequence numbers are expanded merely based on the serial numeric number that should be attached to the last Hangul syllable "히", like "히1", "히2", "히3" ..."히n".

Base of counting (How to check the last number)

In order to assign new Sequence Number, the system checks the last number of it. The base of the counting of Sequence Number is not simply the Year Number. It should be:
"Prefix (if exists) + Classification number + other item number (if exists) + Year Number"

Example 1:
Marc format: MARC21 (Latin)
Published year in 008/07-10: 2007
Prefix: Ref
Classification number: 330
Year Number will be generated by Chronology method: 007
Sequence Number will be generated by Chronology method:

- The base of the counting of Sequence Number is "Ref 330 007" [Prefix + Classification number + Year Number]
- In order to see he last number of Sequence Number, the system should check the call number indexes starting with the above base "Ref 330 007".
- If the last call number which has the base of "Ref 330 007" is "Ref 330 007c", the last Sequence Number is "c".

The following is a sample of a call number browsing list around "Ref 330 007":
.

Ref 330 006히
Ref 330 006히1
Ref 330 007가
Ref 330 007갸
Ref 330 007a

Ref 330 007b
Ref 330 007c
Ref 330 008a
Ref 330 008b

Therefore, new Sequence Number is “d”, based on the MARC21 way.
Eventually, the whole Chronology number will be “007d”.

Example 2:
Marc format: KORMARC (Korean)
Published year in 008/07-10: 2007
Prefix: Ref
Classification number: 330
Year Number is generated by Chronology method: 007
Sequence Number is generated by Chronology method:

- The base of the counting of Sequence Number will be “Ref 330 007” [Prefix + Classification number + Year Number]
- In order to see he last number of Sequence Number, the system should check the call number indexes starting with the above base “Ref 330 007”.
- If the last call number which has the base of “Ref 330 007” is “Ref 330 007갸”, the last Sequence Number is “갸”.

Followings are Call number browsing list around “Ref 330 007”:

..
Ref 330 006히
Ref 330 006히1
Ref 330 007가
Ref 330 007갸
Ref 330 007a
Ref 330 007b
Ref 330 007c
Ref 330 008a
Ref 330 008b

Therefore, new Sequence Number is “거”, based on the KORMARC way. Eventually, the whole Chronology number will be “007거”.

18.4 Other Related Tables

The following is a list of Unicode tables that are consulted by the system when applying Author Number and Chronology sequence:

- . /alephe/unicode/hanja_to_hangul_dictionary_utf8.txt
- ./alephe/unicode/hangul_to_hanja_dictionary_utf8.txt
- ./alephe/unicode/ tab_hangul_syllable_to_letters
- ./alephe/unicode/tab_hangul_syllable_seq

19 Barcode Checksum Routines for Keio

Special for Keio (Japan) – checksum algorithms can be applied for patron and item barcode validation.

The checksums can be implemented using ./xxx50/tab/tab_checksum.

The Keio checksum routines to be set in the 2nd column of tab_checksum are:

- `checksum_valid_item_bar_keio` – Item validation algorithm
- `checksum_valid_bor_id_keio` – Patron validation algorithm

Note: When setting up the table with above algorithms, make sure there are no item barcode creation routines defined in col.1 (routines that starts with CSCR-), such routines lead to a conflict with the new Keio-item validation check.

Setting up tab_checksum with the Keio checksum algorithms:

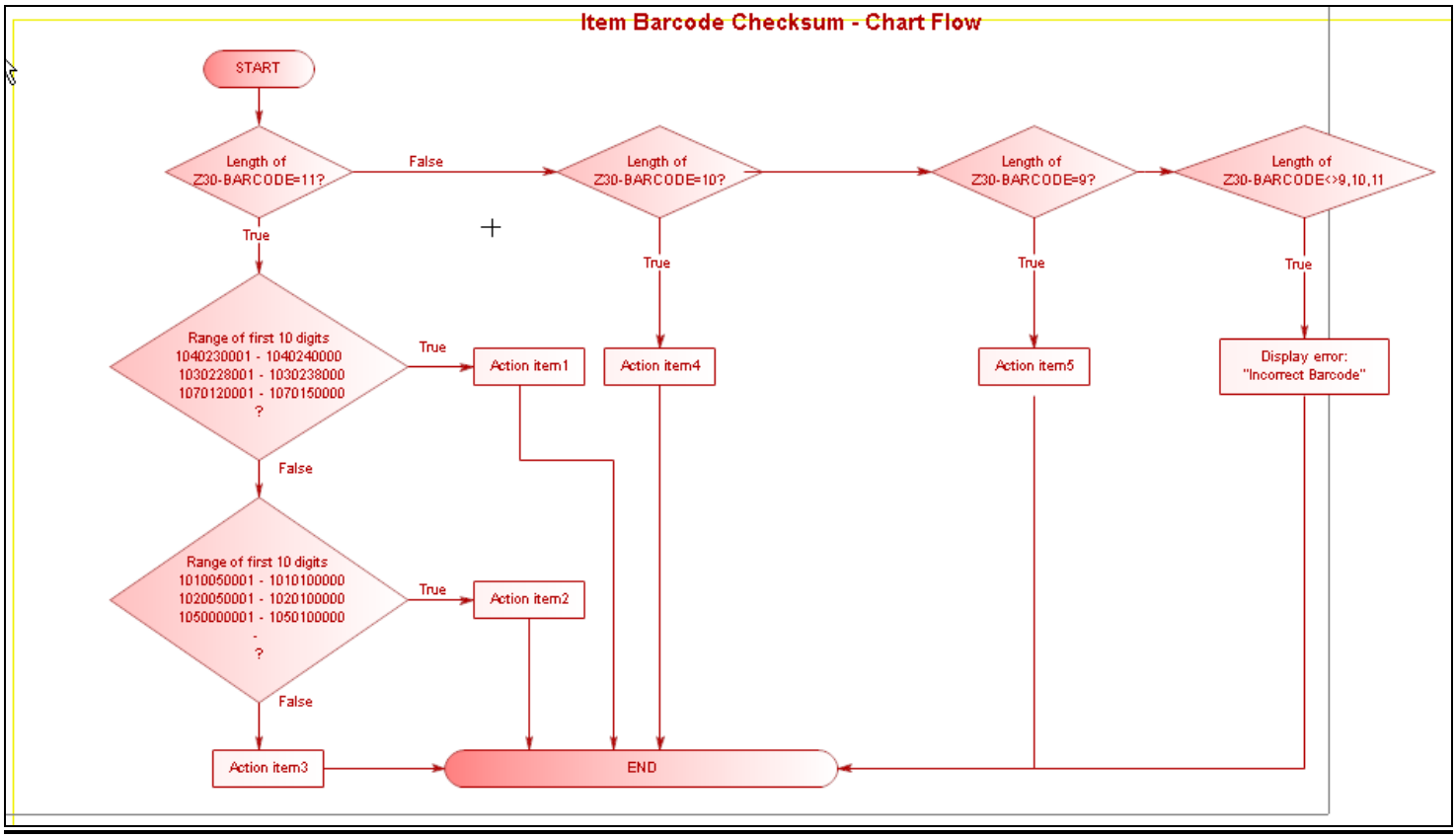
COL 1. - Routine name;

COL 2. - Program name (algorithm) ;

```
!     1                                    2
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CSV-Z30-BARCODE                checksum_valid_item_bar_keio
CSV-LOAN-ITEM-BARCODE          checksum_valid_item_bar_keio
CSV-LOAN-BOR-ID                checksum_valid_bor_id_keio
```

19.1 Item Barcode Algorithm - checksum_valid_item_bar_keio

The following chart describes the algorithm for checking item barcode: checksum_valid_item_bar_keio.
Note that the item barcode is expected to contain a sequence of digits. Only if the barcode length is 10 can the last digit (checksum digit) be X. Otherwise, characters that are not digits produce the error message "Invalid Barcode".



The following is a description of each action shown in the chart above:

Action item 1

1. Multiply each digit starting from the left by 10-9-8-7-6-5-4-3-2-1, and add all the sums.
2. Divide the sum from step 1 by 11 and calculate the remainder.
3. Subtract the remainder from step 2 from 10. This sum is the check digit. (However, if the sum is 10, check digit is be 0.)

For example:
For 1040230001

1 x 10 = 10
0 x 9 = 0
4 x 8 = 32
0 x 7 = 0
2 x 6 = 12
3 x 5 = 15
0 x 4 = 0
0 x 3 = 0
0 x 2 = 0
1 x 1 = 1

Sum 70
70 / 11 = 6 remainder 4
10 - 4 = 6 (10 is 0)
Check Digit = 6 (10402300016)

If the barcode does not match the result of checksum algorithm, the following error message is displayed: “incorrect barcode”.

Action item 2

1. Multiply each digit starting from the left by 1-2-1-2-1-2-1-2-1-2.
   When the sum is a 2-digit number, divide it into each digit, and add all the sums.
2. Divide the sum from step 1 by 10 and calculate the remainder.
3. Subtract the remainder from step 2 from 10. This sum is the check digit.
   (However, if the sum is 10, check digit is 0.)

For example:
For 1010050001

1 x 1 = 1 -> 1
0 x 2 = 0 -> 0
1 x 1 = 1 -> 1
0 x 2 = 0 -> 0
0 x 1 = 0 -> 0
5 x 2 = 10 -> 1 + 0 = 1
0 x 1 = 0 -> 0
0 x 2 = 0 -> 0
0 x 1 = 0 -> 0
1 x 2 = 2 -> 2

Sum 5
5 / 10 = 0 remainder 5
10 - 5 = 5 (10 is 0)
Check Digit = 5 (10100500015)

If the barcode does not match the result of the checksum algorithm, the following error message is displayed: “incorrect barcode”.

Action item 3

1. Multiply each digit starting from the left by 11-10-9-8-7-6-5-4-3-2, and add all the sums.
2. Divide the sum from step 1 by 11 and calculate the remainder.
3. Subtract the remainder from step 2 from 11. This sum is the check digit.
   (However, if the sum is 10 or 11, check digit is 0.)

For example:
For 8010000007

8 x 11 = 88
0 x 10 = 0
1 x 9 = 9
0 x 8 = 0
0 x 7 = 0
0 x 6 = 0
0 x 5 = 0
0 x 4 = 0

0 x 3 = 0
7 x 2 = 14

Sum 111
111 / 11 = 10 remainder 1
11 - 1 = 10 (10 or 11 is 0)
Check Digit = 0 (80100000070)

If the barcode does not match the result of the checksum algorithm, the following error message is displayed: “incorrect barcode”.

Action item 4

The barcode should match one of the following four conditions:

1. Multiply each digit starting from the left by 10-9-8-7-6-5-4-3-2, and add all the sums.
2. Divide the sum from step 1 by 11 and calculate the remainder.
3. Subtract the remainder from step 2 from 11. This sum is the check digit.
   (However, if the sum is 11, the check digit is 0, and if the sum is 10, check digit is "X").

For example:
123456789
1 x 10 = 10
2 x 9 = 18
3 x 8 = 24
4 x 7 = 28
5 x 6 = 30
6 x 5 = 30
7 x 4 = 28
8 x 3 = 24
9 x 2 = 18

Sum 210
210/ 11 = 19 remainder 1
11 - 1 = 10 (10 is X)
Check Digit = X (123456789X)

If the barcode does not match the result, perform the **next check**:

1. Multiply each digit starting from the left by 9-8-7-6-5-4-3-2-1, and add all the sums.
2. Divide the sum from step 1 by 11 and calculate the remainder.
3. Subtract the remainder from step 2 from 11. This sum is the check digit.
   (However, if the sum is 11 or 10 or 0, check digit is 0.)

For example:
For 123456789

1 x 9 = 9
2 x 8 = 16
3 x 7 = 21
4 x 6 = 24
5 x 5 = 25
6 x 4 = 24
7 x 3 = 21
8 x 2 = 26
9 x 1 = 9

Sum 165
165/ 11 = 15 remainder 0
11 - 0 = 11 (11 is 0)
Check Digit = 0 (1234567890)

If the barcode does not match the result, perform the **next check**:

1. Multiply each digit starting from the left by 10-9-8-7-6-5-4-3-2, and add all the sums.
2. Divide the sum from step 1 by 11 and calculate the remainder.
3. Subtract the remainder from step 2 from 11. This sum is the check digit. (However, if the sum is 11 or 10 or 0, check digit is 0.)

For example:
For 123456789

1 x 10 = 10
2 x 9 = 18
3 x 8 = 24
4 x 7 = 28
5 x 6 = 30
6 x 5 = 30
7 x 4 = 28
8 x 3 = 24
9 x 2 = 18

Sum 210
210/ 11 = 19 remainder 1
11 - 1 = 10 (10 is 0)
Check Digit = 0 (1234567890)

If the barcode does not match the result, perform the **next check**:

1. Multiply each digit starting from the left by 9-8-7-6-5-4-3-2-1, and add all the sums.
2. Divide the sum from step 1 by 11 and calculate the remainder.
3. Subtract the remainder from step 2 from 11. This sum is the check digit. (However, if the sum is 11 or 10 or 1 or 0, check digit is 0.).

For example:
For 123456789

1 x 9 = 9
2 x 8 = 16
3 x 7 = 21
4 x 6 = 24
5 x 5 = 25
6 x 4 = 24
7 x 3 = 21
8 x 2 = 26
9 x 1 = 9

Sum 165
165/ 11 = 15 remainder 0
11 - 0 = 11 (11 is 0)
Check Digit = 0 (1234567890)

If the barcode does not match the result of the checksum algorithm, the following error message is displayed: “incorrect barcode”.

Action item 5

1. Divide the 8-digit number into each digit and add them all to get the sum.
2. Divide the sum from step 1 by 10 and calculate the remainder.
3. Subtract the remainder from step 2 from 10. This sum is the check digit. (However, if the sum is 10, check digit is 0.).

For example:
For 12345678

1
2
3
4
5
6
7
8

Sum 36
36/ 10 = 3 remainder 6
10 - 6 = 4
Check Digit = 4 (123456784)

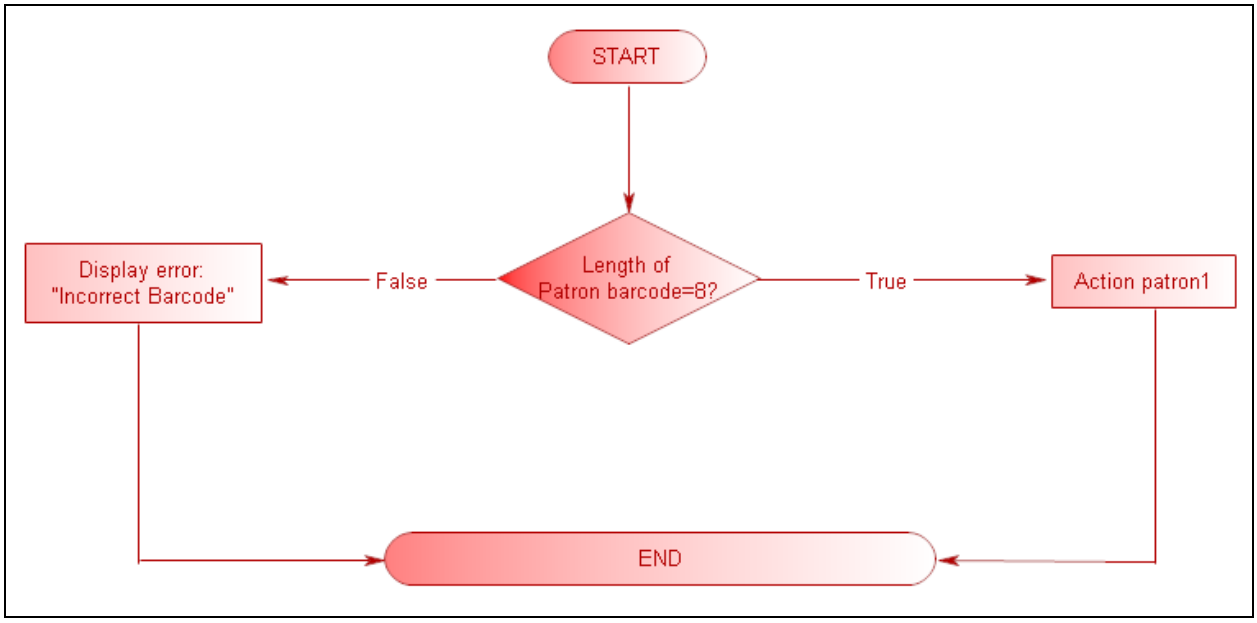
If the barcode does not match the result of the checksum algorithm, the following error message is displayed: “incorrect barcode”.

19.2 Patron Barcode Algorithm - checksum_valid_bor_id_keio

The following chart describes the algorithm, checksum_valid_bor_id_keio, used for checking patron system numbers.

.

As in all other patron barcode check routines, the patron checksum is applied to all patron keys (Z308-KEY-TYPE) **except** of the patron ID (Z308-KEY-TYPE=00).



The following is a description of the action shown in the chart above:

Action patron 1

The barcode should pass one of the following checks:

1. Multiply each digit starting from the left by 9-8-7-6-5-4-3, and add all the sums.
2. Divide the sum of step 1 by 11 and calculate the remainder.
   This remainder is the check digit.
   (However, if the remainder is 10, the check digit is 0.)

For example:
For 1234567

1 x 9 = 9
2 x 8 = 16
3 x 7 = 21
4 x 6 = 24
5 x 5 = 25
6 x 4 = 24
7 x 3 = 21

Sum 140
140/ 11 = 12 remainder 8
Check Digit = 8 (12345678)

If the barcode does not match the result, perform the following check:

1. Multiply each digit starting from the left by 9-8-7-6-5-4-3, and add together all the results.
2. Divide the result of step 1 by 11 and calculate the remainder.
3. Subtract the remainder derived in step 2 from 11. The result is the check digit.
   (However, if the result is 11, the check digit is 0. If the result is 10, the check digit is "X".)

For example:
For 1234567

1 x 9 = 9
2 x 8 = 16
3 x 7 = 21
4 x 6 = 24
5 x 5 = 25
6 x 4 = 24
7 x 3 = 21

Sum 140
140/ 11 = 12 remainder 8
11 - 8 = 3
Check Digit = 3 (12345673)

If the barcode does not match the result, perform the following check:

1. Add the digits of the 7-digit number together.
2. Take the ones place digit from the result of step 1 and subtract it from 10. The result is the check digit.

For example:
For 1234567

1
2
3
4
5
6
7

Sum 28
ones place digit = 8
10 - 8 = 2
Check Digit = 2 (123456782)

If the barcode does not match the result of the checksum algorithm, the following error message is displayed: “incorrect barcode”.

20 Item Changes Alert

You can configure Aleph to display the following message if a user updates an item and then moves to another item without saving the changes:

```
Item has been changed locally, but the changes have not been saved on
the server. Continue anyway?  Yes/No
```

This functionality is flag dependent.

1) In order to have this item change alert, set the following in alephcom.ini:

```
[ItemList]

NotifyItemChanged=Y
```

2) If localization is required, add the following line to `alepcom/tab/lng/message.dat`:

```
    ItemChange                      ALEPH GUI            YN  2 Q Item
%d-%d has been changed locally, but the changes have not been saved
on the server. Continue anyway?
```

Note: This message appears only when moving between items within the items list pane. When moving to the left pane or any other area that is not the items list, the message does not appear.

21 Multiple Item Addition Alert

You can configure Aleph to display a message after the addition of multiple items, indicating the number of added items. For example: "32 item/s (out of 32) created".

This functionality is flag dependent.

In order to have this item change alert, set the following in alephcom.ini:

```
[ItemList]

ConfirmMultiItem =Y
```

Possible values: Y or N

Y - (Default) After adding multiple items a pop- up message indicates the number of created items.

N – After adding multiple items, a pop-up message is not displayed.

2) If localization is required, add the following line to `alepcom/tab/lng/message.dat`:

```
MultiItemsCreation             ALEPH GUI            O     E %d items
(out of %d) created
```

Ex Libris
Aleph

System Librarian's Guide - Printing

Version 23

Ex Libris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 PRINT.INI - SETUP FOR GUI PRINTOUTS....4

2 ALEPHCOM.INI....7

3 CIRC.INI....10

4 THE FORM-PRINT-METHOD TAG IN .TRN FILES....12

5 SENDING ATTACHMENTS IN E-MAIL MESSAGES....14

6 TRANSLATE RULES....14

6.1 Defining Translate Rules....14

6.2 Translate Routines....15

7 LOCALLY SAVING PRINTOUTS....20

8 BIAF LABEL PRINTING....21

9 DEFINING THE SCALE TO PREVENT PRINTOUTS/PREVIEW FROM BEING CUT OFF....23

1 print.ini - Setup for GUI Printouts

You can control where and when printouts are printed. To do this, go to the module's GUI tab directory (for example, AL500/Circ/Tab) and open the PRINT.INI file. This file defines parameters for printouts that are produced by the GUI.

Following is an example of the relevant section from the Circulation module's PRINT.INI file:

```
        1                      2  3 4      5            6.1         6.2
----------------------------+-+++----------+---------+---------
LoanReceipt                  00 N P loanrcpt.prn
PhotoWait                    00 N M photowt.prn
PhotoSlip                    00 N P photoslp.prn
PhotoLetter                  00 N M photoltr.prn withitem   tosend
ReturnReceipt                00 N P retrcpt.prn  NO-PRINT
ReturnPhoto                  00 N P retphoto.prn
ReturnHold01                 00 N M rethold.prn
ReturnHold02                 00 N M rethold.prn
ReturnHold03                 00 N M rethold.prn
ReturnHold04                 00 N M rethold.prn
BorLetter                    00 N B borlettr.prn
HoldWait                     00 N M holdwt.prn
HoldLetter01                 00 N M holdltr.prn
HoldLetter02                 00 N M holdltr.prn
HoldLetter03                 00 N M holdltr.prn
HoldLetter04                 00 N M holdltr.prn
HoldDelete                   00 Y M holddel.prn
PayLetter                    00 N P payltr.prn
```

Column 1 is the function name that has been assigned by the programmer for internal use. Do not change this name. Each function produces one or more printouts based on the templates.

Column 2 is the version number of the template. When editing the templates, you may define up to 99 different versions of each one (for example, acq-order-slip-01). You identify each version by adding the version number as a suffix to the template's filename. Decide which version you want to be printed by the client and enter the version number in Column 2.

Column 3 defines whether or not the "Print" dialog box (shown below) will be displayed before the file is printed. Enter Y if you want the dialog box to be displayed; otherwise, enter N. If you enter N, the files will be printed on the GUI module's default printer, unless they have been set up otherwise in columns 6.1 and 6.2.

Column 4 defines whether the file will be printed (P), sent by e-mail (M), or both (B).

If "M" or "B" are entered in column 4, make sure that the recipient (for example, a vendor or a patron) has an e-mail address in his record, there is an appropriate form_print_method line with EMAIL in column 2 in the translation file, and the

print file used is readable by the recipient's e-mail program (there are plain text versions available).

The printout has a corresponding `trn` file in the library's `form_lng` directory which includes the EMAIL command.

For example:

The Acquisitions Monograph Order Slip uses the acq-m-order-slip template. Either the corresponding `trn` file (`acq-m-order-slip.trn`) or the `global.trn` file should include the following line:

```
!Tag                                  Type       Parameters
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!->
form-print-method                     EMAIL      vendor-email-address
```

Column 5 is the name of the file as it is saved on the client's local drive. You may enter any filename you want.

The system does not save all the source files of all the printouts, rather, only the last file of each printout will be saved in the GUI's <module>/Files/<Library> directory.

Note that files are NOT saved on the client for a particular function if columns 6.1 and 6.2 are filled in by Print IDs.

Columns 6.1 & 6.2 are optional columns that may be used for **either** Print IDs (for offline printing through the Task Manager and Print Daemon) **or** for Device IDs (for online printing to a printer other than the PC's default printer). (Do **not** use Print IDs and Device IDs together.) Note that if you leave columns 6.1 and 6.2 blank, the printouts will be printed on the PC's default printer.

The Print ID must be entered in lowercase.

- **Device IDs**
  Enter a Device ID in these columns if you want the files to be printed out online using a printer other than the PC's default printer. You might want to do this, for example, if you have a printer with special paper for labels.

  Enter in columns 6.1 and 6.2 one of the Device IDs that appear in the ALEPHCOM.INI file, under the [PrinterDeviceList] section. For example, if the list shows:

  ```
  [PrinterDeviceList]
  Device01=Digital DEClaser 3250
  Device02=HP DeskJet 500
  Device03=HP DeskJet 600
  ```

You could enter DEVICE-01 in column 6.1 and DEVICE-02 in column 6.2. Note that you must type a **hyphen** between the word "DEVICE" and the number identifying the device, and that the device ID must appear in capital letters.

Following are two examples where both column 6.1 and 6.2 are used:

Example: In Circulation, the HoldLetter, ReturnHold and PhotoLetter functions produce two printouts if the delivery type is pickup, **column 6.1** is reserved for the Device ID of the printout to be placed with the patron letter and **column 6.2** is for the Device ID of the printout of the item slip.

If you do not want a printout to be produced for a particular function, type "NO-PRINT" in column 6.1 (and/or column 6.2) (as shown above for the ReturnReceipt function).

If you use Device IDs, the system will not save all the source files of all the printouts, rather, only the last file of each printout will be saved in the client's CIRC/FILES/<LIBRARY> directory. This means that in the above example, there will be one `loanrcpt.prn` file, one `photowt.prn` file, one `photoslp.prn` file, and so on, saved on the server. Note that since the PhotoLetter function produces two printouts, the text of both printouts will be included in one `photoltr.prn` file.

- **Print IDs**
  Enter a Print ID in these columns only if you do NOT want the files to be printed out online and you'd rather save the files on the server and print the files out later using the Task Manager.

  Print IDs are used to identify the file as belonging to a particular person (for example, Evelyn Stanhope) or department (for example, Circulation), or purpose (for example, "with item") or to identify the printer that the file should be printed on (for example, dec3500).

  You can enter any Print ID you want, although in the above example, we used the Print IDs to identify the purpose of the printout.

  Following are two examples where both column 6.1 and 6.2 are used:

  Example: In Circulation, the HoldLetter, ReturnHold and PhotoLetter functions produce two printouts if the delivery type is Pickup. **Column 6.1** is reserved for the Print ID of the printout to be placed with the item and **Column 6.2** is for the Print ID of the printout to be sent to the patron.

  The source text of each printout will be saved in its own file on the server. (This is different from the way the files are handled when they are saved on the client. See the explanation for Column 5 above).

  The print ID that you give will be suffixed to the output filename. The

file will be saved on the server in the library's PRINT directory (and not on the client).

If the Print Daemon is not set up, you can use the Task Manager to manually select a file for printing.

If the Print Daemon is set up in the GUI Task Manager, it will recognize the suffix and print the file automatically on the default printer of the PC on which the Print Daemon is running.

If the Print Daemon has printed the files, the system will automatically move the files from the library's PRINT directory to the library's PRINT/SAVE_<DATE> directory, and add a time stamp as a suffix to the filename.

Note that if you use Print IDs, all files that produce a printout will be saved on the server, not just the last file. This means that if ten routing lists were produced, ten files will be saved on the server. The files will therefore need to be periodically deleted after they have been printed.

When using Print IDs, columns 3, 4, and 5 in the file print.ini in the module's GUI tab director are not relevant. The configuration in columns 3, 4 and 5 in file print.ini in alephcom/tab is relevant when later printing out the files using Print Daemon.

For more information about Print IDs, refer to the Task Manager - Print IDs chapter.

2 alephcom.ini

[Print]

```
[Print]
SaveHistoryNumberOfDays=1
PreviewLimit=5
DefaultPrintConfig=1
TempDir=c:\temp\Aleph
TempFileCounter=1063
PrintoutCounter=659
PrintoutFileCounter=2175
NewPrintType=Y
```

Every printout is saved in the GUI's history. The history is built for every module separately. For example, for the USM50 library, the history is built in `Alephcom\files\USM50\print...`

SaveHistoryNumberOfDays=1
This setting determines how many days the print history should be stored. Cleanup of old history files takes place when the GUI is started.

PreviewLimit=5

When using the Preview option for large files with more than five printouts, the following question is displayed: "You are going to Preview <n> additional documents with no option to stop the process. Continue Preview-yes/no?". It is possible to continue or to stop the Preview process. The question is displayed after five printouts (by default) or after the number of printouts set in Alephcom.ini: PreviewLimit.

DefaultPrintConfig=1
Determines the default for the options on the printer icon on the bottom right of the GUI (0=normal,1=preview, 2=view raw XML, 3=browse XML).

TempDir=c:\temp\Aleph
This is the location where temporary files reside.

TempFileCounter=1063
Do not change this line.

PrintoutCounter=659
Do not change this line.

PrintoutFileCounter=2175
Do not change this line.

NewPrintType=Y
This setting defines which of the two printing utilities will be used. The default value 'Y' uses the new utility and is the recommended option.

[PrintExecute]

```
[PrintExecute]
ERROR_EDITOR=notepad.exe
EDIT=notepad.exe $1
EXE_COMMAND=C:\Program Files\Plus!\Microsoft Internet\Iexplore.exe $1
WORD=C:\Program Files\Plus!\Microsoft Internet\Iexplore.exe $1
BROWSER=C:\Program Files\Plus!\Microsoft Internet\Iexplore.exe $1
SaxonVersion=Saxon7.jar
```

This section defines which external program will be used to open an XML print file. This section works according to the setup of column 3 of `USM50/tab/form_print_method`.

SaxonVersion=Saxon7.jar
This variable defines which utility will be used for printing. The available options are Saxon7.jar and Saxon8.jar. Note that both options require an installed Sun Java Virtual Machine. If no parameter is entered, the default is Saxon7.jar.

[PrinterDeviceList]

```
[PrinterDeviceList]
Device01=QC-LJ2100PCL6
Device02=Digital DEClaser 3500

Device03=HP LaserJet 2100 Series PCL 6
```

```
Device04=Q-SUPPORT
```

The [PrinterDeviceList] in the ALEPHCOM.INI file lists the printers that are available for online printing as an alternative to the PC's default printer.

You can add more devices to the [PrinterDeviceList] by following these steps:

Step 1:
Open "Printers and faxes" in your control panel (Start → Settings → Printers and Faxes).

Step 2:
Right click on the relevant printer and select the Properties option.

Copy the name of the printer as it appears in the General form in the first text box.

Step 3:
Open the ALEPHCOM.INI file (found in the client's ALEPHCOM/TAB directory).

Step 4:
Go to the [PrinterDeviceList] section. Following is an example of what you may find there:

```
[PrinterDeviceList]
Device01=HP DeskJet 500
Device02=HP DeskJet 600
```

Define a new Device by typing a new Device number (for example, Device03), and pasting the name of the printer to the *right* of the equal sign, as shown below:

```
Device03=Digital DEClaser 3250
```

The Device ID (for example, Device03) may then be entered in Columns 6.1 and 6.2 of the Circulation module's PRINT.INI file in order to define the printers to be used for online printing. Note that when entering the Device ID in Columns 6.1 and 6.2, a **hyphen** must be inserted between the word "DEVICE" and the device number and that it must be capitalized, for example, DEVICE-03.

[General] Targets

```
[General]
Targets=printer01,printer02,photoslip,photowait,photofill
```
This parameter defines the list of print IDs that is displayed in the drop-down list of a service.

[Mail]

```
[Mail]
```

```
MailServer=exchange01
FromAddress=library@exlibris.com
MailerName=Exlibris Library Services
AuthMethod=
```

MailServer=exchange01
Enter the name of the mail server used by your site's network. You can add the port number where a non standard port is used, using a <host>:<port> format. For example, 10.345.24.11:45.

FromAddress=library@exlibris.com
Enter the name that should appear as the sender of e-mail messages sent to patrons by the system.

MailerName=Exlibris Library Services
Enter the display name that will appear instead of the actual e-mail address.

AuthMethod=
Enter the required authentication method. The supported authentication methods can be LOGIN or PLAIN. If the parameter is left blank, then no authentication method is used. Note that the user name and password can only be set using the Configuration\E-Mail Settings menu option in the AlephADM module.

[PrintDaemon]

```
[PrintDaemon]
PrintdAutoActivate=Y
PrintdAutoDeactivate=Y
```

PrintdAutoActivate=Y
Determines whether or not the print daemon will be activated automatically when any ALEPH module is opened.

PrintdAutoDeactivate=Y
Determines whether or not the print daemon will be deactivated automatically when the last ALEPH module is closed.

3 circ.ini

The setup definitions below include all the circ.ini definitions that relate to printing.

[Payment]

```
[Payment]
CashNoReceipt=1
```

On the right side of the equal (=) sign, type the number of receipts (up to 9) that you want to be printed. If you do not want any printed, type zero ("0").

The actual printer that will be used is defined in the PRINT.INI file, with the function

name PayLetter or LineLetter. If the Pay Selected option is used, then the LineLetter entry is used. If the Pay Sum option is used, then the PayLetter entry is used.

[LoanList]

```
[LoanList]
WantBorListOnRenewAll=Y
```

If the value is N, the system does not automatically print a letter after a Renew All procedure. If the value is Y, printing of the form bor-list-renew-all is automatically triggered.

[HoldList]

```
[HoldList]
WantDeleteLetter=Y
```

Determines whether the system should send a letter to the patron if a hold request is deleted from his Hold List.

[ItemSchedList]

```
[ItemSchedList]
WantDeleteLetter=N
```

If you want the system to automatically generate a letter to the patron informing him that his advance booking reservation has been deleted, type Y. If you do not want a letter generated, type N.

[HoldFilledList]

```
[HoldFilledList]
AutoSelectMode=M
WantDeleteLetter=Y
```

This section provides the option to automatically select and close the Hold List window when using "Print Letter - Hold Request Filled", when no button is clicked and there is only one hold on the list. The system will automatically select the patron from the list, print a letter for that patron and close the window in 1.5 seconds.

AutoSelectMode=M
This parameter controls the auto select of the first entry in the Hold Filled List. If the switch is set to S(ingle) the system will automatically print the letter if there is a single entry in the list and no button is clicked. If there is more than one hold on the list, the selection must be performed manually. If the switch is set to M(ulti) the system will automatically print the first entry in any case, whether there is only one or more entries in the list.

WantDeleteLetter=Y
Determines when a hold request is deleted, whether to send a letter to the patron notifying him that his request was deleted from the "Hold filled List".

[ReturnHold]

```
[ReturnHold]
AutoSelectMode=M
WantDeleteLetter=Y
```

This section gives the option to automatically select and close the Hold List window when the item is returned and there is only one hold on the list. The system will automatically select the patron from the list, print a letter for that patron and close the window.

AutoSelectMode=M
This parameter controls the auto select of the first entry in the Return Hold List. If the switch is set to S(ingle), the system will automatically print the letter if there is a single entry in the list. If there is more than one hold on the list, the selection must be performed manually. If the switch is set to M(ulti), the system will automatically print the first entry in any case, whether there is one entry or more than one in the list.

WantDeleteLetter=Y
Determines when a hold request is deleted, whether to send a letter to the patron notifying him that his request was deleted from the "Returned item hold list".

[Return]

```
[Return]
PrintReceipt=N
PrintReceiptTransfer=N
PrintPhotocopyRequest=Y
```

These three settings are chosen in the "Set Up Return Options" window and are saved in this section.

4 The form-print-method Tag in .trn files

`form-print-method` is a reserved tag that is used, in the relevant `.trn` translation file in the library's `form_lng` directory, to alter the method in which the printout will be printed. The `form-print-method` tag can be used both in the `global.trn` file and in a specific printout's `.trn` file. Only one tag can be defined in one `.trn` file. The options are:

```
!Tag                                Type          Parameters
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!->
form-print-method                   EMAIL

form-print-method                   SAVEAS
form-print-method                   DISPLAY
form-print-method                   NONE
form-print-method                   EMAIL-ONLY
```

EMAIL

This option creates the following XML tag in the XML file's header:

```
<email-address>Y</email-address>
```

using the <email-address> definition from <section-01>.

SAVEAS

This option creates the following XML tag in the XML file's header:

```
<saveas>Y</saveas>
```

Use this option when you want to save the printout as a file on the local disk. Instead of the printout being sent to the printer, the system will offer a standard Microsoft Windows "Save As" dialog box.

DISPLAY

This option creates the following XML tag in the XML file's header:

```
<display>Y</display>
```

Use this option when the print configuration is set to normal printing but you want to force a print preview before printing.

NONE

This option ensures that the <email-address>, <saveas> and <display> tags are not created in the XML file's header.

The `global.trn` file has the EMAIL option by default. However, there may be some printouts that you do not want to send by e-mail. Use the NONE option in these printouts' `.trn` files. The local `.trn` file's NONE definition will override the `global.trn` file's EMAIL definition, and no tag will be created in the XML file that is generated when creating the printout.

EMAIL-ONLY

This option facilitates the sending of slips/letters only to patrons who have a valid e-mail address . This option distinguishes between slips/letters which do or do not have an empty `<email-address>` tag in the XML file. The slips that do not have e-mail addresses are put in the `$data_print/no_email` directory. The slips that do are put into `$data_print` and can be sent later by e-mail.

The TRN file must have the following setup:

```
!                                                M
!                                                /
!Tag                                  Type        S Parameters
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!->
form-print-method                     EMAIL-ONLY S no_email,MESSAGE
```

This syntax is required in order for this option to work.

The second parameter, MESSAGE, is not mandatory and if it is put as parameter it means that a message will be presented to the librarian for every slip without an e-mail address when opening the slip's output in Task Manager. The message text comes from the `$alephe_root/error_lng/server_execute_no_email` file. If there is no such file, the message will come from the `$alephe_root/error_lng/server_execute` file.

5 Sending Attachments in E-mail Messages

You can disable attachments in all e-mails sent by the system via the tab100 variable PLAIN-ONLY. This variable can only be configured in the alephe/tab tab100 table, and can have one of the following values:

Y = Do not send an e-mail attachment, no matter how the patron or vendor records are defined. Only plain text in the e-mail body will be sent.

N = Send e-mail attachments depending on how the patron or vendor records are defined.

H = Do not send an e-mail attachment, no matter how the patron or vendor records are defined. Only HTML format in the e-mail body are sent.

6 Translate Rules

Translate rules provide a mechanism that enables reformatting of ALEPH's output XML files from their raw format into a more readable format. For example, they may be used to change date fields from a yyyymmdd format to the more readable mm/dd/yy format. Translate rule are used to control the format of all XML outputs, including printout XML files and XML files that are used by the GUI for information display.

6.1 Defining Translate Rules

Translate rules are defined in trn files in the following way:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!->

bib-info            SUPPRESS   S ##
```

Column 1 – The XML tag that is to be formatted. The above example defines a formatting rule for the information that is found under the XML tag <bib-info>.

Column 2 – The translate routine. For example, the function SUPPRESS means 'remove from the field's data the characters that are defined in the parameters column'.

Column 3 – Single/Multi field. ‘M’ is used when the tag data is composed of several codes delimited by spaces, and each is less than 100 characters. For example, if the XML file tag <p-currency-name> has the format of:

<p-currency-name>usd gbp jpy</p-currency-name>

then a trn line of

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!-!-
!!!!!!!!!!!!!!!!!!!->

p-currency-name  currency       M
```

will translate the XML tag to

<p-currency-name>US Dollar, Great Britain Pound, Japanese Yen</p-currency-name>

Column 4 – The parameters that the translate routine requires. In the above example, the characters ‘##’ are to be suppressed from the data of the bib-info tag.

The trn file that will be used for formatting the XML output is chosen by using the following rules:

For printout XML files:

A trn file with the same name as the display XSL template will be searched for a relevant translation rule. For example, if the XML file’s XSL printout template is transfer-remaining-balance.xsl, then each tag in the XML file will be searched for a translation rule in the transfer-remaining-balance.trn file.

If no specific trn file is found, or if no relevant translation rule is found, the global.trn file (in xxx01/form_eng) will be searched for a relevant translation rule.

For GUI display XML files

A trn file with the same name as the display HTML template will be searched (in the xxx50/pc_display_lng directory) for a relevant translation rule. For example, if the XML file’s HTML display template is item-return then each tag in the XML file will be searched for a translation rule in the item-return.trn file.

If no specific trn file is found, or if no relevant translation rule is found, the file global.trn in the xxx50/pc_display_lng directory will be searched for a relevant translation rule.

If no global.trn file is found in the xxx50/pc_display_lng directory, or if no relevant translation rule is found, the general global.trn (in xxx01/form_eng) file will be searched for a relevant translation rule.

6.2 Translate Routines

The following translate routines can be used in ALEPH’s trn files:

ADD-SIGN – This routine adds a sign (‘+’ or ‘–‘) mark to the value in the formatted tag. When using this routine a parameter must be supplied in column 4. The parameter must be the name of a tag in the XML file that has the value ‘C’ if the formatted tag is a positive value and ‘D’ if the formatted tag is a negative value. For example:

```
z31-sum                                    ADD-SIGN          S z31-
credit-debit
```

AP-ZIP – This routine is used to append the zip code to the displayed address. The translate routine receives two parameters. The first is the zip code that is to be appended to the formatted tag, and the second is the condition field. Only if the condition field is empty will the zip code (in the first parameter) be added to the formatted tag. The following example demonstrates how this routine may be used to append the zip code to the last address field.

```
z302-address-0                             AP-ZIP                S
z302-zip z302-address-1

z302-address-1                             AP-ZIP                S
z302-zip z302-address-2

z302-address-2                             AP-ZIP                S
z302-zip z302-address-3

z302-address-3                             AP-ZIP                S
z302-zip z302-address-4

z302-address-4                          AP-ZIP              S z302-
zip

z302-zip                                    TO-BLANK         S
```

AP-ZIP-USA – This routine is similar to AP-ZIP. The only difference is that the ZIP code is also formatted in USA format, that is, the first five characters are separated from the rest by a hyphen (xxxxx-yyy).

AP-ZIP-CAN – This routine is similar to AP-ZIP. The only difference is that the ZIP code is also formatted in Canadian format, that is, the first three characters are separated from the rest by a space (xxx yyy).

ARR-STATUS – Determines the z68-arrival-status in slip to vendor when it is not complete with the same algorithm as the arrival status is determined in the order list. The status is "Partial" if any part of the order has arrived; it is "Non Arrived" if no part of the order has arrived.

```
z68-arrival-status                  ARR-STATUS S z68-order-number
```

CASH-DESC – This routine is used to format a cash transaction's description. If the description is empty then the tab18.lng description will be displayed. If the cash description is not empty then bracketed information will be omitted. The following example will cause a description of 'Late Retrun <15.35>' to be formatted into 'Late Return'.

```
z31-description                          CASH-DESC     S z31-type
```

CASH-STATU – This routine is used to display the status of a cash transaction. It replaces the information of the formatted tag with a textual description that is taken from the $aleph_root/error_lng/cash_status_heading. The text that will be used depends of the status of the cash transaction whose record key is supplied as a parameter. For example:

```
z31-status                                 CASH-STATU     S z31-
rec-key
```

CASH-TYPE – This routine is used to translate cash type codes into their description in tab18.lng. For example:

```
z31-type                                    CASH-TYPE       S
```

COLLECTION – This routine is used to translate collection codes into their textual description in tab40.lng. The routine receives the sublibrary to which the collection belongs as a parameter. For example:

```
z30-collection                          COLLECTION     S z30-sub-
library
```

CREDIT-TXT – This routine adds the text that is specified as a parameter to sums that are proceeded by "+". For example, a definition of

```
z31-sum                                      CREDIT_TXT        S
(Credit)
```

will translate a tag of

```
<z31-sum>+5.50</z31-sum>
```

to

```
z31-sum>5.50  (Credit)</z31-sum>
```

A tag of

```
<z31-sum>-5.50</z31-sum>
```

Will be translated to

```
</z31-sum>5.50</z31-sum>
```

DEBIT-TXT – This routine adds the text that is specified as a parameter to sums that are proceeded by "-". For example, a definition of

```
z31-sum                                     DEBIT_TXT         S
(Debit)
```

will translate a tag of

```
<z31-sum>-5.50</z31-sum>
```

to

```
z31-sum>5.50  (Debit)</z31-sum>
```

A tag of

```
<z31-sum>+5.50</z31-sum>
```

Will be translated to

```
</z31-sum>5.50</z31-sum>
```

CURRENCY – This routine replaces a currency code with the currency text, as is set in the Z83 table. For example:

```
z41-copyright-currency              CURRENCY       S
```

DATE – This routine formats a date field from the yyyymmdd format to the format that is specified in the date_style_f or the date_style_s variable of $alephe_root/aleph_start. For example:

```
z108-date-from                          DATE                S
```

DECIMAL – This routine formats fields that are stored as decimal fields into readable decimal form. In the following example, a tag with 000000001000 will be formatted to 10.00.

```
z77-vat-amount                          DECIMAL          S
```

DEL-SIGN – This routine deletes sign marks (‘+’ or ‘-‘) from the formatted field. The below example will translate <local-price>+12</ local-price> to <local-price>12</ local-price>:

```
local-price                             DEL-SIGN         S
```

DUE-DATE – This routine is used to replace the formatted tag with the effective due date that is used by the system. Since a loan’s effective due date depends not only on the z36-due-date field, but also on the z36-recall-date and z36-recall-due-date fields, these fields are used as parameters for calculating the due date. For example, if you need to display a loan’s due date in a printout, use the z36-due-date field in the printout XSL template and use the following definition in the relevant trn file:

```
z36-due-date                            DUE-DATE         S z36-
recall-date z36-recall-due-date
```

INTEGER – This routine formats fields that are stored as integer fields into readable integer form. In the following example, a tag with 000000001000 will be formatted to 1000.

```
status-counter                          INTEGER          S
```

INV-STATUS – Determines the z68-invoice-status in slip to vendor when it is not complete with the same algorithm as the arrival status is determined in the order list. The status is “Partial" if there is any information about the invoice; it is "Non Invoiced" if there is no information at all.

```
z68-invoice-status                  INV-STATUS S z68-order-number
```

ITEM-SEQ – This routine is used for formatting item sequences. It inserts a dot before the last digit, changing an item sequence of 000010 into 1.0. A sample use is:

```
z30-item-sequence                       ITEM-SEQ         S
```

ITEM-STA – This routine is used to replace an item status code with the textual description that is defined in tab15.lng. As tab15.lng texts are sensitive to the item’s sublibrary and process status, these values are used as parameters by this translate routine. For example:

```
z16-item-status                         ITEM-STA         S z16-
sub-library z16-item-process-status
```

LOAN-STAT – This routine is used to display the status of a loan. It replaces the information of the formatted tag with a textual description that is taken from the $aleph_root/error_lng/loan_status_heading. The text that will be used depends of the status of the loan whose record key is supplied as a parameter. For example:

```
z36-status                              LOAN-STA         S z36-
rec-key
```

LOCATION and LOCATION-B – Both of these routines are used for call number formatting. LOCATION is used to add the subfield marks if the item's call number does not include them. The call number type is therefore required as a parameter. LOCATION-B is used to remove subfield marks. It therefore does not require parameters. For example:

```
z30-call-no                                    LOCATION        S z30-
call-no-type
```

or

```
z30-call-no                                   LOCATION-B     S
```

NONE – This routine is used to prevent a field that would otherwise be formatted from being formatted. For example, if a patron's password is blocked from being exported in the XML by a global.trn TO-BLANK definition but it should be exported in some specific output, that specific output can be protected by having a NONE translation routine in its specific trn file.

OBJ-CODE – This routine formats a budget object code to the text that is defined in the OBJECT-CODE section of $data_tab/pc_tab_exp_field.lng.

PC-TAB-EXP – This routine translates the formatted tag into the text that is defined for it in the $data_tab/pc_tab_exp_field.lng table. The section in the table from which the text will be fetched is supplied as a parameter for this routine. For example:

```
z16-delivery-type                              PC-TAB-EXP     S VENDOR-
DELIVERY-TYPE
```

RENEW-INFO – This routine is used to add renew limit information. The information that may be added includes the limit on the total number of renewals and the limit on the renew period. The translate routine uses texts from the $aleph_root/error_lng/circ_renew_heading_renew_info or $aleph_root/error_lng/www_f_heading files. For example:

```
z36-no-renewal          RENEW-INFO S z36-rec-key
```

SIGN1 – This routine is used to format sign indications. It drops the plus sign ('+') from positive values and puts negative values (prefixed by '-') inside brackets.

SIGN2 – This routine is used to format sign indications. It drops the plus sign ('+') from positive values.

SUB-LIB – This routine translates a sublibrary code into its textual description in tab_sub_library.

SUPPRESS – This routine suppresses substrings from the formatted tag. The substrings that will be suppressed are given as parameters. Each parameter is a string, and they get suppressed in the order they appear. For example:

```
z71-data                                       SUPPRESS       S 234 89
```

TEXT – This routine is used to translate codes into texts. The parameters of the routine include code=text pairs. The code part is the value that will be found in the

unformatted XML and the text part is the text that the code will be translated into. If the text is more than one word long, each word will be separated by underlines. For example:

```
has-fines                                   TEXT                    S
Y=Has_debts N=No_debts
```

TIME – routine formats a time field from *hhmm* format to the format that is specified in the time_style_f or time_style_s variable of $alephe_root/aleph_start. For example:

```
z37-booking-end-hour                TIME                    S
```

TO-BLANK – This routine deletes the date from the XML tag. It may be used to delete data that is included in the XML file but should not be exported. For example

```
z308-verification                           TO-BLANK        S
```

USER-FUNC – This routine is used to translate staff permission codes into their descriptive texts, based on the $alephe_tab/user_function table. The function receives the Z67-FUNC and Z67-SUB-FUNC as input parameters, for example:

```
z67-func            USER-FUNC  S z67-func z67-sub-func
```

Z30-MATER – This routine translates material codes into the texts that have been defined in tab25.lng.

ZIP-USA – This routine formats zip codes in USA format, that is, the first five characters are separated from the rest by a hyphen (xxxxx-yyy).

ZIP-CAN – This routine formats zip codes in Canadian format, that is, the first three characters are separated from the rest by a space (xxx yyy).

7 Locally Saving Printouts

Any printout that is processed by the GUI client (i.e. either printed or e-mailed) may be locally saved on the client side, in a location that is configured in the alephcom\tab\saveas.ini file. The file has he following structure:

```
! 1  form name (30 characters)

! 2  Store location (250 characters)

!

! 1                              2

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!+!!!!!!!!!!!!!!!!!!!!!!!!
```

Column 1 – The name of the form that is to be locally saved.

Column 2 – The client location on which the printout will be saved.

For example, a setup such as the following will result in the the cash receipt being locally saved in the C:\TILL_Printouts directory:

```
!
!                    1                                        2
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!+!!!!!!!!!!!!!!!!!!!!!!!!!
cash-pay-receipt-pay   C:\TILL_Printouts
```

8 BIAF Label Printing

BIAF is a powerful program for designing and printing barcode labels and other labels that can read data from Unicode XML sources. BIAF can be used to print customized item labels from ALEPH.

The following are the required setup steps for using BIAF for label printing from ALEPH.

1. Define BIAF form_print_method

The administrative library file $data_tab/form_print_method must include the print method BIAF_LABEL_PRINT in column 3. This should be defined for each form which you wish to be printed via the BIAF label printing program.

For example, if you want to use the BIAF Print Label software for the following forms:

item-copy-label-00
Item-copy-label-07
item-issue-label-00
item-issue-label-20
serial-item-label-00
serial-item-label-01

then $data_tab/form_print_method must be set up as follows:

```
!     1                              2  3                                          4  5
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!-!!!!!!!!!!!!!!!!!!!
```

The BIAF Print Label Software must be installed under the ALEPH version root. For example, if ALEPH is installed under C:\AL500 then the program must be installed in C:\AL500\BiafLabel.

3. Configure the BIAF Software

Edit the BiafLabel\PrintQueueConfig.ini file, defining the form name, language, number and the full path to the print template file. The syntax is as follows:

form name,language,form number@ print template filename with full path

For example:

item-copy-label,ENG,00@C:\AL500\BiafLabel\Templates\Aleph500-Samplec00.lbs
item-copy-label,ENG,07@C:\AL500\BiafLabel\Templates\Aleph500-Samplec07.lbs
item-issue-label,ENG,00@C:\AL500\BiafLabel\Templates\Aleph500-Samplei00.lbs
item-issue-label,ENG,20@C:\AL500\BiafLabel\Templates\Aleph500-Samplei20.lbs
serial-item-label,ENG,00@C:\AL500\BiafLabel\Templates\Aleph500-Samplesi00.lbs
serial-item-label,ENG,01@C:\AL500\BiafLabel\Templates\Aleph500-Samplesi01.lbs

In the above case there will be print templates in path
C:\AL500\BiafLabel\Templates\

Notes:

Do not insert empty lines in the beginning of the PrintQueueConfig.ini file.

It is impossible to use comment marks in the PrintQueueConfig.ini file.

4. Define Printer

When using the BIAF Print Label software, there are two options for defining the labels printer.

a. In the print.ini file

In the ItemLabel line define the non default printer in column 6, for example:

ItemLabel 00 N P item_lbl.prn **DEVICE-07**

b. In the BiafLabel\PrintQueueConfig.ini file

In the last column of PrintQueueconfig.ini define which printer is to be used for the specific line, for example:

item-copy-label,ENG,00@C:\AL500\BiafLabel\Templates\Aleph500-Samplec00.lbs@**il-dc02\hp2420-alph**
item-copy-label,ENG,07@C:\AL500\BiafLabel\Templates\Aleph500-Samplec07.lbs@**il-dc02\hp2420-alph**
item-issue-label,ENG,00@C:\AL500\BiafLabel\Templates\Aleph500-Samplei00.lbs@**il-dc02\hp2420-alph**
item-issue-label,ENG,20@C:\AL500\BiafLabel\Templates\Aleph500-Samplei20.lbs@**il-dc02\hp2420-alph**
serial-item-label,ENG,00@C:\AL500\BiafLabel\Templates\Aleph500-Samplesi00.lbs@**il-dc02\hp2420-alph**
serial-item-label,ENG,01@C:\AL500\BiafLabel\Templates\Aleph500-Samplesi01.lbs@**il-dc02\hp2420-alph**

Note: The non default printer definition in PrintQueueConfig.ini file, will be used only when there is no definition for a non default printer in ALEPH module's print.ini.

Refer to BIAF documentation for description of the BIAF software usage.

9 Defining the Scale to Prevent Printouts/Preview from Being Cut Off

When printing/preview wide tables, the table might be cut off. In order to avoid this problem, define the following parameter: .\Alephcom\Bin\HTMLPrint.ini:

```
[Options]
Scale=75
```

Scale=100 is the default

This change affects all printouts. There is no option to fit specific printout only.

System Librarian's Guide – Search

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 SYSTEM SETTINGS 4

1.1 Sort Limit and Sort Order of Records 4

1.2 Tree View Display 5

1.3 Search Limits 6

1.4 Limit and Format for Separate Display of Linked Records 6

2 LOGICAL BASES 7

3 FIND QUERIES 8

3.1 Boolean Operators Setup 9

4 HOLDINGS INFORMATION 10

4.1 Holdings Note for Serials 10

5 DISPLAYING AND INDEXING COPY INFORMATION 10

6 COLUMN HEADINGS 11

6.1 pc_tab_short.lng - Brief List 11

6.2 pc_tab_col.lng 12

6.3 tab_col.dat 13

6.4 Multi-field Search 14

7 MENUS 14

7.1 Index Menus 14

7.2 Formats Menu 15

7.3 Sort Order Menu 16

8 LOCATE FUNCTION 16

9 DEFAULT TEXT FOR LINKS 17

10 SEARCH FUNCTION TABLES 18

1 System Settings

You define Search function system settings in the `pc_server_defaults` table, located in the `alephe` directory. This table includes Search function defaults.

The elements listed below are defined in the default parameter section of the `pc_server_defaults` table.

1.1 Sort Limit and Sort Order of Records

The following lines in `pc_server_defaults` refer to Search function sort settings:

```
setenv pc_sort_field       "01   D02   A"
setenv pc_sort_field_aut   "03   A01   D"
setenv pc_sort_field_sub   "01   D02   A"
setenv pc_sort_field_lcc   "DISPLAY"
```

The maximum number of records that can be sorted is 1000.

The default sort order for the results of a **Search** query are defined by the parameter below (the staff user can change the sort order of a set when the set is displayed in Brief Table format):

```
setenv pc_sort_field          "01---D02---A"
```

01 and 02 are codes taken from column 1 of the `tab_sort` table. In this case, 01 refers to the Year field and 02 refers to the Author field. To sort the records by a different field, make sure the field is defined in `tab_sort` (see the Tables Section at the end of this chapter for an explanation of `tab_sort`). Here is an example from the table:

```
!1 2    3     4   5   6     7  8   9    10 11 12   13 14 15   16 1718
19
!!-!!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-
!!!!!-!-!!-!!
01 95 008           260## c
08 04

02 01 1#### a
00 00
```

The letter "D" for the Year field stands for "descending" order, which means that the latest dates will be listed first (for example, 1999 will be listed before 1998). The letter "A" stands for "ascending" order and means that the order of the Authors will be from A to Z.

The default sort order for the results of a **Browse** query are defined by the parameters below (the staff user cannot change this sort order):

```
setenv pc_sort_field_aut      "03---A01---D"
setenv pc_sort_field_sub      "01---D02---A"
```

The "aut" and "sub" in the above parameters refer to the Browse Indexes as defined in tab00.lng. In this case, "aut" refers to Author and "sub" refers to Subject. You can define other parameters, but you must use the three-letter code of an index that is defined in tab00 (see the Tables Section at the end of this chapter for an explanation of tab_00.lng).

The character strings
```
02   A03   A
```

are defined in the same way as in the Search query.

If you want to display lists of records sorted by specific heading codes in the Show node, ensure that the following line appears in `pc_server_defaults`:

```
setenv pc_sort_field_<heading code>   "DISPLAY"
```
This line can be repeated for different codes.

For example, the line *setenv pc_sort_field_lcc "DISPLAY"* appears in pc_server_defaults. A user browses by LC classification number and gets a list sorted by ascending LC classification numbers. When he moves selected records to the Show node, the sorting stays the same (that is, a sort by ascending LC classification numbers).

1.2 Tree View Display

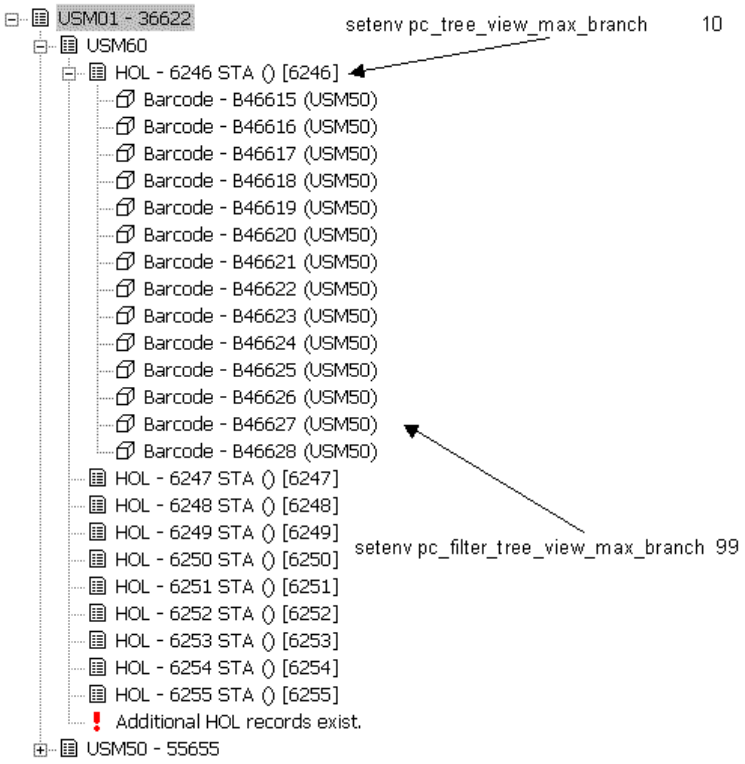
The following line in the table refers to Search function and cataloging module Tree View settings:

```
setenv pc_tree_view_max_branch 99
```

`pc_tree_view_max_branch` defines the limit after which the display of the navigation tree nodes in the Cataloging module and in the Overview tree of the Search tab is truncated with an appropriate message. This parameter limits either the number of ADM or HOL links of the given BIB record. If it is set to 5, only 5 ADM records that are linked to this BIB are displayed. The following parameter:

```
setenv pc_filter_tree_view_max_branch  99
```

defines the limit after which the display of the children of the navigation tree nodes in the Overview tree are truncated with an appropriate message. If both parameters exist, you will be able to see more information. In both cases the trees are limited to 800 lines.



In the example above, only 10 holdings can be displayed. A message, *Additional HOL records exist*, is displayed to let the user know that there are more holdings records available for this bibliographic record. Up to 99 items linked to each Holding record can be displayed.

1.3 Search Limits

There are no limits in creating a set in the Search Function. There are, however, limits for displaying and sorting a set. This means that:

- There is no maximum number of documents permitted in a set created after a Search query.
- There is no maximum number of words that are "collected" when truncation is used (when, for example, *find a?* retrieves all words beginning with *a*).
- There is no maximum number of hits that can be saved on the server.
- The number of hits from a set that display and are sorted can be set in the `set_result_set_limit` environment variable in `pc_server_defaults`. It can be between 1000-20000. If this variable does not exist, the default is 1000.

1.4 Limit and Format for Separate Display of Linked Records

A BIB record can have a number of bibliographic links to other BIB records. The Full view in the lower pane in the Show node can display a single line, pointing to a set of records, instead of a number of linked lines. The set is shown in the Brief list table.

Column 16 of `edit_doc_999.lng` defines after how many link lines the single line, pointing to a set, is displayed. If the column is empty, the default is 99 links.

In Column 15, you can determine a Brief list format for every bibliographic link. If a format is not defined in the table, the program displays the default format type.

Example:

```
!                                                 1     1       1
!1   2   3      4     5 6 7         8          9 0 11  2  13   4 15
16
!!-!!!!!-!-!!!!!!!!!!-!-!-!!!!!!!!!!!!!!!!!!!!-!-!-!!!-!-!!!!!-!-!!-
!!
## UP                    D LUplink              Y Q     E          00
02

## PAR                   D LParall              Y Q     E          00
02
## DN                    D LDown                Y Q     E          00
02
```

2 Logical Bases

You can group bibliographic records into logical bases that the user can search and browse through. Examples of logical bases are records for periodicals or audio-visual materials, or materials about History, or France. When the user performs a Browse or Find Query, the system automatically excludes items that are not included in the logical base.
A given bibliographic record may be included in any number of bases but a base can be defined within one library only. A base cannot be defined to include more than one library.

Your ALEPH system must have word indexes built in order for the base option to work. You can update the word indexes using the Update Word Index (manage-01) service from the Services menu in the Cataloging module.

In order to define a logical base, edit `tab_base.lng`. Following is a sample from the table:

```
!         1          2           3            4     5     6     7   8
9
!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!-!!!!!!!!!>
MONOGRAPHS             Monographs (USM01)   USM01             USM01 Y
wfm=BK
USM01                  ExLibris University                    USM01 N
LINC                   Lincoln Library      USM01             USM01 Y
wsl=ulinc
EDUC                 * Education Library    USM01             USM01 Y
wsl=ueduc
USM01_PAC              USM01 PAC            USM01             USM01 N
wrd=alldoc
uments not wst=(suppressed or deleted)
HISTORY              * History              USM01             USM01 Y
wti=history
```

Column 1 is the code identifying the logical base. When displaying screens in the OPAC, the system looks for files that use this code as an extension to the file name (for example, find-a-3-USM01). If this extension is not found, it uses one of the extensions defined in columns 4,5, or 6 of this table instead.
Column 2 Limits access to signed-in users by means of an asterisk (*). This is relevant for Web OPAC only.
Column 3 is the base name that is displayed to the user.
Column 4 lists the alternative extensions to the filename. If there are no OPAC files with the extension defined in column 1, the system uses the files which have an extension defined in columns 4, 5, or 6.

Columns 4-6 are also used for referring to data format tables (`edit_doc`, `edit_doc_999.lng`, `edit_doc_999_<aut_library>lng`). These tables can be duplicated, with a base extension (for example, `edit_doc_999.eng.mono`).
Column 7 is the code of the physical library in which the logical base is defined.
Column 8 Column 8 defines whether or not Z0102 is used to create a logical base.
Column 9 is the Find command that the system uses to create a logical base of records. Once this base exists, all Scan and Find queries that the OPAC user performs are carried out on the records in the base (and all other records are excluded). In the above example, the HISTORY base includes all records that include the word "history".

To set the bases that are available in (the drop-down list for the Bases field in the Find and Browse nodes), follow these steps:

1. Open the ALEPHADM client.

2. From the Configuration menu, click **Libraries Manager**. The Libraries Manager window is displayed.

3. Open the fourth tab, Bases for Search, and add or delete lines. The values in Base for Search and Library Code must match values in the system's `tab_base.lng` table.

4. Click **OK** to save the changes. The client's `alephcom\tab\searbase` file is updated.

In the Browse node, only the headings that are relevant for the logical base are displayed. An exception to this is when the code of the search database is the same as a library code (for example, base code "USM01"). In this case, the Browse list will include all headings, both linked and unlinked. In order to filter the Browse list to include only linked headings, use a base code that is different than the library code (for example, USMXX).

3 Find Queries

To define the types of Find Queries that are available to the user and the order in which they are displayed, go to the ALEPHCOM/TAB directory on the client and

open the ALEPHCOM.INI file. Go to the section headed [SearchFind]. Following is a sample of the relevant line:

```
FindTypes=S,C,F
```

In this example, all types of Find Queries are available, Advanced **S**earch, Multi-base **C**CL Search and Multi **F**ield Search. The corresponding tabs are displayed in this order (S,C,F), from left-to-right. To change the order in which the tabs are displayed, change the order of the letters above. To make an option unavailable to the user, remove its letter from the list above. Note that the letters are written with commas between them, but without any spaces.

Another line in [SearchFind] is:

```
SelTab=C
```

The content of this line is filled automatically. It defines the last Find query that was used the last time that the GUI client was closed.

In the Search function, the Advanced, Multi-base CCL and Multi-field queries allow the user to select the Words Adjacent option. This instructs the user that the words in the search string must be in the same field (word index) and appear next to each other in the record.

You can determine whether or not the "Words Adjacent" option is selected by default when the user activates the Find Query screen. To do this, open the ALEPHCOM.INI file on the client and go to the section headed [SearchFind]. Following is a sample of the relevant line:

```
FindAdjacency=Y
```

If you want the "Words Adjacent" option to be selected by default, type **Y** to the right of the equal (=) sign. If you do not want the "Words Adjacent" box to be checked by default, type **N**.

Note that this also controls the default setting of the "Words adjacent" option that appears on the right-hand side of the Refine screen.

The AutoShow parameter in the same section sets the threshold of the number of documents that are displayed in Brief or Full format when FIND is used. If the set contains the same number of records, or less than the number set in AutoShow, the records are automatically displayed in full format in the Show node after a FIND query is run. Otherwise, the results appear in a set in the List of Sets in the Find node.

3.1 Boolean Operators Setup

You can set up different Boolean operators according to different interface languages in the `./alephe/tab/tab_ccl_boolean_operators` table. This allows you to use different operators, based on language, other than the standard AND, OR and NOT. For example, you can add the Danish word *Eller* which corresponds to *or*.

```
!1          2              3
!!!-!!!!!!!!!!!!!!!!!!!-!!!>
### |                      OR
### ~                      NOT
### +                      AND
### &                      AND
DAN IKKE                   NOT
DAN OG                     AND
DAN ELLER                  OR
```

Column 1 contains a language code, column 2 contains a string and column 3 contains a CCL operator.

4 Holdings Information

When you click the *Items* button on the Full record, the List of Items is displayed with holdings information such as Description, Item Status, Sublibrary, Collection and Call Number. In addition, if the item is currently on loan, the due date is shown.

4.1 Holdings Note for Serials

The USMARC 866 holdings note field which appears in the holdings record may be displayed in the Full view of the Bibliographic record in the Search function. To set this up, follow these steps:

1. To ensure that these records will be merged (for display purposes only), edit the `tab_expand` table. Make sure that the following line appears in the table:

   ```
   GUI-DOC-D expand_doc_bib_hol_USM
   ```

2. After editing the `tab_expand` table, to ensure that the 866 field is included in the Full view of the BIB record, edit the `edit_doc_999.lng` table.

5 Displaying and Indexing Copy Information

Information about copies can be displayed together with the linked bibliographic record in special ALEPH fields called Z30-1, Z30-2, LOC, PSTS and PST (depending on the program used to compile the information). These fields do not actually exist in the bibliographic record; rather, the system compiles the information from the subscription, holdings and item records and displays it in the bibliographic record as if they were regular fields. Therefore, the Z30-1, Z30-2, LOC, PSTS and PST fields are referred to as "virtual fields."

These virtual fields may also be indexed so that the user may search them using the Find or Browse options in the same manner that he can search subjects, titles, and authors.

There are four different programs for expanding copy information into the bibliographic record. Each program does this in a particular way and the library should decide which program best suits its needs. Following are the existing programs:

- expand_doc_bib_z30
- expand_doc_bib_psts
- expand_doc_bib_loc_usm
- expand_doc_bib_loc_n (together with expand_doc_sort_x)

Refer to Indexing - Chapter 9 Expand Routines, Tables and Indexing Expanded Fields (Expand Record section) for a complete explanation of these programs and how to set up the relevant table (`tab_expand`).

Note that for the indexing and display of copy information, it is necessary to set up all the relevant tables:

- For indexing purposes: `tab00.lng`, `tab01.lng`, `tab11_acc`, `tab11_ind` and `tab11_word`.
- For display purposes: `edit_field.lng`, `edit_doc_999.lng`, and so on.

6 Column Headings

You may define the column headings and column widths for tables of information. To do so, you need to edit one of three files, depending on the type of screen.

6.1 pc_tab_short.lng - Brief List

The columns for the Brief List are defined by editing the `pc_tab_short.lng` file. Following is an extract from the file:

```
!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 01
C01
```

Column 1 is the display format. The standard is 00.
Column 2 is the text of the column heading that is displayed to the user.
Column 3 is the column number (from left to right).

Columns 4-8 are for tag + indicator + subfield. You may enter up to five field tags. If the system does not find the first field in the record, it will look for the alternatives

you define here. The **#** character may be used as a placeholder for indicators in positions 4 and 5. For example, entering *245##* designates *2451*, *2452* and *24514*. BASE1/BASE2 are system codes that can be used to display whether the record is present in a particular logical base. The logical bases for each column are configured in `$alephe_tab/tab_base_count`, col. 4 and 5.

Column 9 is the percentage of the page width that you want the column to take up. The number must be 3 digits. For numbers less than 100, enter leading zeros, for example, *025*. You may remove a column heading from the table by entering *000*.
Column 10 is the font as defined in the FONT.INI file on the PC. The relevant lines in font.ini start with ListBox. If the line has ListBox## in the first column of font.ini it will relate to all font numbers. That means they will all be displayed, some according to the line starting with ListBox##.
Column 11 is the color as defined in the ALEPHCOM.INI file on the PC. The relevant section is [TextListBox]. Please note that after effecting changes you have to restart the server to see your new color setup.
Column 12 is the starting character, that is, the position number of the first character in the field that should be displayed. In the example above, the first character of the Year field that will be displayed is the character in position number 8. The number you enter must be 3 digits. For numbers less than 100, enter leading zeros, for example, *008*. If you want the starting character to be the character in position number one, leave this column blank.
Column 13 is the total number of characters of the field that should be displayed. In the example above, four characters of the Year field will be displayed, those in positions 8, 9, 10, and 11. The number you enter must be 3 digits. For numbers less than 100, enter leading zeros, for example, *004*. If you want the entire contents of the field to be displayed, leave this column blank.

Column 14 contains a language code. It will act as a filter on the field. It uses a subfield 9 that contains a language code as a filter.

6.2 pc_tab_col.lng

`pc_tab_col.lng` defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library table `pc_tab_col.lng`.

For more information about `pc_tab_col.lng`, see the discussion of GUI lists and column headings in the ALEPH User Guide - General chapter - Using and Customizing Your GUI.

The following is a list of the Search Function windows which use `pc_tab_col.lng` for formatting data, and their identifiers (Column 1 in `pc_tab_col.lng`).

Identifier	**Search Function Windows**
CORRECT_HEADING_1	Correct Heading
CORRECT_HEADING_2	Correct Heading

PC_COM_DOC	Navigation Tree
PC_SEAR_ADM_LIBRARY	Global Items List of Record <record #>
PC_SEAR_FULL_000	MARC Tags (tab01)
PC_SEAR_FULL_99	Full+Link (edit_doc_999)
PC_SEAR_FULL_XXX	Catalog Card (edit_doc)
PC_SEAR_HEADINGS	Brief Records list
PC_SEAR_ITEM_LIST	Item list
PC_SEAR_MARC_OPAC	Document code text
PC_SEAR_REFERENCE	Reference List
PC_SEAR_SCAN	Browse List of <ACC name>
PC_SEAR_SCAN_I	Browse List of <index name>
PC_SEAR_SCAN_W	Browse List of <Word index>
PC_SEAR_SCAN_Z39	Browse List of <Z39 index>

Note: the definition for the List of Sets and Cross Set windows are in the pc client in: `ALEPHCOM\tab\<language>\tab_col.dat`.

6.3 tab_col.dat

In order to define column headings for the List of Sets and Cross Set screens, go to the client's `ALEPHCOM\tab\ENG` directory (for English headings; other headings are located in the appropriate `ALEPHCOM\tab\<language>directory`.) Open the `tab_col.dat` file. Following is a sample of the relevant lines in that file:

```
1                 2 3             4  5   6  7   8 9 10

SEAR_CROSS_SET    L Find Request  01 060 01 C04     Find request
SEAR_CROSS_SET    L Base Name     02 020 01 C02     Base name
SEAR_CROSS_SET    L No. Records   03 020 01 C03     No. records
SEAR_RESULT_LIST  L Request       01 050 01 C01     Request
SEAR_RESULT_LIST  L Database      02 030 02 C02     Database
SEAR_RESULT_LIST  L No. Records   03 020 03 C03     No. records
```

Column 1 is the unique code by which the system identifies this set of column headings. The code SEAR_CROSS_SET refers to Cross Set; SEAR_RESULT_LIST refers to the List of Sets, both from the Find node.

The definitions of the other columns are the same as `pc_tab_col.lng`. See the discussion of GUI lists and column headings in the ALEPH User Guide - General chapter - Using and Customizing Your GUI.

6.4 Multi-field Search

To define the list of fields for a Find Multi-Field query, edit the `pc_tab_sear.lng` file. There is a limit of 8 FF lines per base. Following is a sample from that file:

```
!           2            3              4                    5
!!--!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!
FF  USM01                L System number                   SYS
FF  USM01                L Titles                          WTI
FF  USM01                L Exact titles                    TIT
FF  USM01                L Authors                         WAU
FF  USM01                L Uniform Titles                  WUT
FF  USM01                L Barcode                         BAR
```

Key to the pc_tab_sear.lng Table:

- **Column 1 - Code**
  This is the unique code by which the system identifies the set of menu choices. To define Fixed fields to search in, enter FF in this column.

- **Column 2 - Code**
  Enter here the database associated with the menu option you are defining. You must use a database code that was assigned in tab_base.

- **Column 3 - ALPHA**
  ALPHA code. Must always be L.

- **Column 4**
  This is the text of the field that is displayed to the user.

- **Column 5**
  This is the index code for the search query taken from tab00.lng.

7 Menus

You can determine the choices that are available in the following menus of the Search function:

- List of indexes for the Find Query, Browse Query and Direct Query.
- List of formats for displaying and printing bibliographic records.
- List of sort order options.

7.1 Index Menus

To define the list of indexes for Find and Browse queries, edit the `pc_tab_sear.lng` file. Following is an extract from that file:

```
!1          2            3              4                    5
!!--!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!
SC  USM01                L Titles                          TIT
SC  USM01                L Related Titles                  TRT
SC  USM01                L Authors                         AUT
```

```
SC  USM01               L Series                        SRS
SC  USM01               L General Subjects              SUB
FI  USM01               L Anywhere in the record        WRD
FI  USM01               L System number                 SYS
FI  USM01               L Titles                        WTI
FF  USM01               L System number                 SYS
FF  USM01               L Titles                        WTI
FF  USM01               L Exact titles                  TIT
FF  USM01               L Authors                       WAU
FF  USM01               L Uniform Titles                WUT
FF  USM01               L Barcode                       BAR
```

Column 1 is the unique code by which the system identifies the set of menu choices. For the Find Query menu, enter the code FI; for the Browse Query menu, enter SC; for the fixed field (Multi-Field) Query menu, enter FF;
Column 2 is the database code. Enter here the database associated with the menu option you are defining. You must use a database code that was assigned in `tab_base.lng`.
Column 3 is the code for the character set of the menu choice. The standard is L, which stands for Latin.
Column 4 is the text of the menu choice that is displayed to the user. Enter here the name of a word index that was defined in `tab00.lng`.
Column 5 is the code for the above index as it was defined in `tab00.lng`, column 2.

7.2 Formats Menu

To define the list of formats for displaying or printing bibliographic records, edit the `pc tab_sear.lng` file. Following is an extract from that file:

```
!1          2            3              4                    5
!!--!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!
FO  USM01               L Full+Link                     099
FO  USM01               L MARC Tags                     001
FO  USM01               L MARC Public View              002
FO  USM01               L Catalog Card                  037
FO  USM01               L Citation                      040
PF  USM01               L Name tags                     002
PF  USM01               L MARC tags                     001
PF  USM01               L MARC Public View              099
PF  USM01               L Catalog Card                  037
PF  USM01               L Citation                      040
```

Column 1 is the unique code by which the system identifies the set of menu choices. To define **display** formats in the Show node, enter FO in this column. For **print** formats, enter PF. For the **record** formats available when clicking the **Print/Send** button, enter PF.
Column 2 is the database code. Enter here the database associated with the menu option you are defining. You must use a database code that was assigned in `tab_base.lng`. **Column 3** is the code for the character set of the menu choice. The standard is L which stands for Latin.
Column 4 is the text of the menu choice that is displayed to the user. Available options include: Full+Link, Catalog Card, MARC Tags, and Name Tags.
Column 5 is the code for the above display format. Enter 099 for Full+Link, 037 for Catalog Card, 001 for MARC Tags and 002 for Name Tags. Codes are defined in `edit_doc.lng`.

7.3 Sort Order Menu

The user may choose to sort bibliographic records by specific fields. To define the list of fields that the user may choose from, edit the pc_tab_sear.lng file. Following is a sample from that file:

```
!1          2            3              4                    5

!!--!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!
SO  USM01                L Year                               01
SO  USM01                L Call number                        04
SO  USM01                L Author                             02
SO  USM01                L Title                              03
SO  USM01                L Type                               06
```

Column 1 is the unique code by which the system identifies the set of menu choices. For the Sort Order menu, always enter SO.
Column 2 is the database code. Enter here the database associated with the menu option you are defining. You must use a database code that was assigned in tab_base.lng.
Column 3 is the code for the character set of the menu choice. The standard is L which stands for Latin.
Column 4 is the text of the menu choice that is displayed to the user. Enter here the name of a field that was defined in `tab_sort`.
Column 5 is the code for the above field as it was defined in column 1 of the `tab_sort` table.

For example, the following line appears in `tab_sort`:

```
!1 2    3     4   5   6     7   8   9    10   1  12    13   4  15
16   7 18 9
!!-!!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-
!!!!!-!-!!-!!
01 95 008           260## c
08 04
```

Column 4 of `pc_tab_sear.lng` will be Year and column 5 will be 01.

8 Locate Function

The Locate function of the Search function enables you to find records in other databases that are similar to the record you have selected. As System Librarian, you are responsible for setting up the criteria that the system uses in order to determine which records are similar. (For example, your library can decide that if the records have the same words in the title and author fields, then the records are "similar.")

You can define the criteria by editing the `tab_locate` table. For more information see the Locate function in the Cataloging guide.

For example, the following lines determine the fields that will be checked for similarity when searching the Library of Congress:

```
!          1             2             3                4
5
                          6
!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!
!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
LOC                   245## a                     wti=
locate_str_1
LOC                   100## a                     wau=
locate_str_0
```

Column 1 contains the database code as it appears in the first column of tab_base.lng. Columns 2 and 3 contain the fields and subfields that will be checked for similarity. The Find command resides in Column 4. Column 5 contains the Extract function that defines how the contents of the field are handled.

To determine the database that will be available for the Locate function, edit the `ALEPHCOM\TAB\LOCATE.dat` file. The databases on the above list are marked automatically by the system according to the user's last choice. The parameter that defines the marked databases can be found in the `ALEPHCOM\TAB\ALEPHCOM.ini` file:

```
[SearchLocate]
SelectedBase=LOC
```

This means that the Library of Congress database will be marked when the Locate window is opened.

9 Default Text for Links

The cataloger uses the LKR field to create links from the bibliographic record he is currently cataloging to another record. The following table shows several examples of links that can be created by the cataloger and appended to a record in the LKR field:

Subfield	Description	
a	Value	Use
	UP	"Up" link to another BIB type record. A record may have only one link of this type.
	DN	"Down" link to another BIB type record. Multiple links of this type are possible.
	PAR	Parallel link from BIB record to BIB record
	ANA	Analytic link between BIB records of different levels
	ITM	Link from BIB record to ADM record (used for items bound together)

When no other text is available (the Z103-LKR-TEXT-N field is empty), the default text that appears next to the LKR (link) field in the Full View of the bibliographic record is defined through the library's `tab/tab05.lng` table as follows:

```
! 1                        2
```

```
!!!!!------!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
NONE       "text"
```

For example:

```
! 1                        2
!!!!!------!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
NONE       Click to view related record
```

NONE is reserved for the Search function both from GUIs and from the Web OPAC.

The LKR fields can be displayed according to various sorts. For more information see tab_fix_z103.

10 Search Function Tables

GENERAL

1. Table Extensions

You can use different tables for the Search function in GUIs and in the Web OPAC by using the file extensions:[filename].PC and [filename].WWW

Example:

`tab_sub_library.lng.PC` or `tab_sub_library.lng.WWW`

LIBRARY TABLES

1. edit_doc.lng

`edit_doc.lng` is used to define the display of document information. It defines a concatenation of a number of paragraphs. It is used in conjunction with `edit_paragraph.lng`.

2. edit_doc_999.lng

`edit_doc_999.lng` defines the fields to be included in a display of a bibliographic record. It defines the full format for printing and saving in GUIs and in the Web OPAC.

This display includes hypertext linking.

3. edit_doc_999_aut_xxx1n.lng

`edit_doc_999_aut_xxx1n.lng` defines the display of AUT records for cross-reference display. The display of the record is defined in the `edit_doc_999_aut_xxx1n.lng` table where "`xxx1n`" is the code of the AUT library (for example, USM10).

4. edit_field.lng

edit_field.lng defines the display of a tag. It is used in conjunction with edit_paragraph.lng and edit_doc.lng. If a tag is not defined in edit_field.lng, it will be displayed as is. Accordingly, tags entered in the database without punctuation need to be defined here. You can define up to 10 filters for each subfield per tag name.

5. edit_paragraph.lng

edit_paragraph.lng defines the display of a paragraph of data from the bibliographic record. The edit_paragraph table defines a concatenation of a number of fields. It is used in conjunction with edit_field.lng and edit_doc.lng.

Note that formats 001-099 are used by the BIB_FORMAT program for direct display of specific BIB information (for example, on Patron List of Loans, Search function of Items, and so on.)

6. pc_tab_sear.lng

This table is used to define options for the Search function search as follows:

- **SC** - link to the ACCess code as defined in tab00 in the tab directory of the library.
- **FI** - link to the WORD files as defined in tab00 in the tab directory of the library.
- **FO** - link to edit_doc format (must be expressed in three 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** - fields for Find - Multi-field Search.

7. pc_tab_short.lng

This table defines the document fields that are included in the Brief display in the upper pane of the Show node.

8. tab00.lng

tab00.lng defines the system index files. There should be one such table for each language defined.

9. tab01.lng

`tab01.lng` defines per language the type and name of the library and tag definitions for the bibliographic records.

10. tab05.lng

This table defines the caption to be displayed in the Web OPAC before $$n and $$m.

11. tab100

`tab100` is the central configuration table for system-level, server-level and library-level variables.

Example:

`AUTO-TRUNCATE-Z01-FIND`
This variable controls the automatic truncation option in Browse.

`CORRECT-HEADING-ORDER`

This variable defines the order of subfields when adding or updating a specific subfield of a heading via GUI-Search-Browse - Correct heading.
Possible values: 1 or 0.

1 (default) – The correct heading of an existing subfield remains in the original order. The new subfield is added at the end of the field.
0 – The new or modified subfield is added at the beginning of the field.

12. tab11_acc

The `tab11_acc` table is used to assign fields to headings indexes.

13. tab11_ind

The `tab11_ind` table is used to assign fields to direct indexes.

14. tab11_word

The `tab11_word` table is used to assign fields to word indexes.

15. tab20

This table defines the headings (including see references) that are built in the ACC table of the bibliographic base, based on the fields of the authority record that "matches" the BIB ACC Headings.

16. tab22

This table defines the fields that are included in the "short-doc" (z13) - and also defines the fields in the Basket Brief format.

17. tab_aut

This table defines which heading files are linked to an authority database and which authority database is searched.

18. tab_filing

The `tab_filing` table defines which filing procedures are used when building the filing key for Heading (Z01), Index (Z11) entries and Sort keys (Z101). The filing procedures identifier of these is set in col. 5 of `tab00<lng >`.

19. tab_sort

`tab_sort` defines fields for sorting. This table is used in conjunction with:

- `pc_tab_sear.lng` (option SO)
- option-sort HTML screen (for Web OPAC)
- `www_server.conf`
- `pc_server_defaults`
- `tab01.lng` - document record fields (for correct filing of z101 sort keys).

20. tab_word_breaking

This table defines routines for filing of headings - word breaking procedures.

21. tab_z30_sort

`tab_z30_sort` defines items (z30 sort types) in various ALEPH functions. It is located in the library's `data_tab` directory. The instance which affects the Search function is SEARCH.

System Librarian's Guide - Serials

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 PRINT TEMPLATES 4

2 COLUMN HEADINGS (PC_TAB_COL.LNG) 5

3 DROP-DOWN MENUS 6

4 ITEM ARRIVAL FORM 8

5 UPDATE OF SERIALS ITEMS PROCESS STATUS 9

6 CALL NUMBER OPTIONS 10

7 ITEM STATISTIC OPTIONS 11

8 DEFINING 853 CHRONOLOGY AND NUMBERING TERMS 12

8.1 853_Chrono 12

8.2 853_numbering 13

9 SORTING OF ISSUES AND ITEMS 13

10 ROUTING LIST 15

11 SETTING UP BINDING ALERT REPORTS 15

11.1 The Binding Alert Process 17

12 CLIENT SETUP (ACQ.INI) 17

13 SERIALS TABLES 19

1 Print Templates

The names of the Print template files for Serials are as follows (where *nn* is a number link identifying the version). These files can be found in the `form_lng` directory of the BIB library. To read more about printing, go to the Printing chapter.

GUI Templates

Template	Purpose	print.ini function name
Item Label serial-item-label-nn	The label which is printed during check-in (if so indicated for that subscription).	ItemArrival
Regular Label item-issue-label-nn nn=These numbers run from 00-19	The label which is printed when you highlight an issue-type item and click Print Label on the Arrival form.	ItemLabel
Regular or Spine Label item-issue-label-nn nn=These numbers run from 20-39	The label which is printed when you highlight an issue-type item and click Print Label on the Arrival form.	ItemLabel
Claim Letter (GUI) serial-claim-letter-nn nn=These numbers run from 01-04	The claim form is produced when you claim an item from the Check-in List of the Serials module.	ClaimLetter
Routing List serial-rout-list.xsl	The list of routing list members is printed during check-in for titles that are routed. Also the routing list printed when you select a routing group on the Routing List display in the upper pane and click Print.	RoutList RoutReturn ItemArrival
Subscription Information copy-info.xsl	The Subscription Information form is printed when you click Print on the Subscription list in the subscription display in the upper pane.	CopyInfo

Services Templates

Template	Purpose
Arrived Issues arrived-issues.xsl	The report lists the serial items that have arrived in a defined period by the Print Arrived Issues (serial-06) service.
Claim Letter/Report (Services) batch-serial-claim-letter-00	The claim form used when claiming is carried out by the Print Serials Claim Letters (serial-44) service.
Routing Letter	The letter sent to members of routing lists by the Letters to

serial-rout-letter-nn	Routing List Members (serial-08) service.
Claim Report for Unreceived Serial Orders acq-serials-claim-report.xsl	The form which is sent to vendors to claim serial titles on an order where no issues at all were received. (This is not the same as claiming for individual issues). Printed by the Claim Report and Letters for Serials Orders (acq-19) service.
Subscription Renewal acq-subs-renewal-list-nn	The subscription renewal letter sent to vendors by the Subscription Renewal Letters (acq-23) service.
List of Expected Issues serial-expected-arrival.xsl	The report lists issues that are expected to arrive in a given time period (serial-17).

2 Column Headings (pc_tab_col.lng)

`pc_tab_col.lng` defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (USM01) table `pc_tab_col.lng`.

For more information about `pc_tab_col.lng`, see the ALEPH User Guide - General chapter - Desktop Customization - GUI and Toolbars section.

The following is a list of the Serials windows which use `pc_tab_col.lng` for formatting data, and their identifiers (Column 1 in `pc_tab_col.lng`).

Unique Code	**Serials GUI Panes**
PC_SERIAL_CLAIM_L_S	Claim List (under Subscription root)
PC_SERIAL_CLAIM_L	Claim List (under Check-in root)
PC_SERIAL_COPY_L	Subscription List
PC_SERIAL_HOL_LIST	Subscription Information Pane/HOL Links tab
PC_SERIAL_CHECKIN	Check-in List *
PC_SERIAL_GROUPS	Group List/All Items tab
PC_SERIAL_GROUPS_SL	Group List/ Sublibrary Groups/tab
PC_SERIAL_GROUPS_COL	Group List/Sublibrary/Collection Groups tab
PC_SERIAL_GROUPS_VEN	Group List/Vendor/Sublibrary tab
PC_SERIAL_GROUPS_ORD	Group List/Order Information tab
PC_SERIAL_CAPTIONS	Pattern List/Patterns tab
PC_SERIAL_PATTERNS	Pattern List/Level-X tab
PC_SERIAL_SCHEDULE	Pattern Information/Expected Schedule tab

Unique Code	Serials GUI Panes
	Schedule Root/Expected Schedule pane
PC_SERIAL_ARR_LOG	Serial Activity Log
PC_COM_D_TRIG_LIST	List of Triggers
PC_ITEM_ITEM	Items List
PC_SERIAL_ROUT_L	Routing Lists
PC_SERIAL_ROUT_CL	Routing Lists (under Subscription root)
PC_SERIAL_MEMBER_L	Routing List Members List
PC_ACQ_LOGGER	Subscription Log List

* In these GUI tables, an optional color/font can be used by the system for color/font differentiation between values in the same column. The alternative font and color can be defined in Columns 8 and 9 of `pc_tab_col.lng`.

3 Drop-down Menus

You can add or delete choices for some of the menus of the Subscription Information pane and on the Item Information pane by editing the following ADM library tables:

- pc_tab_exp_field_extended.lng
- pc_tab_exp_field.lng
- ALEPH ADM tables:
    - tab40.lng
    - tab25.lng
    - tab15.lng

pc_tab_exp_field.lng
The following list shows the `pc_tab_exp_field.lng` entries for drop-down menus which appear in various panes in the Serials function.

pc_tab_exp_field.lng	GUI Fields
Serial Bar	
SERIAL-SEARCH	Search for Serial Record by *
Control	
SERIAL-GRP-SORT-TYPE	Sort Options for Group list
SERIAL-CHK-SORT-TYPE	Sort Options for Check-in list

pc_tab_exp_field.lng	GUI Fields
SERIAL-PATTERN-CONT	Cont.
SERIAL-PTRN-REG-MODE	Regularity (first component)
SERIAL-PTRN-REG-TYPE	Regularity (second component)
SERIAL-CLM-SORT-TYPE	Sort Options
SERIAL-PATTERN-FREQ	Frequency
Subscription	
SUBSCR-SORT-TYPE	Sort Options
BIND-DISCARD-TYPE	Discard/Binding
ITEM-LOCATION_TYPE	Call Number Type + 2nd Call Number Type
Claims	
SERIAL-CLAIM-FORMAT	Format
SERIAL-CLM-SORT-TYPE	Sort Options
VENDOR-CLAIM-REPLY	Claim Reply
Routing Lists	
ROUT-GROUP-STATUS	Group Status
Triggers	
TRIGGER-DEPARTMENT	Department
Schedule	
TIME-TYPE	New Volume Every New Issue Every

* You can add **Bibliographic** search values. For example, you can add the BIB title by adding the following entry to the SERIAL-SEARCH section of the `pc_tab_exp_field.lng` table:

```
SERIAL-SEARCH        L-BIB Title         BIB-TIT
```

Note: TIT in this example must be an index code (defined in column 2 of `tab00.lng`) that is used by the system. You might also want to add other search values.

pc_tab_exp_field_extended.lng

The following list shows the `pc_tab_exp_field_extended.lng` entries for drop-down menus, which appear in the Subscription Information pane and in the Item Information form of the Serials function.

For entries marked by an asterisk (*), by selecting the following GUI fields, the windows mentioned below will appear. Clicking the arrows to the right of the fields in the windows mentioned below will display the menus.

See also the Call Number Options on page 10 and the Item Statistic Options on page 11 for setup options for the Call Number and the Statistics fields.

pc_tab_exp_field_extended.lng	GUI Fields	Window
Subscription and Item		
ITEM-LOCATION	Call Number	
ITEM-LOCATION-2	2nd Call Number	
* LOCATION-GR	Call Number + 2nd Call Number	Build Call Number Field
* LOCATION-USIG	Call Number + 2nd Call Number	Build Call Number Field
* LOCATION - PREFIX	Call Number + 2nd Call Number	852 Form
* LOCATION-SUFFIX	Call Number + 2nd Call Number	852 Form
* LOCATION-SOURCE	Call Number + 2nd Call Number	652 Form
Item		
INVENTORY-NUMBER	Inventory Number	

ALEPHADM Tables

- tab40.lng - This table is used to define the drop-down menu of Item Collection
- tab25.lng - This table is used to define the drop-down menu of Material Type
- tab15.lng - This table is used to define the drop-down menu of Item Status
- tab45.lng - This table is used to define the drop-down menu of Transaction Type

4 Item Arrival Form

You can determine whether or not the Item Arrival Form is displayed when the user checks issues in by any of the three check-in methods ("Fast", Publication Schedule, and Items window).
This can be set by the pc_server_defaults parameter: modify_item_arrival
For example: setenv modify_item_arrival Y

Value can be Y or N:
Y (default) – Item Arrival Form is displayed for each checked-in item.

N – Item arrival form is not displayed per item. The system registers the arrival of all grouped items (one-action check-in).

If modify_item_arrival is set to N (all items are checked-in in one action), an additional pc_server_defaults parameter is consulted: arrival_group_order.

This variable controls the order in which items are checked-in.

The arrival_group_order parameter is relevant for check in action made for Group: Vendor/sublibrary (tab 4 in Group node), Vendor Groups (tab 5 in Group node), and Order Information group (tab 6 in Group node). These three groups enable a partial arrival so that not all grouped items are necessarily arrived. The arrival_group_order parameter controls the order in which items are checked-in.

For example: setenv arrival_group_order 0

Value can be 0 or 1:
0 – (default) Items are checked-in by their sequence number (Z30-ITEM-SEQUENCE).

1 – Items are checked-in by item's copy ID order (Z30-COPY-NUMBER).
If part of the items is set with a null in the Copy Number field, the ones that have a value are prioritized. The items without the copy number are ordered by their sequence number (Z30-ITEM-SEQUENCE).

If modify_item_arrival is set to Y (each item is separately registered using a separated form), the system does not consult the arrival_group_order parameter.

If modify_item_arrival is set to N (all items are checked-in in one action), an additional pc_server_defaults parameter is consulted: display_check_in_note
This variable controls the display of the subscription's check-in note.

For example: display_check_in_note Y
Value can be Y or N:
Y – (default) The subscription's check-in note is displayed.
N - The subscription's check-in note is not displayed.

If modify_item_arrival is set to Y (each item is separately registered using a separated form), the system does not consult the display_check_in_note parameter.

5 Update of Serials Items Process Status

You can set the system to automatically change the issue item process status when various serial actions are performed. To do so, edit tab42 in the Administrative library. Make sure that Col. 1 is set with SERIALS.

The serials action (col.3 of tab42) can be:

OP – Item opened
NO – Item opened, when Z16 is not active
CL – Item claimed
UAR – Unarrive item
AR – Item arrival registered

PI – Item arrival registered influence already arrived issues.
NP – Item not published (this is not a trigger,
it is a manually entered processing status.

For example:
When an issue is checked in, the item process status can be updated depending on the pre-defined setup of tab42.
In addition, you may set tab42 so that the check in issue that just arrived will invoke a change in the process status of the previous issue that arrived of the same subscription.

Setting a different process status for items which currently arrived or previously arrived enables the library to define a separated circulation policy for those items (tab15.<lng> and tab16 of the Administrative library).

To implement this sample, set AR (Item arrival registered) and PI (Previous Issue Registered) actions in col.3 of ./xxx50/tab/tab42.

```
---1---------2----3-4--5--
SERIAL     ##### AR NW
SERIAL     ##### PI    NW
```

In this setup, when the check-in is performed, the system automatically performs the AR (Item arrival registered) and PI (Previous Issue Registered) actions.
The first line is read and the item process status of the current arrived item becomes NW (new item, or any other process status defined in Col.4).

If the second line is set (see above sample setup), then the item process status of the previously arrived item is updated to null (or any other process status defined in Col.4).
Col.5 is set to ensure that the process status is changed only if the current process status is NW (or any other process status defined in Col.5).
Use ## in Col.5 to indicate that the current item process status is irrelevant, and that the change is to be made on the previously arrived item in all cases.

Note: Col.5 only applicable when column 1.is ACQ or SERIAL with PI action in Col. 3.

6 Call Number Options

Two forms in the Serials function include Call Number information fields: the Subscription Information Form and the Item Form. There are five options for entering data into the Call Number and Second Call Number fields in each of these forms, depending on the client setup:

- A drop-down menu with predefined call numbers from the `pc_tab_exp_field_extended.lng` table.
- No drop-down menu. Enter the call number directly into the field/s.
- Special form for libraries with Closed Stack management.

- An index with all previously used call numbers already entered. The Call Number Type must also be entered in this window.

- A special form for entering subfields directly into a MARC 21 Holdings 852 field (with the possibility of using the index call numbers as in type 4).

Which one of these options is available to you is defined in your client's `alephcom/tab/alephcom.ini` file, under the heading [ExpandField]:

```
[ExpandField]
;
; Describe the Expand button Status of special fields :
; 1 - Regular (pc_tab_exp_field_extended.lng) (default)
; 2 - Non Visible
; 3 - Special (for location=close stack)
; 4 - Special (for location=location index list)
; 5 - Special (for location=852 form)
; 6 - Special (for location=852 form, including special method for
calculating Author Number and Chronology Number). Valid only for
ItemLocation1Field and ItemLocation2Field.
; 7 - Special (for location=852 form, including special method for
calculating Author Number). Valid only for ItemLocation1Field and
ItemLocation2Field.

ItemLocation1Field=2
ItemLocation2Field=5
SerialLocation1Field=1
SerialLocation2Field=5
```

In this file, ItemLocation1Field and ItemLocation2Field refer to the Call Number and Second Call Number fields in the Item Form.

SerialLocation1Field and SerialLocation2Field refer to the Call Number and Second Call Number fields in the Subscription Information Form.

The digits to the right of each line refer to these five options.

7 Item Statistic Options

Two forms in the Serials function include Item Statistics information fields: the Subscription Information Form and the Item Form (in the General Information (2) tab). There are three options for entering data into the Item Statistic fields in each of the windows, depending on the client setup:

- No drop-down menu. Enter the data directly into the field.

- A drop-down menu with pre-defined options from the `pc_tab_exp_field.lng` table.

- A Special four-field form, with the drop-down menu for each field defined in pc_tab_exp_field.lng (ITEM-STATISTIC-1 to ITEM-STATISTIC-4).

Which one of these options will be available to you is defined in your client's `alephcom/tab/alephcom.ini` file:

```
; 1 - Simple edit (default)
; 2 - Regular (pc_tab_exp_field)
; 3 - Special form (4 part statistic field)
;
ItemStatisticField=3
SerialItemStatisticField=3
ItemStatisticDelimiter=/
```

In this file, ItemStatisticField refers to the Statistic field in the Item Form.

SerialItemStatisticField refers to the Item Statistic field in the Subscription Information Form.

The digits to the right of each line refer to these three options.

8 Defining 853 Chronology and Numbering Terms

The codes used for months and seasons in the ALEPH 853 field are defined, per language, in table 853_chrono in the alephe/tab directory.

Table 853_numbering, in the same directory, defines the numbering terms (or ordinals) used, per language, by these fields.

8.1 853_Chrono

```
!1  2   3
!!!-!-!!!!!!!!!!!!!!!!!!
eng M Aug.
eng M Sep.
eng M Oct.
eng M Nov.
eng M Dec.
eng S Spring
eng S Summer
eng S Autumn
eng S Winter

fre M janv.
```

Column 1 defines the language. Depending on the language code entered in positions 35-37 of the 008 tag for the title, the relevant month/season name in that language will be picked up for the description field of each issue. Be sure that the language you add here has also been defined in `./alephe_tab/allowed_languages`. If no language appears in tag 008, the default for English will be used instead.

Column 2 defines the type of chronological term: M for month, S for season.

Column 3 is the text you want the system to use for the month or season names. You can use up to 20 characters.

8.2 853_numbering

```
!1      2       3
!!!-!!!!!!!!!-!!!!!
eng 11        th
eng 12        th
eng 13        th
eng ?1        st
eng ?2        nd
eng ?3        rd
```

Column 1 defines the language. Depending on the language code entered in positions 36-38 of the 008 tag for the title, the relevant ordinal name in that language will be picked up for the description field of each issue. Be sure that the language you add here has also been defined in `./alephe_tab/allowed_languages`. If no language appears in tag 008, the default for English will be used instead.

Column 2 states the number (or group of numbers) for which the ordinal is defined. Question marks may be used as wildcards. In the example, if the number is 13, the ordinal will be "th". However, if the number is 23, 33, 43 etc. (i.e., '?3'), the ordinal will be "rd".

Column 3 is the ordinal that will be used for the numbers defined in column 2.

9 Sorting of Issues and Items

The sorting of items in the Items list display is affected by the relevant lines in the `tab_z30_sort` table in the ADM library. The Sort Option available in this window is based on definitions in the table, as seen below. The name used for each of the options in the drop-down menu is defined in `pc_tab_exp_field.lng` (line SERIAL-SORT-TYPE). Note that if neither enumeration nor chronology levels are present in a Z30 item record, its sorting uses the default value present in z30-description.

```
tab_z30_sort
!    1               2 3  4 5
!!!!!!!!!!!!!!!!!!!-!-!!-!-!!
SERIAL-0             A 03 D 00
SERIAL-1             A 01 D 01
SERIAL-2             A 00 D 03
SERIAL-3             A 02 D 04
SERIAL-4             A 06 D 06
SERIAL-5             A 08 D 06
SERIAL-6             A 04 D 01
```

Column 1 defines the sort option.

Column 2 defines the sorting order for **Issue** type items: A-Ascending or D-Descending.

Column 3 defines the sorting type for Issue type items. The following is a partial list of sorting types. For the full list, refer to the tab_z30_sort header.

```
!   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 - Special
!   ----------------------------------------------------
!        sublibrary name+
!        collection code+
!        chronological-i(year)+
!        chronological-j(year)+
!        chronological-k(year)+
!        description+
!        copy-id+
!   ----------------------------------------------------
!   08 - by 85x
!   ----------------------------------------------------
!        85x-type+
!        sublibrary+
!        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.
!   ----------------------------------------------------
!   09 - by vendor code
!   ----------------------------------------------------
!        vendor code+
!        sublibrary
```

Column 4 defines the sorting order for non-**Issue** type items: A-Ascending or D-Descending.

10 Routing List

When a routed item is checked in, the item is loaned to the Group Leader, and the system can perform circulation checks on this automatic loan. The checks are performed against the Group Leader to whom the loan is registered. Checks can be defined in the ROUT-LOAN section of `tab_check_circ` of the ADM library.

Note that the loan is performed whatever the result of the checks, but a new trigger record (Z104) is created for every inconsistency in the loan identified by the system.

```
!    1           2
!!!!!!!!!!-!!!!!!!!!!!!!!!!
ROUT-LOAN  check_circ_1_a
ROUT-LOAN  check_circ_1_b
ROUT-LOAN  check_circ_1_c
ROUT-LOAN  check_circ_2_a
```

11 Setting Up Binding Alert Reports

In order to generate the Binding Alert reports (bind-01 and bind-01), set up the following tables:

- tab_binding (under the ADM library)
- pc_tab_exp_field.eng (under the BIB library)

- sort tables (tab_expand_join, tab_expand, tab_sort under the BIB library)
- bib_format (under the ADM library)

tab_binding
Specify sets of parameters for binding or discard rules for journal subscriptions (as shown below)

Each set is identified by a unique two-digit number. This identifier is entered in the Discard/Binding field of the subscription form.

```
!1  2  3    4    5       6          7              8
!!-!!-!!!-!!!!-!!!!-!!!!!!!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!
01 01 999 0003 0001 high       standard   BINDER01
02 01 999 0003 0001 high       standard   BINDER02
03 01 003 0003 0001 low        high       BINDER02
04 01 999 0015 0001 high       basic      BINDER01
05 01 002 0003 0001 secondary  standard   BINDER01
06 01 002 0003 0001 secondary  basic      BINDER02
```

pc_tab_exp_field.eng
Define codes for sets of binding or discard rules in the BIND-DISCARD-TYPE menu as shown below.

Note

The codes defined in column 4 must be equivalent to the sets identifiers (column 1) in `tab_binding`.

```
!        1              2      3                                                  4
!!!!!!!!!!!!!!!!!!!!-----!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
BIND-DISCARD-TYPE        L Bind, using setup 1                                 01
BIND-DISCARD-TYPE        L Bind, using setup 2                                 02
BIND-DISCARD-TYPE        L Bind, using setup 3                                 03
BIND-DISCARD-TYPE        L Bind, using setup 4                                 04
BIND-DISCARD-TYPE        L Bind, using setup 5                                 05
BIND-DISCARD-TYPE        L Bind, using setup 6                                 05
BIND-DISCARD-TYPE        L Bind, using setup 7                                 07
BIND-DISCARD-TYPE        L Bind, using setup 8                                 08
BIND-DISCARD-TYPE        L Discard after one year                              90
BIND-DISCARD-TYPE        L Discard after two years                             91
BIND-DISCARD-TYPE        L No bind, Discard after five years                   92
```

Sort Tables
The report files which are generated by the Bind Alert Report (bind-01) and Serial Binding Slip (bind-02) services can be sorted by the Author +Title sort method, which results from a combined field. The `tab_expand_join` table is used to create a virtual field. This virtual field is used in tab_sort to create a new Z101 sort key. For example

tab_expand_join:
```
BND   1#### -e46  a     245## a     t
```

tab_expand:

```
SORT-DOC   expand_doc_join_permute
```

tab_sort:

```
07 01 BND          245## a     2                                    00 00
```

bib_format
The serial-bind-slip entry controls the content of the bibliographic citation in the report that is created by the service Serial Binding Slip (bind-02). For example:

```
serial-bind-slip              014
```

11.1 The Binding Alert Process

In the Discard/Binding field of the subscription form in the Acquisitions/Serials GUI, the staff user selects a coded value from the drop-down menu. This code defines the set of binding or discard rules that will be applied to the subscription.

From the Services menu, Binding submenu, the staff user runs the bind-01 and bind-02 services.

For each subscription that passes through the sublibrary and collection filters, the coded value in the Discard/Binding field is retrieved.

The system determines whether the Discard/Binding rule in the subscription record matches the service's parameters, by using `tab_binding`. The values in columns 6,7,8 of the equivalent rule (as in the Discard/Binding rule in the subscription record) are compared to the parameters submitted in the service. If a match is found, the issues linked to the subscription are analyzed for readiness for binding by checking the data in columns 3,4,5 of `tab_binding`.

12 Client Setup (acq.ini)

The following are sections in the acq.ini file which are unique to the Serials function:

[General]

`SerialFilterLogger=`
The **SerialFilterLogger** defines the Subscription Log filter parameters. According to the parameter selected, the system saves the number of transactions by which the Subscription Log was recently filtered. When the parameter has no value specified, "All" becomes the default value in the filter list.

`FilterCheckinSublibs=`
The **FilterCheckinSublibs** flag is used to define whether the sublibraries checked in the Control List pane will be saved and loaded.

`SerialEnableSchedule=`
The **SerialEnableSchedule** flag is used to define whether the Schedule node does or does not appear in the Serials Tree.

`SerialEnablePattern=`
The **SerialEnablePattern** flag is used to define whether or not the Patterns node appears in the Serials Tree.

`SerialEnableRouting=`

The **SerialEnableRouting** flag is used to define whether or not the Routing Lists root and the Routing Lists node under Subscription root in the Navigation Tree of the Serials Tab will be displayed.

[SerialBar]

The **[SerialBar]** section has the following parameter:

`IndexCode=`
Saves and loads last selection in the Serials Bar drop-down menu. For example:
`IndexCode = ADM Sys No`

[SerialTree]

The **[SerialTree]** section has the following parameters:

`BkColor=255,255,255`

Sets the Background color of the Serial Tree
`2=277`
`4=231`
`5=350`
`14=225`
`12=213`
`6=266`
`3=283`
`20=460`
The above lines refer to the positioning of the splitters between the panes.

Note: There is no need to change these settings.

[ItemList]

The **[Item List]** section defines the default sort for the Items List displayed in the Items tab. The options are set in the `pc_tab_exp_field.lng` table. The options themselves are set in `tab_z30_sort` in the ADM library.

This section has two parameters:

`ItemSortType=`
This setting is used for non-issue items; i.e., items whose material type does not begin with "ISS".

`SerialSortType=`
This setting is used for issue items; i.e., items whose material type begins with "ISS". The setting has to match one of the options defined in the ITEM-SORT-TYPE section of the ADM library's `pc_tab_exp_field.lng` table which in turn is based on definitions in the `tab_z30_sort` table.

[CheckInListExpand]

The **[CheckInListExpand]** section has the following parameter:

`ArriveAutoPrintLabel=`
The **ArriveAutoPrintLabel** flag is used to define whether or not a label will be printed while an issue is checked in by selecting the Arrive button on the Check-in List display.

The subscription record contains a flag which indicates whether or not to print a label at check-in time. If the **[ArriveAutoPrintLabel]** parameter is set to "Y", a message pops up when the issue checked in is linked to a subscription that has the label print unchecked. If the checked in issue is linked to a subscription that has the label print checked, and the parameter is set to "N", no label will be printed during the check-in process.

[**SubscrList**]

The **[SubscrList]** section has the following parameter:

`SubscrSortType=`

Defines the default sort type in the Sort Options field of the Subscription List pane and subscription list displayed in Acquisitions- Subscription List of serial order. It refers to the SUBSCR-SORT-TYPE section of table `pc_tab_exp_field.lng` of the ADM library. The available sorting options are: STANDARD, SUBLIB, VENDOR.

[**GroupList**]

The **[GroupList]** section has the following parameter:

`ArrivePrintLabel=`

The **ArrivePrintLabel** flag is used to define whether or not a label will be printed while an issue is checked in by selecting the Arrive button on the Group List display. The subscription record contains a flag which indicates whether or not to print a label at check-in time. If the **[ArrivePrintLabel]** parameter is set to "Y", a message pops up when the checked in issue is linked to a subscription that has the label print unchecked. If the checked in issue is linked to a subscription that has the label print checked, and the parameter is set to "N" , no label will be printed during the check-in process.

[**SerialClaimList**]

The **[SerialClaimList]** section has the following parameter:

`ClaimSortType=`

Defines the default sort type in the Sort Options field of the Claim List pane. It refers to the SERIAL-CLM-SORT-TYPE section of the table `pc_tab_exp_field.lng` of the ADM library. The available sorting options are: CLAIM-A, CLAIM-B, REPLY-A, REPLY-B.

[Messages]

DisplayRequestedMessage=Y

This section determines whether the staff user will be warned when a change is made in the status of a serial item if the item is requested. The warning will be issued when registering an item’s (or group’s) arrival, un-arrival, or when sending a claim letter for an item. Note that this variable affects also changes that are made on a non-serial order. Refer to the Orders System Librarian Guide for more information.

13 Serials Tables

ALEPH TABLES

1. **853_chrono**

   List of full (or partial) names of months and seasons, arranged by language, to be used when creating Publication Schedules by the 853/853X method.

2. **853_numbering**

   List of ordinals (for example, 1st, 3rd), arranged by language, to be used when creating Publication Schedules by the 853/853X method.

LIBRARY TABLES

1. **tab_85x_import**

   This table relates to the import of serials prediction fields (853, 853X, etc.) from other libraries into your own library. It defines the way in which fields imported from another library (using the b-serial-52 service) for checking against your serials records, will be used for matching.

   It is possible to define which index should be used for the match; whether the entire field from the input file or a specific subfield should be used for the match; and which subfield in that field in your own database should be used.

2. **tab_z30_sort**

   Sort options for the Items window in the Acquisitions/Serials GUI client. Use in conjunction with `pc_tab_exp_field.lng` (line SERIAL-SORT-TYPE).

3. **tab100**

   The variable UPDATE-Z30-COPY determines whether or not the Z16-copy-sequence is copied to the Z30-copy whenever an item is created for an issue.

   The variable UPD-SUBS-FROM-ORDER determines whether or not updating of serial information within the order record will affect the equivalent fields in the order's related subscription record. The relevant fields are: Vendor Code, Vendor Reference, Delivery Type, Subscription Start, and Subscription End Dates

   Values can be: Y or N.

   Y- Update of a serials order affects its subscription record (new enhancement).

   N- Default value - Update of a serials order does not affect the subscription.

4. **tab42**

   This table defines changes in the Item Process Status of issue-type items, as a result of events in the serial control process (for example, opening expected issues, claiming and check-in).

5. **tab45**

This table defines whether or not a transaction creates an order log record, and if the transaction is system-generated or can be entered manually.

6. **tab40**

   This table defines the expansions of the Collection Code 02 entered in the Collection field in the expanded form to be displayed in the Web OPAC.

7. **tab25**

   This table defines the possible values for Z30-Material (Item Material Type).

8. **tab15**

   This table defines Item Statuses and various parameters connected to these statuses.

9. **tab_binding**

   This table specifies sets of parameters for binding or discard rules for journal subscriptions. Each set is identified by a unique two-digit number. This identifier is entered in the Discard/Binding field of the subscription form.

10. **user_func.lng**

    This table includes serials functions which support authorization control.

11. **tab_checksum**

    `tab_checksum` is used to define the checksum programs. The programs can be used for creation, validation and correction of items barcodes.

    Note: For issue items generated by the Prediction Pattern process, you can use the following entry for the creation of the ISSUE-ITEM barcode.

    For example:

```
! 1                                    2
!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
CSCR-Z30-BARCODE-ISS       checksum_create_z30_by_doc_no
```

12. **tab_hld_stmt**

    This table is consulted when the procedures: **expand_doc_hld_stmt** or **fix_doc_hld_stmt** are invoked for generating summary holdings statements. Items with processing status NP (not published) or NA (not arrived) are not included in the summary statement.

    This table is used to map items with a process status different from NA or NP which are to be excluded from the summary holdings statement. This is done by mapping them as if they have a processing status of NP or NA.

    This mapping can be done for items that have a particular sublibrary and/or collection, item status, item process status, and/or break indicator.

For example: If a library wants to map the code of the process status MI (missing) as NA, the table will include the following line:

```
! 1     2   3  4  5 6
!!!!!-!!!!!-!!-!!-!-!!
##### ##### ## MI # NA
```

ALEPHE TABLES

1. tab_sublibrary

`tab_sublibrary` is used to define the list of sublibraries that are displayed in the Control Setup tab in the Control pane. Only sublibraries defined as type 1 and type 4 in column 2 are displayed.

System Librarian's Guide – Staff Privileges

Version 23

ExLibris The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2015. All rights reserved.

Document released: November 30, 2015

Web address: http://www.exlibrisgroup.com

Table of Contents

1 PASSWORDS 4

1.1 Default Password 4

1.2 Entering Passwords 4

2 FUNCTIONS AND SUBFUNCTIONS 4

3 DROP-DOWN MENUS 5

4 DROP-DOWN LISTS THAT CANNOT BE MODIFIED 6

4.1 Tab100 Variables 6

5 COLUMN HEADINGS (PC_TAB_COL.LNG) 7

6 ALEPH STAFF MENU PERMISSIONS 8

7 ACCESS RIGHTS TO NON STANDARD SERVICES 8

1 Passwords

The passwords for all libraries are contained in a single password file that is managed by a single administrative library. This means that the file is physically part of one library's Oracle tables and that when you want to update passwords through the GUI, you must connect to that library. The administrative library that manages the passwords file is defined in the `aleph_start` file in the /alephe_root directory. Following is an example of the line in the aleph_start file that defines the password library:

```
setenv    pw_library          USR00
```

The default password library is usually USR00 and must be changed during installation to one of the local libraries. Please consult Ex Libris before changing the library.

1.1 Default Password

Ex Libris has provided you with the default user name and default password for managing access rights in the Staff Privileges function. You may change the default password.

However, if you forget the new password of the default user, the only way to reinstate the default password is to delete the files that contain users and passwords (files Z66 and Z67), using UTIL A/17/1. Before deleting these files, consult Ex Libris.

Note that you may give other users the authorization to manage passwords and access rights.

1.2 Entering Passwords

The first time a user activates a module of the system, he will be asked to identify himself by entering his user name and password. The user name and password are then stored on the local disk in encrypted form and become the default for the client. In this way, the user does not have to enter his user name and password each time he uses the client.

However, if you **DO** want the user to identify himself each time the client is activated, go to the `ALEPHCOM/TAB` directory of the client and open the ALEPHCOM.INI file. Make sure that the SavePassword flag in the [Main] section is set to **N**.

2 Functions and Subfunctions

The list of functions and subfunctions to which users may be given access rights is located in the `user_function` file.

The following is a sample of the table:

PASSWORD	L Staff Privileges (Passwords)	GLOBAL	L	All Unspecified Subfunctions
PASSWORD	L Staff Privileges (Passwords)	USER-LIST	L	User list - Display
PASSWORD	L Staff Privileges (Passwords)	DUPLICATE-AUT	L	User - Duplicate
PASSWORD	L Staff Privileges (Passwords)	UPDATE-USER	L	User - Update
PASSWORD	L Staff Privileges (Passwords)	PERMISSIONS-GET	L	Access rights - Display
PASSWORD	L Staff Privileges (Passwords)	PERMISSIONS-PUT	L	Access rights - Update
PASSWORD	L Staff Privileges (Passwords)	PERMISSIONS-DELETE	L	Access rights - Delete
PASSWORD	L Staff Privileges (Passwords)	CHANGE-PASS	L	Password – Change

The columns in capital letters (columns 1 and 4) are codes used by the system to identify the function and access rights. Do not change these codes.

You may, however, edit columns 3 and 6 which appear in uppercase and lowercase letters (for example, "User Registration" and "All subfunctions"). The text in columns 3 and 6 is displayed in the Staff Privileges function accessible from all modules. The changes you make in these columns should be limited to changes of grammar, spelling or phrasing and should not alter the meaning.

3 Drop-down Menus

Following is a list of the fields on the Staff Privileges window that have drop-down menus. Note that in order to edit entries in the `pc_tab_exp_field.lng` file, be sure that you are first connected to the same administrative library that manages the passwords file.

Cataloger Level
You may add new cataloger levels by editing the `pc_tab_exp_field.lng` file. In that file, the menu is identified by the ID "LEVEL."

Circ. Override Level
You may add new circ. override levels by editing the `pc_tab_exp_field.lng` file. In that file, the menu is identified by the ID "CIRC. LEVEL."

Cat. Own ID
Cat. Own IDs are taken from the `tab_exp_own.lng` table of the library's tab directory. You can add new Cat. Own IDs by editing this table.

Note: The value entered in the Cat. Own ID field is checked against the values defined in the `tab_exp_own.lng` table. If the value does not appear in the table, it will be rejected and an error message will pop up.

Cat. Own Permission
Cat. Own Permissions are taken from the `tab_exp_own.lng` table of the library's tab directory. You can add new Cat. Own Permissions by editing this table.

Note: The value entered in the Cat. Own Permission field is checked against the values defined in the `tab_exp_own.lng` table. If the value does not appear in the table, it will be rejected and an error message will pop up.

Department
Staff department is taken from the `pc_tab_exp_field.lng` table of the library's tab directory. You can add a new Department by editing the STAFF-DEPARTMENT section this table.

4 Drop-down lists that cannot be modified

User Library
The values displayed in the drop-down list depend on the connected user. If an ADMIN "master user" is connected, the list will include all the ADM libraries in the system (i.e., libraries that are defined in the `./alephe/aleph_start` file). If the connected user is authorized for opening new passwords or authorizations for a particular library, the list will include only the ADM library to which the user is assigned.

ILL Unit
This drop-down list of allowed ILL units depends on the type of user that is being defined. If the user belongs to a specific administrative unit, then the ILL units of that specific administrative unit will be displayed. If the staff user is an ADMIN type then all ILL units will be displayed.

4.1 Tab100 Variables

The following tab100 variables define password management policies:

ALPHA-NUMERIC-PASSWORD

Defines whether or not the staff password must contain a combination of alphabetic and numeric characters. The possible values are:

Y = Password must have both alphabetic and numeric characters.

N = Password does not have to have both alphabetic and numeric characters.

This variable can only be defined in the alephe/tab tab100 variable.

The default value is N.

MINIMUM-PASSWORD-LENGTH

Defines the minimum length of a staff user's password. The maximum length is 10.

This variable can only be defined in the alephe/tab tab100 variable.

The default value is 0, meaning that there is not length restriction.

PASSWORD-CHANGE-PERIOD

Defines the number of days that can elapse before changing the password is required

This variable can only be defined in the alephe/tab tab100 variable.

The default value is 0, meaning that there is no password change enforcing policy.

PASSWORD-FAIL-BLOCK

Defines whether or nor three failed login attempts will block the user.

Y = Three failed login attempts will block the user.

N = Three failed login attempts will not block the user.

The default value is N.

PASSWORD-NO-USE-PERIOD

Defines whether or not a staff user should be blocked if the login has not been used for a defined number of days.

The default value is 0, meaning that that there is no non-use block policy.

PW-EXPIRY-WARNING

Defines how many days before expiration the user will get a warning when logging into the system.

FIRST-LOGIN-PASS-CHANGE

Defines whether or not new staff users should change the password when logging for the first time. The possible values are:

Y = password change is required when logging in for the first time.

N = password change is not required when logging in for the first time

The default value is N.

5 Column Headings (pc_tab_col.lng)

`pc_tab_col.lng` defines the columns of information that are displayed in list windows in the GUI clients.

In order to define column headings, edit the bibliographic library (USM01) table `pc_tab_col.lng`. For more information about `pc_tab_col.lng`, see the Column Headings (pc_tab_col.lng) section in the General chapter - Using and Customizing Your GUI chapter.

The following is a list of the Staff Privileges windows which use `pc_tab_col.lng` for formatting data and their identifiers (Column 1 in `pc_tab_col.lng`):

Unique Code	**Staff Privileges GUI Windows**
ADMIN_PASSWORD_LIST	Staff Privileges
ADMIN_L_PERMIT_LIST	Modify <User Name>s Access Rights
ADMIN_LIB_LIST	Access Rights - Select Library
ADMIN_MODIFY	Modify Permitted Sublibraries (left screen)
ADMIN_SUBLIB	Modify Permitted Sublibraries (right screen)
PC_TABLE_ONE_COL	Choose Administrative Library

Unique Code	Staff Privileges GUI Windows
BUDGET_PER_LIST	Budget Permissions for <USER> in <library> Summary of <USER>'s Access Rights
RECORD_SUBLIB	Sublibrary/Order-Unit Management

6 ALEPH Staff Menu Permissions

The list of actions that are displayed in the Staff Privileges interface and their grouping is not configurable. However, the texts are derived from the alephe/tab/user_function.lng table. Columns 3 and 6 of this table may be configured with the relevant language sensitive texts.

7 Access Rights to Non Standard Services

A special group of 30 privileges is assigned in the system for non standard batch jobs, i.e. offline batch jobs that have not been supplied by Ex Libris, and have been written by ALEPH users for their own special needs. The privileges are grouped in the $alephe_tab/user_function.lng Private Services group (PRIV-S), as follows:

PRIV-S L Private Services GLOBAL L All Unspecified Subfunctions
PRIV-S L Private Services PRIV-01 L PRIV-01 Private Service 01
PRIV-S L Private Services PRIV-02 L PRIV-02 Private Service 02
PRIV-S L Private Services PRIV-03 L PRIV-03 Private Service 03
PRIV-S L Private Services PRIV-04 L PRIV-04 Private Service 04
PRIV-S L Private Services PRIV-05 L PRIV-05 Private Service 05
PRIV-S L Private Services PRIV-06 L PRIV-06 Private Service 06
…

In order to control the private jobs' privileges, every private batch job must be mapped to one of these privileges. This is done by implementing the following two actions:

1. Map the batch job to a PRIV-xx privilege, where xx is a number between 01 and 30. This is done in the service's xml form that is found in the alephe/pc_b_lng directory. For example, if you have a private batch job that is named p-exam-01, then the batch job's xml form (alephe/pc_b_lng/p-exam-01.xml) must contain a line such as:

<permission>p_priv_01</permission>

For example:
<?xml version = "1.0"?>
<!DOCTYPE dialog SYSTEM "dialog.dtd">
<dialog>
<title>Example Service (exam-01)</title>
<proc>p_exam_01</proc>
<control>
</control>
<ignore_print_id>
</ignore_print_id>
<ignore_form_language>
</ignore_form_language>
<permission>p_priv_01</permission>
</dialog>

This would make the system check the Z67 record with the PRIV-S function and PRIV-01 subfunction.

2. In the $alephe_tab/user_function.lng file, change the description of the PRIV-xx privilege to describe your private batch job. For example, change:

PRIV-S L Private Services PRIV-01 L PRIV-01 Private Service 01
To be
PRIV-S L Private Services PRIV-01 L EXAM-01 Example Service 01

This will enable managing the Z67 record with the PRIV-S function and PRIV-01 subfunction from the interface. After you implemented the previous step, this is the record that sets the privilege for the p-exam-01 batch job.

System Librarian's Guide – Task Manager

Version 23

The bridge to knowledge

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: December 14, 2017

Web address: http://www.exlibrisgroup.com

Table of Contents

1 LIBRARIES 4

2 START/STOP BATCH QUEUE 4

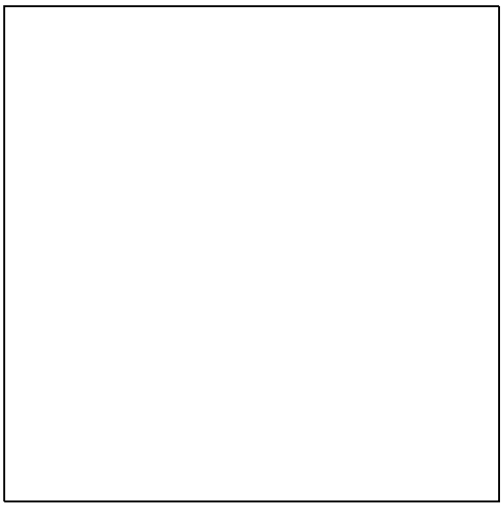
3 COLUMN HEADINGS 4

4 SENDING ALERTS 6

1 Libraries

To define the libraries to which the user can connect in the Task Manager, add the library to the LIBRARY.INI file in the ALEPHCOM/TAB directory of the client. This can be done in either of two ways:

- Open the ALEPHADM client. From the Configuration menu, click **Libraries Manager** (Ctrl+M) and add the library to the Libraries tab:



or

- Go to the `ALEPHCOM/TAB` directory of the client and open the `LIBRARY.INI` file located there. Make sure that the desired libraries are defined in this file. Following is a section from the file:

```
!-------------1---------------+--2--+-------------3---------------->
USM01 - USMARC Bibliographic   USM01 ram34.exlibris.co.il:6545
USM10 - USMARC Authority       USM10 ram34.exlibris.co.il:6545
USM30 - USMARC Course Reading  USM30 ram34.exlibris.co.il:6545
USM50 - USMARC Administrative  USM50 ram34.exlibris.co.il:6545
```

Column 1 is the text as it will appear in a menu for the user to choose from.
Column 2 is the code of the library as defined in the `tab_base` table.
Column 3 is the IP address and port number.

2 Start/Stop Batch Queue

To start or stop the Batch Queue, use UTIL C on the server.

3 Column Headings

You can define the column headings for tables of information. Such tables of information are displayed in the File List, Batch Log and Batch Queue windows and the Print ID Setup dialog box.

In order to define column headings, go to the client's `ALEPHCOM/TAB/ENG` directory (for English headings. Other headings are located in the appropriate

ALEPHCOM/TAB/<LANGUAGE> directory.) Open the TAB_COL.DAT file. Following is a sample of the relevant lines in that file:

```
1                    2 3              4  5   6  7   8  9  10

JOBMGR_LCL_FILE_LIST L Local Name     01 030 01 C01        File Name
JOBMGR_LCL_FILE_LIST L Local Date     02 025 02 C02        File Date
JOBMGR_LCL_FILE_LIST L Local Time     03 025 03 C03        File Time
JOBMGR_LCL_FILE_LIST L Local Size     04 020 04 C04        File Size
```

Column 1 is the unique code by which the system identifies this set of column headings. Following are the codes for the other sets of column headings in the Task Manager:

Code in Column 1	Table & Screen
JOBMGR_LCL_FILE_LIST	Local Files in File List
JOBMGR_RMT_FILE_LIST	Remote Files in File List
JOBMGR_ERR_FILE_LIST	Errors (displayed in the "Remote Files" area)
JOBMGR_BATCH_LOG	Upper portion of Batch Log
JOBMGR_BATCH_DETAIL	Lower portion of Batch Log
JOBMGR_ERR_BATCH_LOG	Errors (displayed in upper portion of Batch Log)
JOBMGR_BATCH_QUE	Batch Queue
JOBMGR_ERR_BATCH_QUE	Errors (displayed in the Batch Queue area)
JOBMGR_SETUP_TARGETS	Upper portion of Print ID Setup screen
JOBMGR_SETUP_LIBS	Lower portion of Print ID Setup screen

Column 2 is always L.

Column 3 is the text of the column heading that is displayed.

Column 4 is the column number (from left to right).

Column 5 is the percentage of the column's page width. The number must be 3 digits. For numbers less than 100, enter leading zeros, for example, *025*. You may remove a column heading from the table by entering *000*.

Column 6 is the font as defined in the FONT.INI file on the PC.

Column 7 is the color as defined in the ALEPHCOM.INI file on the PC.

Columns 8 and 9 are not relevant to the Task Manager.

Column 10 is a note that is not displayed on-screen.

4 Sending Alerts

The system may be configured to send email alerts when a batch jobs finishes execution. This must be done by configuring the tab_alert table, found in each library's data_tab directory. Below is a description of the table:

Column 1 – The method in which the alert is sent. Currently only emails (E) are supported.
Column 2 – The information of the recipient of the alert. Currently only email addresses may be specified.
Column 3 – The type of task for which the alert should be sent to the specified recipient. It currently may be used to specify the job type. Enter **ALL** to specify all batch jobs.

Possible values are
CIRC,ACQ,CATALOG,ILL,INDEX,UNION,ITEMS,ADAM,STAFF,SYSTEM.

In order to get an alert for a specific job, specify the job ID (e.g. CIR-07, ACQ-10, etc.)

3- Distribution: the distributed example of the table is under xxx01/tab. Again it lacks examples of alerts for specific jobs (as opposed to ALL jobs).

```
!1                          2                              3           4
!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!-!!!!!!!!!!!!!!
E yoel.kortick@exlibrisgroup.com                          ALL     859_8_TO_UTF
E yoel2091@yahoo.com                                      ALL     8859_8_TO_UTF
!* E yoel.kortick@exlibris.co.il                          ALL
```

Column 4 – The encoding which is used when sending out the email.

System Librarian's Guide – Web OPAC

Version 23

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

"Ex Libris," the Ex Libris bridge , Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2017. All rights reserved.

Document released: August 7, 2017

Web address: http://www.exlibrisgroup.com

Table of Contents

1 INTRODUCTION 8

2 SETTING UP THE SYSTEM 8

2.1 aleph_start Parameters 21

3 PATRON REGISTRATION AND PROFILES 22

3.1 Patron Registration 22

3.2 Patron Profiles 24

3.3 Interface Language 25

3.4 Brief View of Records 26

3.5 Full View of Records 27

3.6 Additional Optional Fields 27

4 HTML FILES AND THE INTERFACE 27

4.1 ALEPH and CSS 28

4.2 Include Files 29

4.3 File Extensions 29

4.4 Menu Bar 30

4.5 Encryption of patron connection information using SSL 30

4.6 Setting up the Browse and Search Screens 31

4.7 Result Screens 36
4.7.1 Brief View (Table Format) 37
4.7.2 Other Brief Format Options 41

4.8 Full View 42

4.9 Holdings Screens 48

4.10 User Screens 49

4.11 Submitting a Photocopy Form 50
4.11.1 General 50
4.11.2 Photocopy Submission for Article Document 50

4.12 ILL Screens 51

4.13 Setting up the Special Request Function....53
4.14 Support For Right-To-Left Text Display In All Formats....56
4.15 Adding an SFX Link Button....57
4.16 Setting Up the Refine Window....60
4.17 Photocopy Information....61

5 SORT 61

6 FILTERS 62

6.1 Setting Up Bases and Filters....65

7 ERROR MESSAGES 69

8 DEFAULT TEXT FOR LINKS 70

9 RANK 71

10 ADDING A NEW INDEX 71

11 SETTING UP THE SDI FUNCTION 74

12 LOGICAL BASES 75

12.1 Logical Base Counter....76
12.2 Denied Libraries....77

13 PATRON ADM LIBRARY (SINGLE/MULTIPLE) 78

14 SYNONYMS DICTIONARY FOR WORD SEARCHING (OPTIONAL) 78

15 SETTING UP SAVE/MAIL ENCODING OPTIONS 79

16 SAVE/MAIL ON MULTI-APACHE TOPOLOGY 80

17 LOCAL OWN CHECK PROCEDURE 81

18 ACQ REQUEST FOR LIBRARY WITH MORE THAN ONE BIB LIBRARY 82

19 APPLYING ADM FILTER IN OPAC-ACQ REQUESTS 83

19.1 Activating the ADM Library Filter 83

19.2 ADM Library Filter Routine 84

20 LOG FILE EXPANSION 84

21 SETTING UP BIB REQUEST POLICY 85

21.1 Group Definition and Availability 85

21.2 Pickup Location List 85

21.3 Item Level Request 86

22 SETTING UP MY E-SHELF 86

22.1 My e-Shelf and Course Reading Addresses 87

22.2 Setting the Maximum Number of Records in a Folder 88

22.3 Setting Up the Note Column in the Records Which Were Not Requested Table 89

22.4 Setting Up the Submitted a Hold Request Column in My e-Shelf 89

23 SEARCH KEYWORD IN HEADINGS 90

23.1 Keyword Search in Heading List Functionality Via Web OPAC-Browse 90

23.2 Search Keyword in Heading via Web OPAC-Search/Browse 91

23.3 Search Keyword in Heading via GUI-Search-Browse 92

23.4 Search Rules and Logical Operators for Keyword Search in Headings 92

23.4.1 Truncation 93

23.4.2 Logical (Boolean) Operators 93

23.4.3 Parentheses 93

23.4.4 Phrase Searches 94

23.4.5 Using the "-" Symbol 94

23.5 Setting Up Keyword Search in Heading List Functionality 94

24 WEB OPAC AND OPENSEARCH™ COMPATIBILITY 97

25 WEB OPAC SECURITY 98

25.1 XSS Validation 98

25.2 CSRF Valuation 98
25.3 Session Fixation 99
25.4 Apache Security Vulnerability Patch 100

26 WEB OPAC ACCESSIBILITY 100

26.1 Adding Descriptions to Images 100
26.2 Defining Titles with the Heading Attribute 101
26.3 Bulleted and Numbered Lists 101
26.4 Indicating Table Headers 102
26.5 Assigning Labels to Input Form Elements 103
26.5.1 Text Fields 103
26.5.2 Text Area Fields 104
26.5.3 Checkbox Fields 104
26.5.4 Radio Button Fields 105
26.5.5 Drop-Down List Fields 105
26.6 Color Indications 107
26.7 Additional Changes for Aleph Web Files 107
26.8 Skipping Repetitive Links 109
26.9 Exceptions 110

27 LINK TO GOOGLE BOOKS SERVICE 110

28 FORGOT PASSWORD 111

28.1 Reset Password by Answering a Predefined Question 111
28.2 Reset Password by Getting an Email 112

29 WEB OPAC-RELATED TABLES 114

30 CANCELLING HOLD REQUESTS 122

31 STAFF REQUEST AND REQUESTER NAME FIELD 123

31.1 Requester Name Field on Request Creation Forms 123
31.2 Requester Name Information in My Library Card 125
31.3 Sort Based on Requester Name Information 126

32 SERVER SIDE SCRIPTS IN THE ALEPH OPAC 126

32.1 Script Input 127

32.2 Script Output 127

33 LIST OF WEB OPAC FILES 127

33.1 List of Web OPAC Files 130

1 Introduction

You can run the Web OPAC using the default settings in your ALEPH 500 installation. However, most libraries choose to customize the Web OPAC in line with their policies and needs.

As Systems Librarian, you can modify HTML screens, display formats, messages, access points and command codes. A working knowledge of HTML is necessary in order to change your Web OPAC's screen settings. No additional programming skills are required. Any changes you make are updated in the system immediately.

2 Setting Up the System

You can customize the display of the OPAC interface by modifying the parameters listed in the table below. These parameters are defined in the default parameter section of the `www_server.conf` table in the `alephe` directory. Edit the `www_server.conf` table. You can disable a setting by placing a # at the start of the relevant line.

The following table lists the default parameters together with a brief explanation of their associated functionality.

Default Parameter	**Set this parameter to ..**
`allow_create_new_profile`	Allow the creation of new personal profiles
`create_statistics`	Enable statistical analysis of Web server use
`direction_support`	Set up right-to-left text display
`due_date_format`	Set up the due date format of loans
func_in_log	Set up the name of the function (program) to the ending of each transaction in the Web server log file.
`hol_filter_op`	
`hol_sort_op`	Determine the filter and sort options of the Holding records list
`ill_total_limit_period`	Define period limits for ILL requests
`Keep_z39_gate`	Prevent the automatic start of the z39_gate
`Mark_match_words_prefix`	
`Mark_match_words_suffix`	Format highlighted search terms

Default Parameter	Set this parameter to ..
patron_notice_cash	Check the patron's cash transaction after successfully logging into the Web OPAC and display the 'Pending Cash Transaction' notice if unpaid cash transactions are detected.
patron_notice_delinq	Check the patron's delinquencies after successfully logging into the Web OPAC and display the 'Delinquency Notice' if delinquencies are detected.
patron_notice_overdue	Check the patron's loaned items after successfully logging into the Web OPAC and display the 'Overdue Items' notice if overdue items are detected.
patron_expiry_alert	Check if any of the patron's local accounts (z305) expire in N days. The number of days is defined in patron_expiry_alert variable (Possible values: numbers between 1- 999). If one (or more) of the accounts expire during this period, a message is displayed in the patron notice page, alerting the patron about the coming expiration and the 'Library Card' includes a link for extending the user's account.
server_default_charset	Set up a default character set for non-UTF-8-enabled Web browser
set_default_z110_create_order	Control the order in which a result set is populated
set_prox_limit	Limit the number of documents that are checked for proximity
set_result_set_limit	This parameter defines the find command limit
set_word_limit	Define the maximum number of words retrieved when truncation is used
sfx_base_url	Define the IP of your SFX server
sfx_fetch_address	Set up your SFX server to fetch additional data from an ALEPH server
sfx_issn_routine	The format from which the ISSN component of the SFX link will be constructed
Show_copyrights	Control display of copyright information

Default Parameter	Set this parameter to ..
www_a_s_time_out	Define the length of user session before timeout
www_bor_registration_ends	Define the valid period of the patron's address
www_browse_previous	Set up the Browse list display
www_con_lng	Set the default language of the OPAC interface
www_course_email_address	Set up the destination e-mail address for course reading material
www_dn_link_display	Set up down link display
www_dn_view_display	Set this parameter to View down links display
www_f_sfx	
www_f_sfx_dim	Set up display of SFX button
www_feedback_email_address	Define the e-mail address to which the feedback will be sent from the Web OPAC
www_item_schedule	Determine the number of days items can be booked in advance
www_lib_info_link	Enable user to see more information about a branch library
www_myeshelf_limit	Set the maximum number of records allowed in a My e-Shelf folder (including BASKET)
www_login	Set the screen to appear after log-in
www_no_items_display	Determine the number of items displayed per page
www_parallel_search_base	Define the databases searched when Multi-Base search is selected in the Web OPAC
www_short_max_no_ext_links	Limit the number of electronic locations in the Brief display
www_short_max_no_locations	Determine the maximum number of location links in the Brief list
www_sort_field	Define the default sort order for Browse results
www_sort_field_aut	

Default Parameter	Set this parameter to ..
www_sort_field_sub	Define the default sort order for Browse results
www_sort_limit	Limits the sorting of result sets
www_update_session_id	Define the default OPAC user profile
www_user_z37_sort_routine	
www_user_z37_sort_order	
www_user_z38_sort_routine	
www_user_z38_sort_order	Configure the hold/photo/loan sort order
www_z36h_max_record_limit	Limit the number of loans displayed in the Loan History list
www_z37h_max_record_limit	Limit the number of records displayed in the Hold History list
www_z37_course_request_days	Define the number of days before a course that a hold request becomes active
www_z37_date_to	Define the end of the time slot when a hold request is created
www_z37_recall_type	Set up the recall status for patron hold requests
ichain_mode	Use the ichain option

```
setenv www_no_items_display          100
```
This parameter determines the number of items displayed in one page of the Holdings Window.

If no value is defined for this parameter, the default value is 40 items displayed per page.

```
setenv www_server_timeout            300
```
This parameter defines the maximum number of seconds for the www_server timeout. It must be in the range of 5-300. Recommended values are in the range of 5-120.

```
setenv www_myeshelf_limit            100
```
This parameter is used to determine the maximum number of records that can be stored in each folder of the My e-Shelf function.

If this parameter is not defined, the default is 500 records. If the defined number is higher than 500 (for example, 1000), the system will consider it to be the default value of 500, which is the maximum that can be defined in this parameter.

```
setenv www_sort_field          "01---D02---A"
```

This setting defines the default sort order for the results of a Search query (the OPAC user can change the sort order of a set when the set is displayed in Brief Table Format). The numbers 01 and 02 are codes taken from column 1 of the tab_sort table. In this case, 01 refers to the Year field and 02 refers to the Author field. To sort the records by a different field, make sure the field is defined in tab_sort.

The letter "D" after 01 in the Year field stands for "descending" order which means that the latest dates will be listed first (for example, 1999 will be listed before 1998). The letter "A" stands for "ascending" order and means that the order of the Authors will be from A to Z.

You can, of course, set the order for any field to be ascending or descending.

```
setenv www_sort_field_aut      "03---A01---D"
setenv www_sort_field_sub      "01---D02---A"
```
These settings define the default sort order for the results of a Browse query (the OPAC user cannot change this sort order).

The "aut" and "sub" in the above parameters refer to the Browse Indexes as defined in tab00.lng, column 2. In this case, "aut" refers to Author and "sub" refers to Subject. You can define other parameters, but you must use the code of an index that is defined in tab00.lng.

The character strings are defined in the same way as in the Search query (setenv www_sort_field).

```
setenv www_browse_previous  Y
```
You can set the Browse list of the headings index to display in two ways:

- The entry that matches the user's query heads the list.
- The entry immediately preceding the user's query heads the list.

The starting point of the list is determined by the above setting, whereby Y (default) displays the list headed by the preceding entry, and N displays the list headed by the entry.

We recommend that the Browse list be set to display the immediately preceding entry, as it helps the user understand where he is if the entry that he requested does not appear in the list.

```
setenv www_item_schedule        14
```
This setting determines the number of days of the Reserve Item Schedule that are accessible in the OPAC for advance booking. In the above example, a user is able to book an item up to 14 days in advance.

```
setenv www_dn_link_display          01
```
Down links in a record can be displayed in one of two ways:

- Separate Entry For Each Down Link
- Referral To Links Window

The value of the above setting determines which of the two options is used. This value is defined by the system librarian. If the number of records found is less than, or equal to, the value defined, there will be a separate entry for each down link in the Full View of the Bibliographic Record. Otherwise, the down links are displayed as a referral to other windows that contain the down links. These down links are often used with analytic records.

The text that displays when a record has more than the number of links defined in www_dn_link_display is set in the first line in `tab05.lng`, for example:

```
NONE       Click to view related record
```

```
setenv www_f_sfx          "&#60;img src=&icon_path/f-sfx.gif border=0
alt='Use SFX services'&#62;"
setenv www_f_sfx_dim      "&#60;img src=&icon_path/f-sfx-dim.gif
border=0 alt='No SFX services are available'&#62;"
```

The *www_f_sfx* and *www_f_sfx_dim* variables define whether or not an SFX button will be displayed in the Web OPAC and if it will be dimmed or not, depending on the availability of SFX services.

In the Full display (direct-head, full-set-head), the SFX button can be made sensitive to different SFX services. The placeholder for the SFX icon is $1100. The services that are checked are defined in tab_sfx. For example, if you want the SFX button to be available only if full text is available, the tab_sfx setup is:

```
USM01                www_f_sfx_ft         12 N getFullTxt
```

Note that although the SFX button can be displayed on the list of items (placeholder $1500 on item-global-body) and on the Brief list (www_tab_short), this SFX button cannot be made sensitive.

```
setenv www_dn_view_display Y
```

This parameter, if set to Y, allows "View down links" to display in the full view of a record in Web OPAC. This creates a list based on the z103-lkr-text-n sorted by z103-sort. It also displays a link “Create set of down-linked records". This link creates a new set in the results list containing the linked records.

The display works in accordance with the `www_dn_link_display` variable.

```
setenv set_word_limit           1000
```

set_word_limit defines the maximum number of words that are retrieved when truncation is used (for example, find a? runs a FIND query on all words beginning with *a*). If a find limit exists, this is the number of distinct words included in a given search. The maximum value allowed is 9999.

```
setenv mark_match_words_prefix      "<span class=text2 id=normalb>
setenv mark_match_words_suffix      "</span>
```

In the Full View of record display (controlled by edit_doc_999.lng), the system highlights the search terms that have been used to find the record. This includes highlighting words from a FIND query and the entry from a browse list. The format of the highlighting is defined using HTML prefix and suffix coding. The above example uses the exlibris.css cascading style sheet to set bold highlighting.

The system default is:
```
mark_match_words_prefix "<b>"
mark_match_words_suffix "<b>"
```
The highlighting can be turned off by defining a non-HTML prefix and suffix. For example:
```
mark_match_words_prefix "<x>"
mark_match_words_suffix "<x>"
```
The highlighting of particular words can be turned off by listing them in tab06.

```
setenv www_user_z37_sort_routine      00
setenv www_user_z37_sort_order        A
setenv www_user_z38_sort_routine      00
setenv www_user_z38_sort_order        A
```

The above display parameters allow you to configure the hold/photo/loan sort order in the Web OPAC:

The routines available are:

z37 (Holds):
00 = z37-priority z37-request-date z37-open-date z37-open-hour

01 = z30-sublibrary z30-item-status z30-collection z37-open-date z37-open-hour

02 = z30-sublibrary z30-item-status z30-collection z37-status z37-open-date z37-open-hour

04 = sort using group ID and sequence, priority, request interest date and request date

05= sort using open date and requester name

z38 (Photocopy):

00 = z38-open-date
08 =z38-open-date and z38-requester-name

The sort orders available are:
A = Ascending Order
D = Descending Order

```
setenv www_short_max_no_ext_links "02"
```
This parameter limits the number of electronic locations that can appear in the Brief Display format in the Electronic Location column.

```
setenv www_short_max_no_locations "03"
```
This parameter determines the maximum number of location links in the Web Short Display (Brief List).

If this limit is exceeded, a message displays. The message is set in lines 9553 and 9564 of `/alephe/error.lng/www_f_heading`

The location information is limited to 5000 characters.

```
setenv sfx_base_url    http://demo.exlibrisgroup.com:9003/demo
```

Defines the IP address of the SFX server.

The sfx-base-url variable is ADM library-sensitive. This means that different URLs can be defined for different ADM libraries, in the following manner:

```
setenv sfx_base_url       http://demo.exlibrisgroup.com:9003/demo
setenv sfx_base_url_usm50 http://demo.exlibrisgroup.com:9004/demo
setenv sfx_base_url_usm51 http://ram47:8888/exlibris
```

When a patron logs into the Web OPAC, his session identifiers include which base URL will be used for the SFX link. The base URL will be set in the following manner

- If there is a Web cookie named "BASE-URL", the value of that cookie should contain the SFX host name and instance, in the format *http://host-name:port/instance*. In that case, the value of the cookie will set the base URL, and the `www_server.conf` variable is irrelevant.

- If there is no Web cookie, the ADM-specific `www_server.conf` variable will be used, according to the patron's profile ADM library. For example, if the patron has the ADM library USM50 set as the ADM library in his profile (or his group profile or ALEPH profile), the `www_server.conf` variable *sfx_base_url_usm50* will be used to set the patron's base URL.

- If there is no matching ADM-specific `www_server.conf` variable, the *sfx_base_url* variable will set the base URL.

`setenv sfx_fetch_address ,Ip=whitman.exlibris.com,Port=9909`

The Ex Libris SFX server can retrieve additional data from an ALEPH server. For this purpose, the OpenURL contains a link to the full document record in the ALEPH database, in the "pid" parameter. The setup for the pid is defined in the display parameter above. You must define ",Ip=" followed by your ALEPH server address and your Z39.50 port.

`sfx_issn_routine`

This variable defines the field from which the ISSN component of the SFX link will be constructed. This parameter has the following configuration options:

00 – Extract the last 022 $$a

01 – Extract the last 022 $$y if the default routine does not find any ISSN.

`setenv show_copyrights Y`

This parameter determines whether or not copyright information is displayed when a patron uses an external link via the Web OPAC. If it is set to Y, copyright information is shown to the patron if the 856## subfield $$4 is not cataloged.

If `show_copyrights` is set to N the system does not show copyright information when a patron uses an external link via the Web OPAC.

`setenv www_con_lng ENG`

This sets the default language of the Web OPAC interface.

`setenv www_course_email_address`

This determines the e-mail address to which course reading material will be sent (from the Web OPAC) for adding to the Course Reading module. If the library decides to use an alternative mail, it must be defined in `/alephe/www_f_eng/myshelf-course`. In this case, the system ignores the address in `www_server.conf`.

If you have a multi-ADM environment, it might be useful to define an alternative e-mail address of a Course Reading administrator.

`setenv create_statistics Y`

This setting defines whether or not records are to be written in the Z34 (IP Statistics) table for statistical analysis of Web server use. If it is set to Y, the "last-z34-sequence" counter must be defined in the Z52 table (UTIL G/2) of the OPAC library.

`setenv server_default_charset iso-8859-1`

If the Web server detects that the Web OPAC browser is not UTF-8 enabled (that is, the browser is not Netscape 6+ or Microsoft Internet Explorer 5+), the system overrides the UTF-8 parameter in the HTML files, 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 www_z37_recall_type 03`

A Web OPAC user can request an item by placing a hold request. The user's request automatically receives the status set in the above parameter. The patron cannot change this status.

These are the available statuses:

01 = Regular
02 = Rush
03 = No Recall

If you choose 02 (Rush), all hold requests made in the Web OPAC automatically receive the highest level priority of "00", the lowest being "99".

`setenv www_z37_course_request_days 20`

This parameter defines the number of days prior to the beginning of a course that a hold request, on an item that needs to be placed in the Course Reserves module, becomes active.

`setenv www_z36h_max_record_limit 30`

From the User Information screen in the Web OPAC, users can view the history of their loans sorted by due dates in descending order, that is, a due date of 15th July, 2004 precedes a due date of 20th June, 2004. This line allows you to set a limit to the number of previous loans that displays in this list. If the line does not appear in the table or if it is commented out, the default number of loans is 80.

```
setenv www_z37h_max_record_limit          30
```

From the User Information screen in the Web OPAC, users can view the history of their hold request sorted by History Date in descending order, so that a request that moved to history on August 30, 2011 precedes a request that moved to history on August 20, 2011. This line allows you to set a limit to the number of hold requests that are displayed in this list. If this line does not appear in the table or if it is commented out, the default number of holds is 80.

```
setenv www_update_session_id          "UPDATE_ALEPH"
```

This parameter defines the default user profile for OPAC users.

```
setenv set_default_z110_create_order          "D"
```

The display of a result set is limited to a maximum of 1000 records. When the result list is over the maximum number, the *set_default_z110_create_order* environment variable comes into play. If it is set to *"D"*, the displayed set includes the 1000 highest record numbers, and if it is set to *"A"*, the displayed set includes the lowest 1000 record numbers.

```
setenv hol_filter_op            BASE
setenv hol_sort_op              BASE
```

These parameters determine the filtering and sorting options of the Holding records list on the Holding Information display. HOL records are filtered according to the search base and the parameters given in `tab_base.lng.conf`. The rest of the HOL records are sorted again, according to the list in `tab_base.lng.conf`.

If the variables are not present in www_server.conf or if the value of hol_filter_op and hol_sort_op in tab_base.lng.conf are "NO-FILTER" and "NO-SORT" respectively, the HOL records are not filtered or sorted.

```
setenv allow_create_new_profile             Y
```

If set to Y, ALEPH creates a personal profile for the user (if a personal profile does not exist) when the user requests that his preferences are saved.

```
setenv set_prox_limit      100
```

This parameter places a limit on the number of documents that will be checked for proximity. The default is 500.

If the value of the set_prox_limit parameter is less then the number of results retrieved before the proximity filter is selected, then none of the results are checked for proximity. For example, the set_prox_limit value is 100, and a user searches for *art !4 history*. If he retrieves more than 100 records, none of the results will be checked for proximity.

If the ADJACENCY_TYPE parameter in aleph_start is set to 0, (that is, word pairs are not built), when a Search query includes Adjacency, the query is parsed as %0, and the set_prox_limit is active.

If the ADJACENCY_TYPE parameter in aleph_start is set to 2, then set_prox_limit is not relevant for a Search query including Adjacency.

```
setenv direction_support          Y
```

Supports right-to-left, right-aligned and right-oriented (for Hebrew and Arabic) text display in all formats in Web OPAC files.

```
setenv www_login                    "find-b"
```

Sets the default page which appears after a user has signed in. Valid values for this variable are find-a, find-b, find-c, find-m and find-d.

```
setenv www_z37_date_to              M003
```

This variable defines the end of the time slot (Dates of Interest) when a hold request is created. The definition must include one character that describes the time period (D-day, M-month, Y-year), followed by a three-digit value for the number of time units.

For example, M003 creates a hold request that starts from the current date and ends in another three months.

If the variable is not defined properly, the default time slot is three months from the current date.

```
setenv www_lib_info_link            Y
```

If set to Y, a user can click on an "About library" hypertext link in order to see more information about a branch library.

Note: this setting does not influence ITM4 (links to remote library items, with no circulation information, set in the edit_doc_999.lng table of the library's tab directory).

```
setenv www_parallel_search_base USM01-UNI01
```

This parameter defines the databases searched by a parallel find command (find-m-acc). The find command is activated from the ALEPH services menu "Find M documents in the database". In this example, the parallel find command will search USM01 and UNI01.

```
setenv due_date_format              2
```

The due date format of loans which appear in the Holdings window are displayed according to the following formats:

- 1 - Display a single effective due date, based on the RECALL-METHOD.
- 2 - Display the recall-due-date, if there is one (even if it is later); otherwise, display the regular due date.
- 3 - Always display the z36_due_date; (for recall methods 2 and 3 this means that the recall-due-date is never displayed).
- 4 - Display both: the regular due date and (if there is one) the recall-due-date.

`setenv www_bor_registration_ends 20050101`

Defines the default date that is entered in the *Address valid to* field in the New Borrower Registration form if no date has otherwise been entered. This date represents the end of the borrower's address validation.

`setenv ill_total_limit_period 1001`

This variable defines the date from which the counting of an ILL request is started. The ill_total_limit_period format is *mmdd* (in this example, 1001 is the first of October). All GUI and Web OPAC ILL requests created between these dates are counted and compared to the patron's ILL total limit. The patron is blocked if he exceeds its limit within this period. To override this calculation, either comment out this variable or enter 0000.

`setenv www_a_s_time_out 100`

Defines the time period (in seconds) that elapses after usage is discontinued before the user is disconnected from the network.

The above definition means that a user can stay logged in for 100 seconds.

The predefined default setting in an ALEPH installation is 600 (secs.).

`www_feedback_email_address`

This parameter defines to which e-mail address the feedback will be sent from the Web OPAC. If this parameter is not defined in `www_server.conf`, define it on the feedback page (`../alephe/www_f_eng/feedback`) in the Web OPAC as follows:

```
</div>
<form name=form1
      method=post
      action="&server_f"
      onSubmit="my_replace(document.form1.text)";>
<input type=hidden name=func value=feedback>
<input type=hidden name=file_name value=find-b>
<input type="hidden" size=50 name="EMAIL" value="Please fill here the
library email address">

<div class=text3>
    Your constructive feedback will be greatly appreciated.
</div>
```

Note that defining this line does not cause it to display on the Feedback page.

`setenv set_result_set_limit 2000`

This parameter defines the find command limit. The number of hits for a find command has the default of 1000. It is now possible to change this limit to a number in the range of 1000-20000 by setting the `set_result_set_limit` variable.

To facilitate this, you can change the following line in `short-head-2` from:

```
Records $0100 -$0200 of $0300 (maximum display and sort is 1000
records)
```

To:

```
Records $0100 -$0200 of $0300 (maximum display and sort is
$4500 records)
```

```
setenv keep_z39_gate                Y
```

This parameter prevents the automatic start of `z39_gate`.

For every incoming request, the Web server checks if `z39_gate` is up and running. If the gate is down, the gate server process will be started automatically. To prevent the gate server process starting automatically, set `keep_z39_gate` to "N".

The default is "Y".

```
setenv www_sort_limit             800
```

This parameter limits the sorting of result sets in the Web OPAC.

It contains a number of up to six digits; its default value is 999999 (that is, no sort limit). If the number of hits returned by the search query exceeds the sort limit, the result set will not be sorted.

Note

The maximum number of documents a result set can contain is controlled by the environment variable "set_result_set_limit", set either in www_server.conf or in pc_server_defaults. Make sure that the sort limit is not greater than "set_result_set_limit".

`ichain_mode`

Setting this variable to Y will run the www server in ichain-mode. When in this mode, the web server session is synchronized with the HTTP "Authorization" header. In other words, if there is no "Authorization" entry in the HTTP header, or if the header contains invalid login information, the ALEPH session will be that of a guest. If the "Authorization" entry contains valid login information, then that information will be automatically used to log in the patron.

```
setenv patron_notice_cash           Y
```

When this parameter is set to Y, the system checks the patron's cash transaction after the patron logs into WWW OPAC and displays the “Pending Cash Transaction” notice if unpaid cash transactions are detected. (HTML notice page: ./alephe/www_f_eng/patron-notice).

```
setenv patron_notice_delinq         Y
```

When this parameter is set to Y, the system checks the patron's delinquencies after the patron successfully logs into WWW OPAC and displays the “Delinquency” notice if delinquencies are detected. (HTML notice page: ./alephe/www_f_eng/patron-notice).

```
setenv patron_notice_overdue        Y
```
When this parameter is set to Y, the system checks patrons loaned items and displays the “Overdue Items” notice if the patron currently is holding overdue items. (HTML notice page: ./alephe/www_f_eng/patron-notice).

```
setenv patron_expiry_alert   030
```
When a patron logs on to OPAC, the system checks if any of the patron’s local accounts (z305) expire in N days. The number of days is defined in the `patron_expiry_alert` variable (Possible values: numbers between 1- 999).
If one (or more) of the accounts expire during this period, a message is displayed in the patron notice screen, alerting the patron about the coming expiration and the Library Card displays a link for extending the user's account. The extension period is defined in tab31 (current date plus number of days defined in tab31 columns 12-14). Depending on tab18.<lng>, the user is charged for the amount specified for cash transaction 029.

Accounts that are already expired are not extended.

```
setenv header_no_server_info        Y
```
When this parameter is set to Y, the www server does not display information (“ALEPH/5.20”) in the HTTP headers.

2.1 aleph_start Parameters

To ensure the display of authority records with multiple XYZ subfields use the following variable:

```
setenv include_all_aut Y
```

If the parameter is set to "Y", all entries which are connected to authority records will be displayed in the browse list of the Web OPAC, even if there are no records connected to this access. This applies only to codes using Z01 and not to those using Z0102. Z0102 continues to use the XYZ mechanism.
If the parameter is set to "N", the entries will not display. For example, the following will not be displayed:

```
450 0 L $$aFolk$$xCollectors and collecting$$xFolk and stories
```

This variable is optional and its default is N.

The following variable effects how ALEPH sends bibliographic information to the SFX server when using the SFX button:

```
sfx_first_isbn
```

If the BIB record has more than a single ISBN, the variable can be used to set which ISBN of the BIB record will be sent to SFX. Possible values are:

Y – Only the first ISBN of the BIB record will be sent to the SFX server.

N – Only the last ISBN of the BIB record will be sent to the SFX server. This is the default value if nothing is specified.

This environment variable is relevant for USMARC only.

3 Patron Registration and Profiles

3.1 Patron Registration

The patron registration screen is determined by the bor-new HTML file. The patron has to fill in the mandatory fields on this screen in order to complete his registration. If certain non-mandatory fields are not filled, their values are taken from the following places:

- The Patron ID field value is taken from the USR00 library's last-bor-id-1 counter.
- The Patron Verification field value is taken from the USR00 library's last-bor-verify-2 counter.
- The Address Valid From field's value contains the current date
- The Address Valid To field's value is a default value set in the program. If at least one of the "Address Type", "Address valid from" and "Address valid to" fields is empty, then the system assigns default values in the following way.

  If the ADM library to which the patron's registered home library belongs has a non-empty tab_bor_address, the values are taken from it. Otherwise, they are taken from the tab100 variable DEFAULT-ADDRESS.
- The system may be set up to allow the registering patron to register also an additional ID. This can be done by adding the following to the HTML page bor-new:

```
<tr>
 <td class=td2 width=20% nowrap>Additional ID:</td>
 <td class=td2> </td>
 <td class=td2>
     <input name=N_ADDITIONAL_ID_DATA
      size=30 maxlength=20 value="$6800">
 </td>
</tr>
<tr>
 <td class=td2 width=20% nowrap>Additional ID Ver:</td>
 <td class=td2> </td>
 <td class=td2>
     <input type=password name=N_ADDITIONAL_ID_VER
```

```
      size=30 maxlength=20>
  </td>
 </tr>
 <tr>
  <td class=td2 width=20% nowrap>Add.ID Ver.Repeat:</td>
  <td class=td2> </td>
  <td class=td2>
      <input type=password name=N_ADDITIONAL_ID_VER_CHECK
       size=30 maxlength=20>
  </td>
 </tr>
```

The Additional ID field is optional, and the verification is required only if the ID has been filled in.

The type of ID that will be added is set in the usr_library's tab100 variable ADDITIONAL-ID-TYPE.

- You can set up the registration page (bor-new) to require the registering patron to fill in the home library. To do this, add the following to the bor-new page:

```
<select name=M_Z303_HOME_LIB>
   <option value=" "></option>
   <option value=LIB1 $$2400-S"Library 1"> Library 1</option>
   <option value=LIB2 $$2400-S"Library 2"> Library 2</option>
</select>
```

If you do not want the home library to be a mandatory field, either use N_Z303_HOME_LIB instead of M_Z303_HOME_LIB, or replace:

```
<option value=" "></option>
```

With:

```
<option value=EMPTY $$2400-S"EMPTY"></option>
```

In the event that the home library is not filled in, the library that is registered with patron privileges (‘Library Branch’) will be used as the patron’s home library.

If the above section is not added to the bor-new page, the patron will not be asked to fill in a home library. The library that is registered with patron privileges (‘Library Branch’) will be used as the patron’s home library.

- A local patron record (Z305) is created for the sublibrary chosen by the user. The settings for the patron's Local Privileges and expiry date are defined in the bor-new HTML file. An ALEPH local patron is automatically created, with privilege and date values taken from `tab31` in the ADM library. For more information on Patron Registration, see the Patrons chapter in the Circulation guide.

3.2 Patron Profiles

Patron profiles are set in the Circulation GUI, either by using the Patrons / Patron Profiles menu commands, or by using the Patron bar. You must define a default user profile (ALEPH) for Web OPAC users. You can also define individual or group user profiles for library patrons. An individual profile has the same ID as the user, and does not have to be entered in the profile field of the user record. A group profile must be entered in the profile field of the user record. These profiles set preferences for display, language, database and home library, as well as permissions for specific utilities and libraries. A comprehensive explanation of the default user profile setup is contained in the Patron Profiles section of the Circulation - Patrons chapter.

You can also allow the automatic setup of new profiles (see explanation of setenv allow_create_new_profile.

There are two types of Web OPAC users: identified and unidentified.

A user who has not identified himself (by signing in) is assigned a default profile. If the system librarian has set up profile(s) that match the IP address of the user workstation, the profile that most closely matches the IP address is used. If there is no matching IP address profile, the default profile is ALEPH.

When a user identifies himself, if the user has a matching personal profile (that is, the profile ID is the same as the user ID), this profile is used. If there is no matching profile, the system uses the group profile registered in the user's profile field. If the profile field is blank, the IP or ALEPH profile remains in effect.

Both unidentified and identified users can override the default user profile when they access the Web OPAC by setting their own preferences. Identified and unidentified users differ as follows:

- Identified users who have a personal profile - A user who has signed in can enter his own preferences and save them, using the Display Formats window accessed via Preferences on the menu bar. When an identified user clicks End Session on the menu bar, the display format available for the next user reverts to the default user profile.

- Identified users who have a group profile or Unidentified users - Any user can enter his own preferences, using the Display Formats window accessed via Preferences link on the menu bar. If the user has been assigned a group profile, or if the user has not signed in, the changes in preferences cannot be saved.

- Identified users who do not have a profile listed in their user record - When the user enters his preferences, if the profile field of the user's record is blank, a profile record is created (in which his preferences are saved and permissions are defaulted to the permissions of the default profile), and the profile field is updated.

User profiles are contained in the Z61 table. Apart from setting user preferences for the display and retrieval of information in the Web OPAC, the Z61 record is also used to define the list of logical bases to which the user is denied access. This table also determines authorizations such as:

- Whether or not the user has access to electronic locations (mainly field 856) from the Web OPAC.
- Permission to save sets on the server.
- Permission to create and send lists for Course Reading.
- SDI permission.
- Special Request permission.
- Address update permission.

The Z61 record also includes:

- Denied bases.
- The display format of the records.
- The number of records displayed per page and the order in which records are sorted for display.
- The default sublibrary for holdings display in OPAC.

To ensure that the default sublibrary for holdings is displayed, check that the following line in `edit_doc_999.lng` is present:

```
## ITM5                 D LHoldings (5)        Y       E
```

The User Profile can be created through the Profile option in the Circulation module. The only fields that can be changed by the user through the Web OPAC are those related to the display format. Other fields, such as the denied records (base), can be changed and updated only through the Circulation module.

Usually, the Z61 table is not converted and it is built in the system online.

3.3 Interface Language

You can set up the Web OPAC interface in numerous languages. Each language interface has its own set of HTML files (in the `www_f_lng` directory), where *lng* is replaced by the three-letter code of the language. Tables which contain formats for data display (located in the library's tab directory) must have the proper language extension as well (for example, `edit doc_999.ger`).

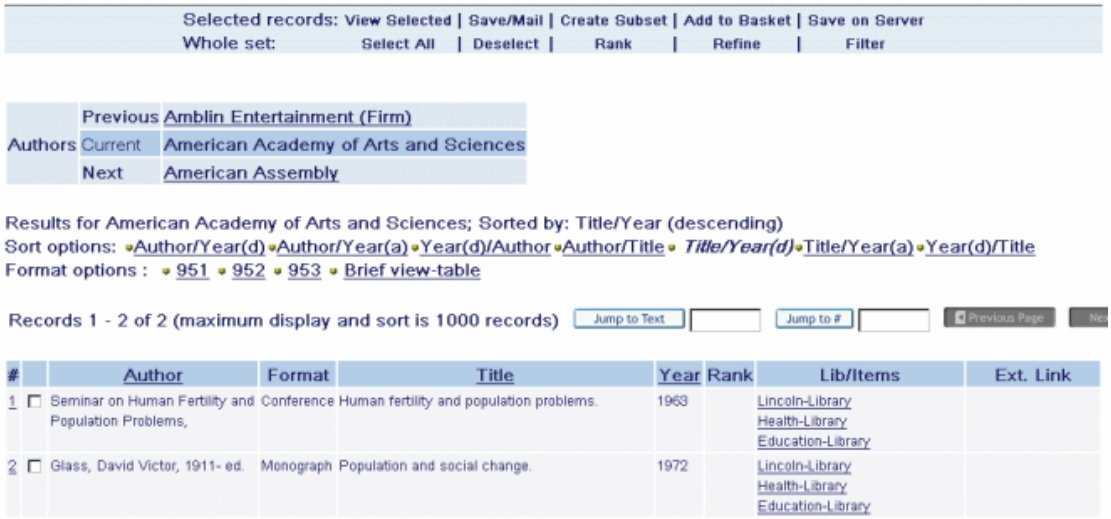
When a Web OPAC user signs in, he is assigned his interface language default as defined by the System Librarian in the Profile Information window in the Circulation GUI (See the Patron chapter in Circulation, Patron Profiles).

If a Web OPAC user does not sign in, he is assigned the default profile (including the default language) that has been defined for guest users by the System Librarian in the Profile Information window in the Circulation GUI.

Any Web OPAC user can change the interface language, but only a signed-in user can save his change of the interface language default.

3.4 Brief View of Records

The Brief View is available in a number of different formats. The table format is the most common:



You define which format is displayed by default and which fields are displayed by entering the appropriate setting in the Profile Information window in the Circulation GUI for the default profile (ALEPH). (See the Patron chapter in Circulation, Patron Profiles).

Table Format

If you want the **table format** to be displayed, enter "000" in the Brief Format field from the Profile Information window in the Circulation GUI for the default profile (ALEPH). (See the Patron chapter in Circulation, Patron Profiles).

To determine which fields are displayed in the brief view table format, edit www_tab_short.

Customizing Other Formats

You can set up other custom formats by editing the `www_f_tab_short_999` table.

If you want a custom format to be displayed by default, enter its format number in the Brief Format text box in the Profile Information window in the Circulation GUI for the default profile (ALEPH). (See the Patron chapter in Circulation, Patron Profiles).

Brief Records
To define the number of Brief records displayed on one page, enter a value in the **Records per file** text box, for example 20, from the Profile Information window in the Circulation GUI for the default profile (ALEPH). (See the Patron chapter in Circulation, Patron Profiles).

In this example, up to 20 records can be displayed on one screen (20 being the maximum number of records possible). The Web OPAC user can go to the next 20 records by clicking **Next**.

3.5 Full View of Records

You can set up the system to automatically display records in Full View, if the number of records in a result set does not exceed or equal a specified number. You set this number in the Auto Full box in the Profile Information window in the Circulation GUI for the default profile (ALEPH). (See the Patron chapter in Circulation, Patron Profiles).

For example, if you enter "05", records are displayed in Full View if a query results in less than five records.

Note that if you enter "00", all records will automatically display in Full View.

3.6 Additional Optional Fields

The following fields can also be configured in the bor-new file:

Z304_ADDRESS1
Z305_PHOTO_CHARGE
Z305_HOLD_ON_SHELF
Z305_BOR_TYPE
Z303_ILL_ACTIVE_LIMIT
Z303_ILL_TOTAL_LIMIT
Z303_ILL_LIB

4 HTML Files and the Interface

ALEPH Web OPAC screens are based on HTML (HyperText Markup Language) files. The design of these screens is based on Cascading Style Sheets (CSS). The HTML and CSS files are located in the `/alephe/www_f_` directory.

You can design Web OPAC files in accordance with your library's needs. Files must be written in accordance with the rules of HTML. This section of the ALEPH User Guide shows you which parameters can be modified and which parameters must not be changed. This section assumes that you have a working knowledge of HTML.

4.1 ALEPH and CSS

ALEPH uses Cascading Style Sheets (CSS) for adding style definitions to Web files. CSS gives more control over the look and layout of HTML files. The use of CSS controls:

- Fonts, font sizes, and font colors
- Line spacing and length
- Margins and indents
- Background colors

... and more.

ALEPH uses a single external style sheet, **exlibris.css**, located in the alephe/www_f_<lng> directory. The external style sheet is linked to HTML files through the HTML <link> tag:

```
<link rel="stylesheet"
href="&server_css/exlibris.css"TYPE="text/css">
```

The <link> tag is placed in the **meta-tags** HTML file located in the `alephe/www_f_<lng>` directory. The external style sheets must not contain any HTML tags such as <head> or <style>. The style sheet must consist of style rules or statements only. Among the benefits of using a single CSS are:

- The ability to ensure a consistent style.
- The ability to update the look of the site with a single change.
- The HTML code becomes much simpler and more manageable.

Font definitions in the `exlibris.css` file control more than just the displayed text in Web files. Font definitions also control the display for the retrieved data from the catalog. If the catalog contains special scripts such as CJK, Hebrew, Arabic, and so on, you must ensure that appropriate fonts have been defined. It is important that lists of font names have the fonts listed in the same order, since the browser picks the first one available. If the catalog does not have special scripts, you do not need to define fonts for them in the `exlibris.css` file.

Note that font definitions in the `exlibris.css` file are not used if the user's browser has been set up to ignore font styles. In MS Internet Explorer this is set up under Internet options -> General -> Accessibility -> "Ignore font styles specified on Web files". In Netscape this is set up under Edit -> Preferences -> Fonts -> "Use my default fonts overriding document specified font".

Note: Each browser supports Cascading Style Sheets in different ways and triggers different display behavior.

4.2 Include Files

Include files are files that contain parts which are repeated in many screens within Web OPAC, such as the menu bar that appears at the head of each file. Include files save repeated writing of HTML code and help keep a uniform look for the interface.

The include file is invoked in the HTML file through the `<include>` tag. This tag is not a standard HTML tag and is recognized only by ALEPH programs. The rules for use of this tag are as follows:

- The format for the code is:
  `<include>[file name]`
- The <include> tag must be located on the left side of the file. ALEPH programs ignore an `<include>` tag that is located in any other part of the file.
- The include files are located together with the other Web files in the `/alephe/www_f_<lng>` directory.

4.3 File Extensions

Web files can be base-sensitive. This allows the use of different files for different libraries when needed. For example, the pull-down menu of possible search lists (find-code-include) might differ from library to library. Let us say that USM01 and USM10 have different search codes. To use different search lists, you must add the library code as an extension to the "find-code-include" file. For example, find-code-include-usm10, find-code-include-usm30, and so on.

The base-sensitive functionality works in conjunction with the tab_base.lng columns 1, 4,5 and 6. Web programs search for Web files in the following order:

- Look for the file with the base name extension (Col. 1).
- If not found, look for the file with Col. 4 extension.
- If not found, look for the file with Col. 5 extension.
- If not found, look for the file with Col. 6 extension.
- If not found, look for the file with no extension.

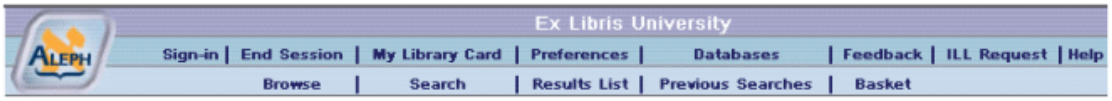
The file extension mechanism also allows you to create a different look for each base or logical base in a multi-base system. In addition, the extension code mechanism described above is also used for the following tables that govern Full record display:

- edit_doc.lng
- edit_doc_999.lng
- edit_doc_999_<aut_library>.lng.

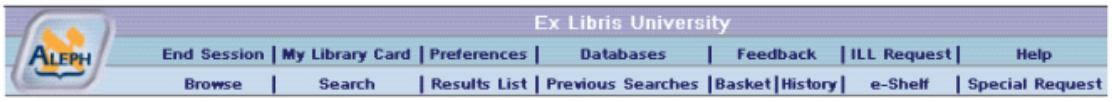
The use of the extension code with these tables lets you customize different Full record displays for different bases.

4.4 Menu Bar

The menu bar appears at the top of the Web OPAC screen. The options shown on the menu bar vary according to whether or not the user has signed in. If the user has not signed in, several options are unavailable and the menu bar looks like this:



When a user has completed sign-in, the display of the menu bar is as follows:



There are two relevant HTML files: head-1-bor and head-1-no- bor. The difference is that if a user is signed in, the head-1-bor file is displayed, and if the user has not signed in, the head-1-no-bor file is displayed.

This has been designed so that the "Sign-In" link can be changed to an "End Session" link once a user has signed in.

4.5 Encryption of patron connection information using SSL

Patron connection information is considered sensitive information. You can encrypt it using the SSL encryption protocol.

In order to send the information through HTTPS (SSL) instead of regular HTTP, carry out the following steps in all login screens:

1. In the form method definition, replace action=”&server_f” with action=”&server_f_ssl”.
2. Immediately after the form definition, add the following line:

```
<input type=”hidden” name=”ssl_flag” value=”Y”>
Here is an example:
<form method=post
      name="form1"
      action="&server_f_ssl">
      <input type="hidden" name="ssl_flag" value="Y">
```

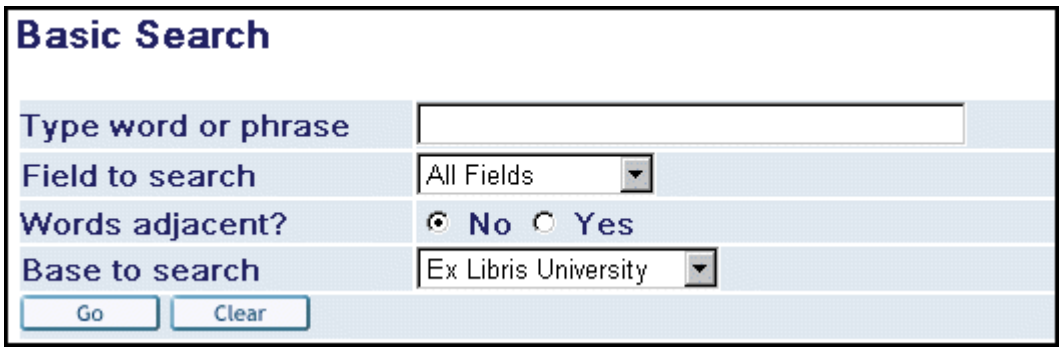
Note that in order for this setup to work, Apache and SSL must be installed, and a valid certificate placed in the system.

4.6 Setting up the Browse and Search Screens

The Browse and Search screens have similar elements as they are all made up of a combination of a number of HTML files. The HTML files are combined by using the "include" command.

Pull-Down Menus

The pull-down menus that are displayed in the Browse and Search screens are defined by a number of files. For example, the codes in the following pull-down list from the Browse screen:



are defined by the scan-include-2 HTML file, located in the `alephe/www_f_lng` directory.

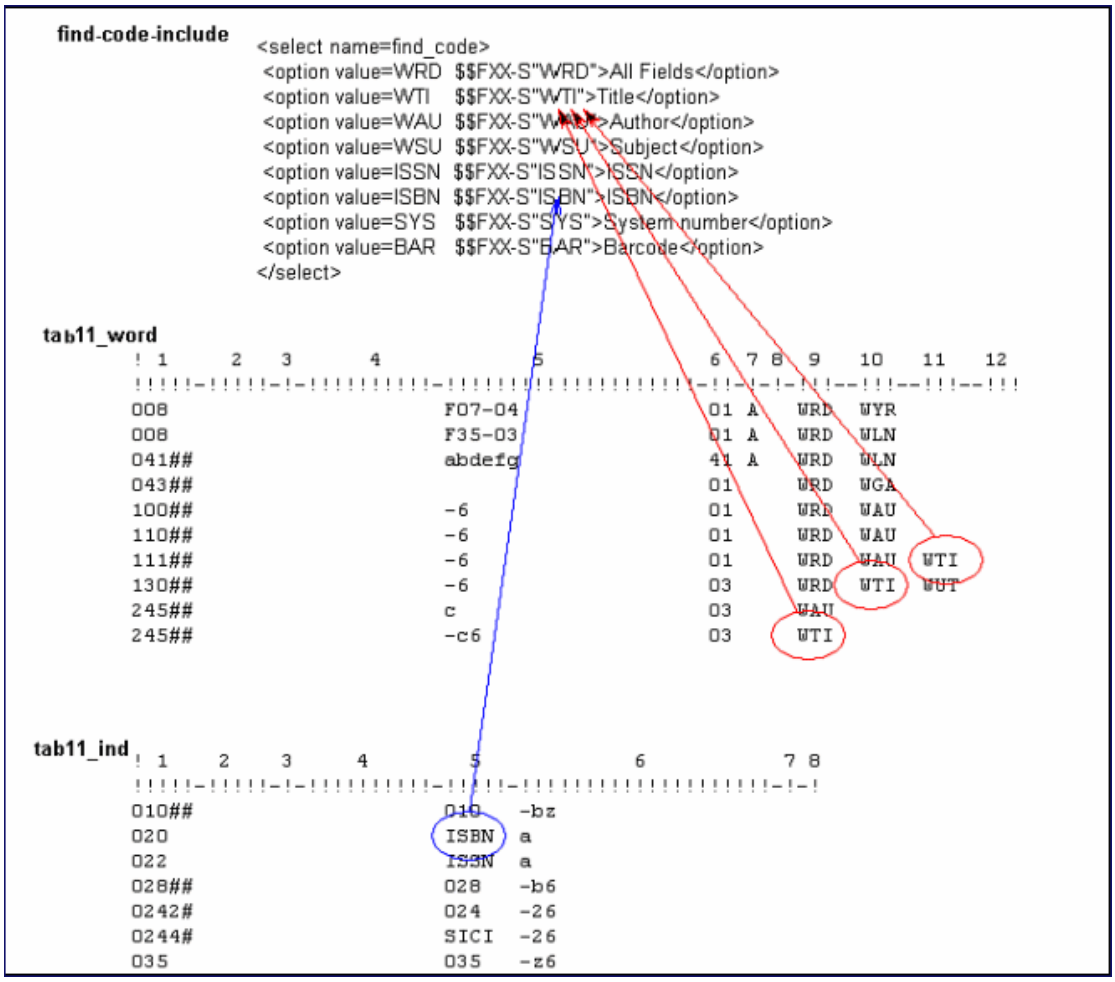
```
<select name=scan_code>
<option value=TIT $$5000-S"TIT">Title (omit initial
articles)</option>
<option value=AUT $$5000-S"AUT">Author</option>
<option value=SUB $$5000-S"SUB">Subject</option>
<option value=SUL $$5000-S"SUL">Subject - L.C.</option>
<option value=SUM $$5000-S"SUM">Subject - MESH</option>
<option value=SYS $$5000-S"SYS">System number</option>
<option value=PLA $$5000-S"PLA">Place of Publication</option>
<option value=PUB $$5000-S"PUB">Publisher</option>
<option value=SRS $$5000-S"SRS">Series</option>
<option value=SHL $$5000-S"SHL">Location</option>
<option value=ISSN $$5000-S"ISSN">ISSN</option>
```

The scan-include-2 HTML file includes a special command that causes the user's last choice to be displayed as a default choice the next time the same type of search is made.

For example, if the user chooses the "Subject" option from the Browse screen, the $$5000_S "SUB" portion of the following line causes the system to recall the last option chosen, in this case, subject.

```
<$$5000-S"SUB">Subject
```

The following figure shows the relationship between the `tab11_acc` and `tab11_ind` tables which assign fields to headings indexes and the HTML code:



There are other pull-down menus that are set by the system. For example, a similar menu exists in the Holdings screen for the Select Sublibrary combo-box.

Additional Search Options

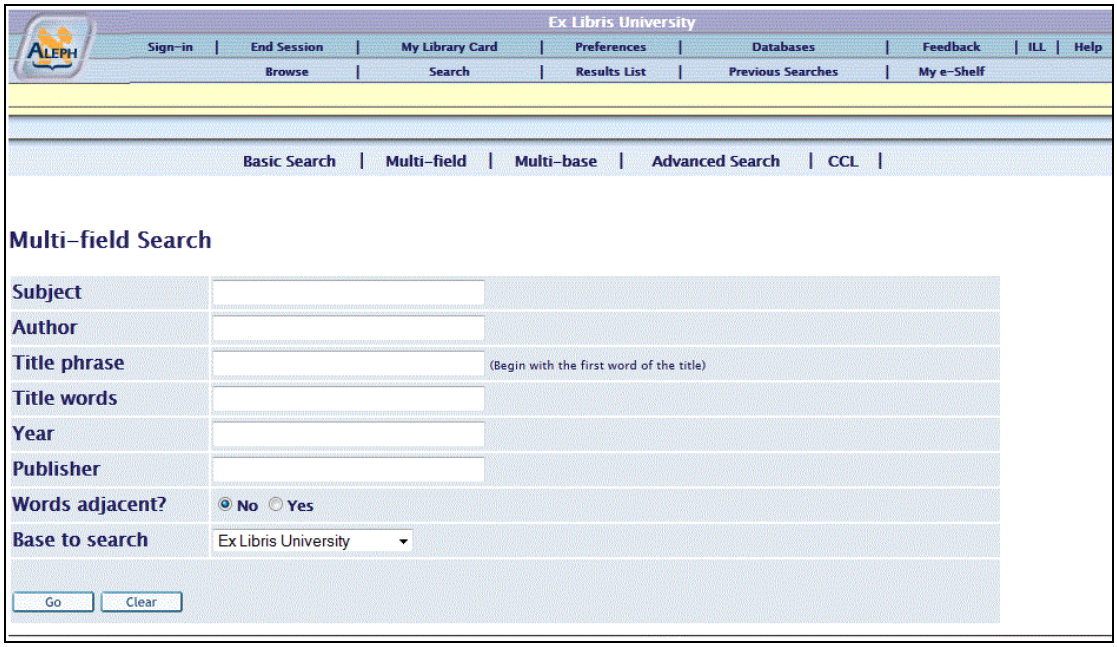
This section is intended to show you how search windows are set up in ALEPH. All the screens and codes shown in this section are pre-defined and should not be changed without consultation with ALEPH Customer Support.

In addition to basic Browse, Search and Direct Access functions, ALEPH offers a variety of more sophisticated search functions. From the Search window, the following are available:

- Multi-field Search
- Multi-base Search
- Common Command Language Search

(See Search for detailed explanations of all of these functions.)

The following *Multi-field Search* screen (filename "find-a") shows how the more sophisticated search features are incorporated into the Web OPAC:



This screen has a number of different fields for the user to fill in and each field is used for a pre-programmed purpose. For example, the ability to search the Subject index is defined in the HTML file by the following:

```
<tr>
  <th class=td2 align=left width=10% nowrap>Subject</th>
  <td class=td2 id="small">
      <input type="hidden" size=13 name=find_code value="WSU">
      <input size=35 name=request value="$0100">
  </td>
</tr>
```

Note the following points about this example:

- The word "Subject" in the first line may be changed or translated.
- The fact that the purpose of this field is pre-programmed and may not be affected by the user, is shown in line 3 by **input type="hidden"**.
- The name of the hidden field is the parameter **find_code**. Each hidden field on the screen has the same name.
- The index that will be searched is the "Words from Subject" index, indicated by **value="WSU"**. The codes for the indexes used in the screen must be ones that are defined in tab00.lng.

```
H WRD    W-001       00        01        Words
H WTI    W-002       00        02        W-titles
H WAU    W-003       00        02        W-authors
H WSU    W-004       00        03        W-subjects
```

- The name of the input field (the field that users can actually type in) is the parameter request. Each input field on the screen has the same name.

- The value of the input field is "$0100". Each input field on the screen must be given a unique value. The objective is to stores the values in a cache in the system's memory for future use.

The HTML code for the Author field is similar to the Subject field, with the addition of a hidden Boolean operator, **AND**, as shown below:

```
<tr>
  <th class=td2 align=left>Author</th>
  <td class=td2 id="small">
     <input type="hidden" size=3 name=request_op value="AND">
     <input type="hidden" size=6 name=find_code value="WAU">
     <input size=35 name=request value="$0200">
  </td>
 </tr>
```

Note that every Boolean operator is assigned a name. The name is the parameter **request_op**. Also note that in this example, the index being searched is the "Words from Author" index, represented by the code "WAU".

In the next example, for the Title field, the index being searched is the "Title Headings" index, represented by the code "TIT". Note that in this case, the search is on a headings list, and that the user must enter the title from the beginning text, although not all text.

```
<tr>
  <th class=td2 align=left>Title phrase</th>
  <td class=td2 id="small">
     <input type="hidden" size=3 name=request_op value="AND">
     <input type="hidden" size=5 name=find_code value="TIT">
     <input size=35 name=request value="$0300">
       <span class="text1">(Begin with the first word of the title)
</span>
  </td>
 </tr>
```

The ability to choose a base to search in is defined in the HTML file by the following:

```
<include>base-list-include\FXX\9000\
```

This line calls the base-list-include Web file, which is responsible for presenting the drop-down list containing the different bases. The content of the drop-down list is taken from tab_base.lng.

The drop-down list automatically selects the current base entry by using the placeholder $9000 when calling the base-list-include file.

This line also appears in the following Web files:

find-b
find-b-list-head
find-d
find-c
short-tail

The operation of the Boolean operators, AND, OR and NOT, in Web OPAC is as follows:

- Operations in parentheses are evaluated first.
- When there are no parentheses, the execution order (precedence) of the Boolean operators is as follows: NOT, AND, and OR.
- Search statements are evaluated left to right.
- The system recognizes the operators both in lowercase and uppercase.
- When the Search query is performed on a headings index (for example, TIT= gone but not forgotten), AND/OR/NOT are treated as real words, that is, not as Boolean operators.

In search statements with only AND and OR, operations in parentheses are evaluated first. Within any group of operations, NOT operations are done first from left to right followed by AND operations, and then by OR operations executed from left to right.

```
Search Statement     Executed as
----------------     -----------
A and B not C        A and (B not C)
A or B not C         A or (B not C)
A not B and C        (A not B) and C
A and B or C         (A and B) or C
A or B and C         A or (B and C)
A and (B or C)       A and (B or C)
```

Note that the operation of the Boolean operators is predefined and cannot be changed by the System Librarian.

For translation to other languages, see `tab_ccl_boolean_operators`.

Set Up the Search/Browse Page

The Search/Browse screen allows the user to make a search query and a browse query in the same page.

To implement this functionality, perform the following steps:

1. Replace the include file `scan-include-2` with `find-e-head`, in the following files:
   ```
   /alephe/www_f_eng/scan
   /alephe/www_f_eng/scan-acc-head
   /alephe/www_f_eng/scan-acc-head-hesci
   /alephe/www_f_eng/scan-acc-head-liedu
   /alephe/www_f_eng/scan-acc-head-usm10
   /alephe/www_f_eng/scan-acc-head-usm30
   /alephe/www_f_eng/scan-acc-head-usm90
   /alephe/www_f_eng/scan-acc-ug-head
   /alephe/www_f_eng/scan-ind-head
   /alephe/www_f_eng/scan-sys-head
   /alephe/www_f_eng/scan-word-head
   ```
2. Replace the include file `find-b-head` with `find-e-head`, in the following files:

```
/alephe/www_f_eng/find-b
/alephe/www_f_eng/find-b-list-head
```

3. Replace `find-b-0` with `find-e-0` in the following files:
```
/alephe/www_f_eng/base-list
/alephe/www_f_eng/base-list-usm90
/alephe/www_f_eng/find-b-head
/alephe/www_f_eng/find-b-top
/alephe/www_f_eng/find-include-buttons
/alephe/www_f_eng/find-include-buttons-usm90
/alephe/www_f_eng/head-1-bor
/alephe/www_f_eng/head-1-bor-ext-ill
/alephe/www_f_eng/head-1-nobor
/alephe/www_f_eng/head-1-nobor-ext-ill
/alephe/www_f_eng/head-1-nobor-ext-ill-pds
/alephe/www_f_eng/head-1-nobor-pds
/alephe/www_f_eng/head-replace
```

4. Remove the Browse link from the top control buttons.

Note that you can either use the browse/search functionality or the separate Browse and Search functionalities.

Floating Keyboard

Multi-script floating keyboard functionality is available in Web OPAC in the following screens:

- find-a (multi field search)
- find-m (multi base search)
- find-d (advanced search)
- short-tail

The floating keyboard can contain different writing scripts. The content of the floating keyboard can be edited in the floating-keyboard.xml file under ./alephe/www_f_lng.

A link such as the following should come after the **Go** and **Clear** buttons in the above Web pages:

For other languages click here

4.7 Result Screens

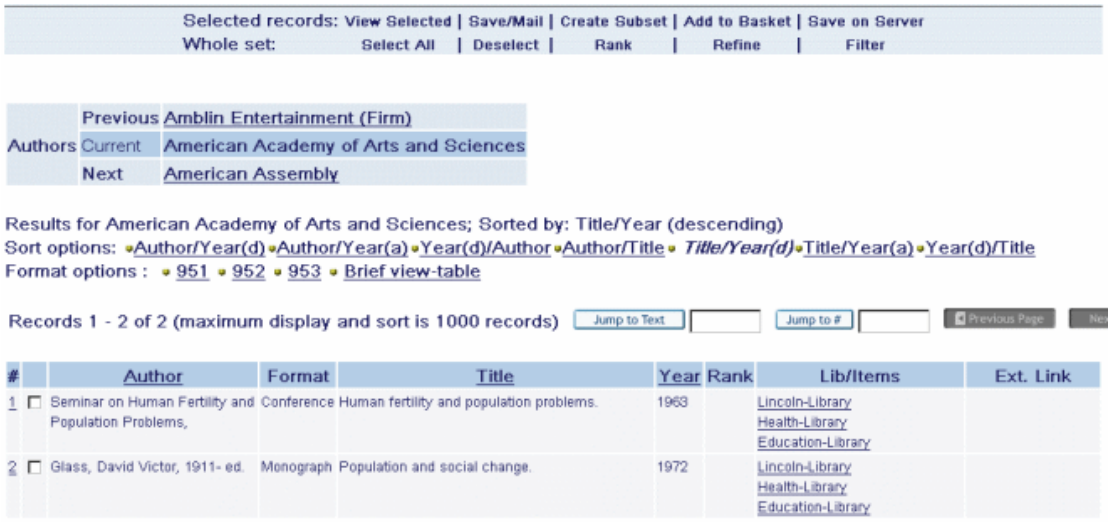
The following screens are discussed in this section:

- Brief View (Table Format)
- Other Brief Format Options
- Full View

- Sort
- Filter

4.7.1 Brief View (Table Format)

There are two brief views available in the Web OPAC, Brief View-Table and Brief View-List. This section is intended to show the reader how the Brief View-Table is set up. A basic knowledge of HTML is assumed. The Brief View, shown below, displays a list of records in a table format.



The Brief View-Table will be displayed when:

- The user chooses Brief View-Table format from the Preferences screen;

  or if

- The library has "000" set as the default display format in the Brief Format field for the profile that is in effect. The default Brief format is set in the Profile Information window in the Circulation GUI for the default profile (ALEPH). (See the Patron chapter in Circulation, Patron Profiles).

The **www_tab_short.lng** table defines which information will be displayed in the columns of the short display in table format in Web OPAC. The `www_tab_short.lng` table is located in the library's tab/ directory. You can edit `www_tab_short.lng`.

The order of the columns defined in `www_tab_short.lng` in the `alephe/tab` directory determines the order of the display, in conjunction with the short-a-head and short-a-body HTML files in `www_f_lng`. These files should have placeholders in sequential order, and should have as many placeholders (from 0300) as the number of columns defined in the `www_tab_short.lng` table.

The field you define for a column can be dependent on the bibliographic record format (FMT). You can also set up alternative definitions by assigning them the same identifying number in column 1 of this table.

Example:

```
!-!-!!!!!!!!!!-!!!!!!!!!!!!-!!-!!-!!!!-!-!-!!!!!!!!!!-!!-!!!!!-!!!!!-
>
1 L Author     02---A01---A 00 00 0100 S 9 -          BK 1####  700##
1                                                     SE 260##
1                                                     ## 1###
```

In the above example the "Author" column in the Web OPAC will be taken from fields 1#### or 700## if the document is a book (BK), from field 260## if the (SE) document is a journal, and from field 1#### in any other instance.

TYP is a virtual field defined in the tab_type_config.lng table. The TYP field indicates the type of record - (Electronic Resource, Web Resource, Map, Kit, and so on), based on coding in the LDR (pos. 06,07), 006 (pos. 00), 007 (pos. 00,01) and 008 (pos. 23) fields.

You can set a single column to display more than one field of information. In the following example, a single column is set to display more than one field of information, by repeating the column number (in this example the column number is 2) in column 1, entering the name of the field in column 3 (in this example the field name is title), and entering the field codes in columns 12 and 13 in each of the column number lines.

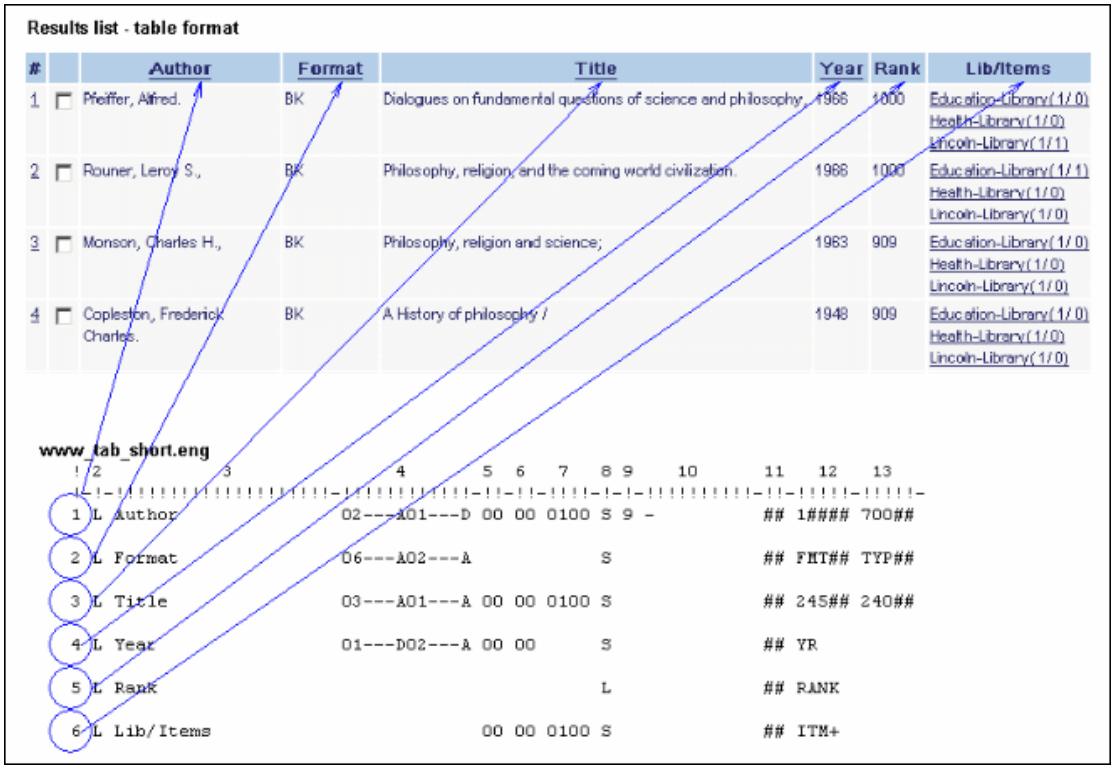
Example:

```
2 L Title      03---A01---A 00 00 0100 S              BK 245## 240##
2                                                     ## 245## 240##
```

In the above example, both column 12 and column 13 have field codes. If the record does not contain the field code in column 12, the system searches for an alternative using the field code entered in column 13.

You can display more than one field name in a column by defining a second field name by repeating the column number in col. 1, entering a name in col. 3 (this name will not display), and entering the data element code in col. 12, and (optional) alternative codes in col. 13 and on.

The following figure shows the relationship between the `www_tab_short.lng` table and the Brief View in Table format (`short-a-head` and `short-a-body`):



short-a-head and short-a-body

short-a-head and **short-a-body**, together with `www_tab_short.lng`, determine the order in which columns are displayed in the Brief View Table. short-a-head and short-a-body are HTML files located in the `alephe/www_f_lng` directory. These files should have placeholders in sequential order, and should have as many placeholders (starting from 0300) as the number of columns defined in the `www_tab_short.lng` table.

The short-a-head file defines the display of the headers according to `www_tab_short.lng`. To remove a column, for example, the "Year" column, you must delete a placeholder line from "`short-a-head`" and "`short-a-body`". If you want to add a column from `www_tab_short.lng`, you must add the relevant parameter to `short-a-head` and `short-a-body`.

The `short-a-body` file defines the display of the content of each column.

Here is an example of a typical definition for `short-a-head`. The placeholders following are used for column captions, except for 0100 and 0200. 0100 is a placeholder for a number link to the full record display, and 0200 is a placeholder for a check box. The other placeholder numbers relate to `www_tab_short.lng` where 0300 is the first column defined, 0400 is the second column defined, and so on.

Example:

```
<tr class=tr1>
  <th class=text3>$0100</th>
  <th class=text3>$0200</th>
  <th class=text3>$0300</th>
  <th class=text3>$0400</th>
```

```
  <th class=text3>$0500</th>
  <th class=text3>$0600</th>
  <th class=text3>$0700</th>
  <th class=text3>$0800</th>
</tr>
```

Note that the column headers here must match with the corresponding contents columns defined in short-a-body and must have the same order.

Here is an example of a definition for a corresponding short-a-body file:

Example:

```
<tr valign=baseline>
  <td class=td1 id=centered width="1%" valign=top>$0100</td>
  <td class=td1 id=centered width="1%" valign=top>$0200</td>
  <td class=td1 $2300 width="20%" valign=top>$0300  </td>
  <td class=td1 $2400 width="5%" valign=top>$0400  </td>
  <td class=td1 $2500 width="30%" valign=top>$0500  </td>
  <td class=td1 $2600 width="5%" valign=top>$0600  </td>
  <td class=td1 $2700 width="5%" valign=top>$0700  </td>
  <td class=td1 $2800 width="20%" valign=top>$0800  </td>
</tr>
```

The **width** parameter specifies the width of each column as a percentage of the width of the whole line.

In our examples, the placeholders refer to the following column headings:

```
Column heading   Placeholders
Author           300
Format           400
Title            500
Year             600
Rank             700
Lib/Items        800
```

If you want to add another column, for example, *Collection*, follow these steps:

1. Add a line number, 9, to column 1, and a new field name (collection) to www_tab_short.lng, (located in the library's tab directory).

2. Add a new line to short-a-head (located in the www_f_lng directory) with a sequenced placeholder.

   For example:

   ```
   <th class=text3>$0900</th>
   ```

3. Add a new line to short-a-body (located in the www_f_lng directory) with sequenced placeholders.

   For example:

   ```
   <td class=td1 $2900 nowrap valign=top>$0900  </td>
   ```

4.7.2 Other Brief Format Options

Apart from the Table format, the Brief View can also be set up to display in a List format which is customizable.

To customize Brief View List formats, enter the xxx01/tab/www_f_tab_short_999 file and modify format as required:

```
!1  2     3     4     5     6     7     8     9     10    11    12
13    14
!   15   16    17    18    19    20    21    22    23    24    25
26    27
!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!!!!!-!!!!!-!!!!
!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!!!!!-!!!!!
951 1#### 24### ITM5# DIS
952 24### DIS
999 1#### 24### ITM5# DIS
```

Note that the 999 format is reserved for the default view, and that the 950 format is reserved for the Full View. Custom formats must be numbered from 951-998. When adding a field to a custom format, make sure that the field is also present in the `xxx01/tab/edit_doc_999.lng` file. The fields are displayed according to the rules set in `edit_doc_999.lng`.

Regardless of the default, patrons using the OPAC can choose their own preferred display formats from the Preferences screen.

To edit the brief view display formats available from the Preferences screen:

1. Modify the file `alephe/www_f_eng/option-display` (for signed-in patrons) and the file `alephe/www_f_eng/option-display- session` (for guests).

2. Find the following lines in the Web file which contain the display format options available from the Preferences screen:

   ```
   <td class=td2 align="left" valign=top>Format:
     </td>
     <td class=td2 valign=top>
      <span class=text9>
      <input type="radio" name=SHORT_FORMAT
        value="951"
        $$0300-C"951">Option 1(951)
      <input type="radio" name=SHORT_FORMAT
        value="952"
        $$0300-C"952">Title only  (952)
      </span>
     </td>
   ```

3. Edit them as required.

   In addition, a library patron who has signed in can update his Personal Profile. To update the brief display format options available from this screen:

Open `alephe/www_f_eng/bor-profile-display`, and find the following lines:

```
<td class=td2 valign=top>
      <span class=text9>
      <input type="radio" name=SHORT_FORMAT
             value="951"
       $$0400-C"951">Option 1(951)
<input type="radio" name=SHORT_FORMAT
       value="952"
       $$0400-C"952">Option 2 (952)
</span>
</td>
```

These are the lines that govern the display of the brief format options available. To add options, edit these lines as required.

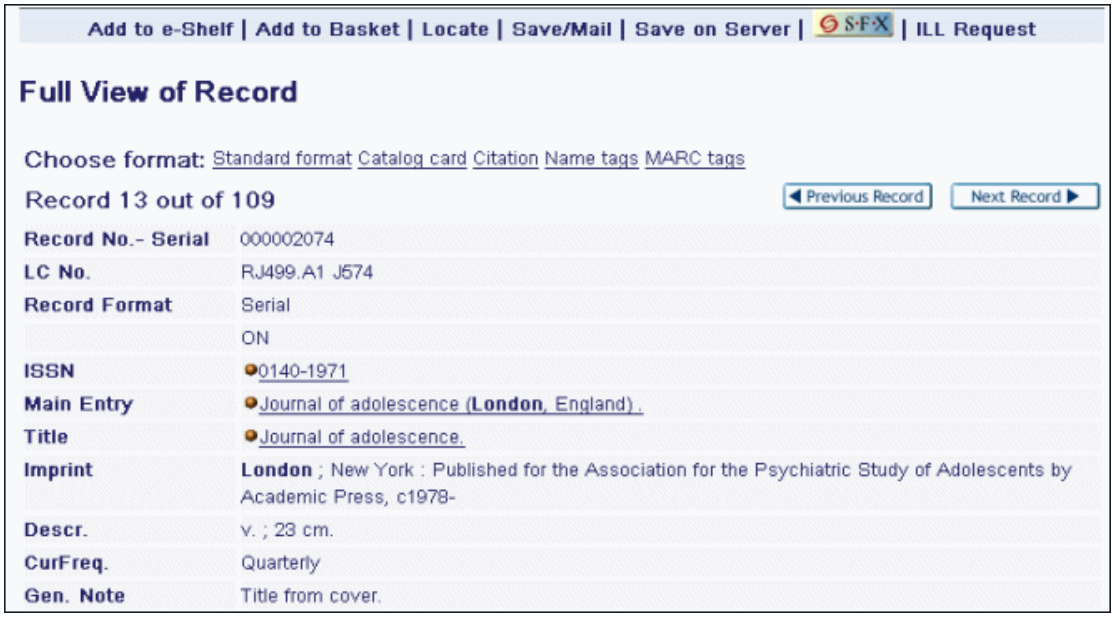
4.8 Full View

There are five Full view formats that the user can choose from:

- Standard
- Name and MARC Tags
- Catalog Card
- Citation

In addition, you can define your own Full view (which is a subset of Standard view) and Save/Mail options.

Full View - Standard



This screen is made up of full-set-head (which displays the information above the table and the top border of the table), full-999-body (which determines the

characteristics of the font and the width of the columns) and full-set-tail (which closes the table and displays the information below the table, in this case, the icons for Previous Record/Next Record).

The following is an example of the HTML code for the `alephe/www_f_eng/full-999-body` file:

```
<tr>
  <td class=td1
      id=bold
      width=15%
      valign=top
      nowrap>
    $0100
  </td>

  <td class=td1 $AL02>
    $0200
  </td>
 </tr>
```

The first <td> section shown defines the field tag; the second <td> section shown defines data display.

The values that begin with dollar signs, $0100 and $0200, are placeholders for data. The system fills in the actual data when the screen is displayed. Do not change these placeholders - otherwise the program will not recognize the data to display.

You can determine exactly which fields are displayed in the Full view by editing the edit_doc_999.lng table. Both the numeric code and the field name are set in this table.

Interaction with Syndetics

If your library works with Syndetics, you can enrich your Web OPAC by adding a cover image to the full record display. Cover images are derived from the record's ISBN and an interaction with the Syndetics Internet server. You can initialize this service by adding the value `ICON` to the `edit_doc_999.lng` table. For example:

```
## ICON                 D LCover Image          Y       E       W
```

Cover images are hyperlinked to additional catalog enrichment elements, such as summaries, author information, excerpts, book reviews, and so on. When the cover image is clicked, a pop-up window appears, containing the available enrichment options.

To indicate that your library has a license to use Syndetics, add the following line to `$alephe_tab/license`:

```
SYNDETICS           Y
```

The default is N.

The client code assigned by Syndetics to a customer must be defined in $alephe_root/www_server.conf under the variable name syndetics_client_code.

For Example:

setenv syndetics_client_code your_code

Note that the Syndetics link will not work if this code is not defined.

In the Full View of Record window, there are services which are available for a particular field.

The services currently available are scan (acc or ind), find (acc), aut, aut-bib, engine, and find-m-acc. These services are defined in `xxx01/tab/tab_service`.

- **scan** activates the display of a browse list. All the browse lists defined for the field in the tab11_acc table are displayed for you to choose from.
- **find** activates the Find command. All the word lists defined for the field in the tab11_word table are searched.
- **aut** activates the display of the authority record that is linked to the field's heading.
- **aut-bib** is used only for an Authorities library (XXX10). This service allows the user to initiate a search in an Authorities 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 Bibliographic records, that are indexed using the same text as one of the AUT fields. This service invokes a FIND query from the AUT record on a SCAN (Z01) list in the BIB library. The parameters for the query (that is, which BIB library and BIB SCAN list) are defined in the www_f_<lng>/service-aut-bib HTML file. The FIND command is truncated at 50 characters and normalized using filing procedure **98**. You can strip unwanted subfields from the AUT record field. Do not include a character conversion in this procedure; the BIB library's relevant character_conversion program is used. This is particularly useful for libraries which use UDC/DDC classification for subject indexing of BIB records, and for an authority database in which each UDC/DDC is an authority record with textual descriptor.
- **engine** activates an external search engine, using parameters stored in the xxx01/tab/tab_z121 table. The parameters include the search engine name, the URL and the search query string. For example

```
!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
GOOGLE
http://www.google.com/search?q=$0100&meta=lr%3D%26hl%3Den
```

  You can add a new search engine, using the same syntax as above. Note that the search engine name is the same as the link in the www_f_lng/service-engine HTML file, except that in tab_z121, it must be uppercase, and in service-engine, it can be in uppercase or in lowercase. For example:

```
<select name="engine">
          <option value=google>Google</option>
                .../...
       </select>
```

If you do not want the search engine to search the whole field but only specific subfields, you can define the subfields in column 3 of `tab_service`. If you leave column 3 blank, the search is made on the whole field.

Here is an example:

```
! 1   2                                3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
100##  www_f_service_engine           a
```

- **find-m-acc** activates a parallel find command on a number of databases simultaneously. The databases to search are defined in the www_server.conf table in alephe, in the environment variable www_parallel_search_base. The base codes are entered in one string, with a hyphen separating each one. Here is an example:

  ```
  setenv www_parallel_search_base       USM01-UNI01
  ```

  The creation of a link is defined by:

  The presence of *M* in Col. 10 of the field defined in edit_doc_999, and the presence of the field tag in tab_buf_Z403, or

  The presence of *S* in column 10 of edit_doc_999.lng as shown in the following example:

  ```
  !1   2   3      4     5 6 7         8          9 0 11  2  13
  4 15 16
  !!-!!!!!-!-!!!!!!!!!!-!-!-!!!!!!!!!!!!!!!!!!!-!-!-!!!-!-!!!!!-
  !-!!-!!
  ## 500##                D LGen. Note           Y S     E
  ```

  The *S* link type modifies a URL with HTML tags so that no copyright warning is shown and no patron authorization is checked. The *S* link type is designed to be used in free-text fields.

Full - Name & MARC Tags

These screens are made up of the following files in the `alephe/www_f_lng` directory: `full-set-head` (which displays the information above the table and the top border of the table), `full-000-body` (which determines the characteristics of the font and the width of the columns) and full-set-tail (which closes the table and displays the information below the table, such as the icons for Previous/Next).

In the `full-000-body` file, do **not** change the placeholders for data (placeholders are the values that begin with a dollar sign ($)).

These formats display all of the fields that are included in the Catalog record. The names of the fields are derived from `tab01.lng`: MARC tags are taken from Col. 2; name tags are taken from Col. 9.

Services (aut, aut-bib, and so on) cannot be displayed in these formats, and editing is not possible.

Full - Catalog Card

This screen is made up of the following files: `full-set-head` (which displays the information above the Catalog Card and the top border of the Card) and `full-xxx-body` (which determines the font characteristics of the text displayed in the Catalog Card) and full-set-tail (which displays the bottom border of the Card and displays the information below the table, such as the icons for Previous Record/Next Record).

In the `full-xxx-body` file, do **not** change the placeholders for data (placeholders are the values that begin with a dollar sign ($)).

You can control the fields that are displayed in the Catalog Card by editing the `edit_doc.lng` table, and changing the format numbered "037", and/or editing the lines in `edit_paragraph.lng` that are used by the `edit_doc.lng 037` format.

Full - Citation

This screen is made up in the same manner as Full - Catalog Card. The only difference is that Citation uses the "038" format in the `edit_doc.lng` table.

Define Your Own Full View

You can define other paragraph-style formats in addition to this Catalog Card format supplied with the system. To do so:

1. Add a new format (with its own format number) to the edit_doc.lng table, such as, for example, format number 012.

2. Add the fields you want to be included in the new paragraph to edit_paragraph.lng table. For example:

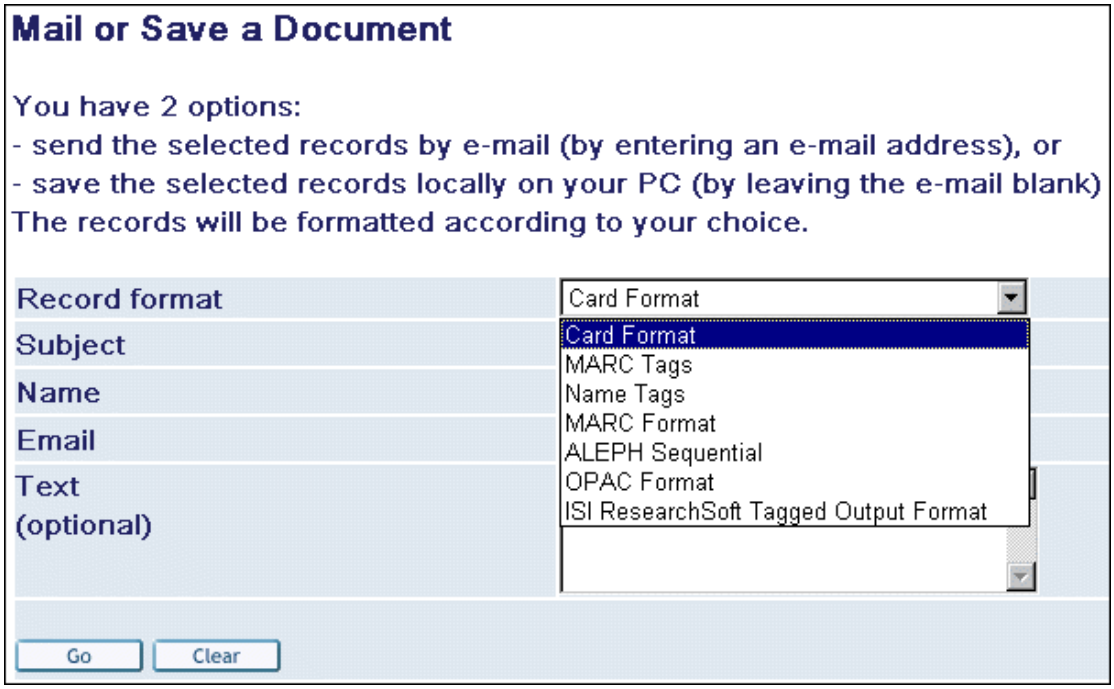
```
012 LOC## D    ##
012 1#### D    ^.^
012 245## 9            .
012 260## W ^          .
012 300## D ^          .
```

3. In the full-set-head HTML file, the new format option might look like this:

```
<a href=&server_f?func=full-set-
set&set_number=$0800&set_entry=$0500&format =012>
```

Mail or Save a Document

This screen allows the user to save documents to disk or to forward them via e-mail. This screen is based on the full-mail HTML file. There are several record format options available for saving and sending records.



The format options reside in the **include-select-format** include file.

Here is an example of a file definition:

```
<select name=format>
          <option value=037 $$FXX-S"037">Card Format</option>
          <option value=001 $$FXX-S"001">MARC Tags</option>
          <option value=002 $$FXX-S"002">Name Tags</option>
          <option value=997 $$FXX-S"997">MARC Format</option>
          <option value=998 $$FXX-S"998">ALEPH Sequential</option>
          <option value=999 $$FXX-S"999">OPAC Format</option>
          <option value=777 $$FXX-S"777">ISI ResearchSoft Tagged
Output Format</
option>
   </select>
```

You can use formats from `edit_doc.lng`, or the 777 format, which is defined in the `edit_doc_777` table. This is the format used to import records into reference managers such as Endnote. The conversion is driven by the `edit_doc_777` table in the tab directory of the bibliographic library. In this table, the ALEPH tags that should be exported are defined. When a record is saved or mailed in the 777 format, the program uses the **www-set-mail-777-00** form. More formats are described above in this Full view section.

4.9 Holdings Screens

The Holdings screen contains three parts:

- Top - BIB record information
- Middle - HOL record information
- Bottom - Item information.

This section explains how you can set up the display of the BIB and HOL information.

The user can click a link on the screen to place an expand, photocopy or hold request. The actual text that is displayed for the link is determined by message numbers 9058, 9059 and 9060 in the `www_f_heading` file.

The Expand link, displays complete information about the item. The actual word that is displayed for the link is determined by message 9058 (for example, 9058 L expand) in the `www_f_heading` file. This "Item Record Expand View" is useful because it enables the System Librarian to display just a few columns of item information in the main Holdings screen, and to save the details for the expanded view. The file that controls the expanded view is `item-global-exp`.

The display of the *Due date* column of the Holdings window, is taken from `$alephe_root/error_lng/circ_status_holding`, error codes 8000 - 8013.

An item that has a hold or a booking request is reported as Requested in the Due Date column. To also report items that only have photocopy requests as Requested:

In $alephe_root/www_f_lng/item-global-body, replace the following line:

<td class=td1>$0400</td>

With:

<td class=td1>$1800</td>

The Heading code 8013 is used when the item has a Z30-ITEM-PROCESS-STATUS. This message appears with the process status description taken from the ADM library's `tab15.lng`. If there is no Z30-ITEM-PROCESS-STATUS, message number 8012 (*On Shelf*) appears.

The display of the BIB record at the head of the list of items is defined by the 012 format, in `xxx01/tab/`**`edit_doc.lng`**.

The display of the HOL record at the head of the list of items is defined by the `xxx60/tab/edit_doc_99.lng` table.

To implement the functionality of the display in the No. of Requests column, the following placeholder ($1600), must appear in `./alephe/www_f_eng/item-global-body` as follows:

```
<!--No Requests-->
<td class=td1>$0900 $1600</td>
```

You can set up the Holdings screen to include links to external locations, such as HTML files or addresses. The links themselves are defined in fields (such as 856) in the BIB and / or the HOL record. The link can be to any machine that is served by a HTTP daemon.

Normally, multiple 856 links can be seen displayed in the Holdings information section of the record, just above Item information.

To include fields from the bibliographic record with electronic contents into the section for the holdings record, for example the 856 field, ensure the following:

- That the `tab/tab_expand` table of the holdings library includes the following line:

  ```
  BUF-Z403    expand_doc_hol_bib
  ```

- That the selected fields (for example, 856) are defined in the `edit_doc_999.lng` (with M in column 10) table of the holdings library as in the following sample:

  ```
  ## 856##                D LElectronic Location Y M     E
  ```

- That the selected fields (for example, 856) are defined in the `tab_buf_z403` table of the holdings library as in the following example:

  ```
  ########## get_buf_z403_bib                856##
  ```

A staff user can see the details of a patron who has borrowed an item by clicking on an underlined due date. In order that the due date link is active, Col.17 in tab31 of the ADM library, must be set to Y for the specific patron's status.

4.10 User Screens

Library patrons can view and carry out their library activities via their online **Library Card**. The My Library Card screen is derived from the `alephe/www_f_lng/bor-info` HTML file.

The format of the bibliographic information is set to 015 in edit_doc.lng configuration file of the BIB and ILL libraries.

You can customize the display on the Library Card in accordance with library policy and setup. For example, your library does not want to allow Advance Booking. To remove the Advance Booking option, open `alephe/www_f_lng/bor-info` and comment out the following lines (that is, by placing **<!--** before the HTML code that controls the information you do not want to be displayed, and by placing **-->** after the HTML code that controls the unwanted information):

```
<tr>
  <td class=td1 align=left>
    <a
href="javascript:replacePage('http://ram19:8995/F/GJ91IA1283R2R62I37H
3NPND8HHTU97IL1A9QIN3N816H5SRJQ-04033?func=bor-item-schedule');">
       Advance bookings</a>
  </td>
  <td class=td1 align=left>
     $2000
  </td>
 </tr>
```

Note that you must **not** change the placeholders for data (placeholders are the values that begin with a dollar sign ($)), although you can comment them out.

All of the other HTML files accessed from `bor-info` (`bor-item-schedule`, `bor-ill`, `bor-photo`, `bor-cash`, and so on) can be customized in the same way. Note that these files have a different structure - they include head, body, and tail files.
Every bor-xxx file contains a link to bor-xxxxx-exp. Some contain a link to bor-xxxxx-exp-head files. These files contain expanded activity information. The bor-xxxxx-exp-head files contain relevant bibliographic information.

4.11 Submitting a Photocopy Form

4.11.1 General

Library patrons can place photocopy requests on items. The photocopy submission window is derived from the `alephe/www_f_lng/item-photo-request` HTML file.

In order to display BIB information on the top of photocopy submission form, insert the following lines in alephe/www_f_lng/item-photo-request-head:

- <input type=hidden name="bib_doc_num" value=$3400>
- <input type=hidden name="bib_library" value=$3500>

4.11.2 Photocopy Submission for Article Document

Libraries can decide to use a pre-filled photocopy submission form when requesting documents that are articles (that is, the document's TYP field = "AN").
When the regular photocopy submission form is used, all fields are displayed empty on the form. A special photocopy submission form can be used for displaying pre-filled information: alephe/www_f_lng/item-photo-request-tail-***an***. This form is optional. In order for this form to be used for article documents, define the following in $alephe_root/www_server.conf:

www_photo_article_prefill Y

According to this setup, when you request documents that are articles (the document's TYP field = AN), the following photocopy form fields are filled with bibliographic information:

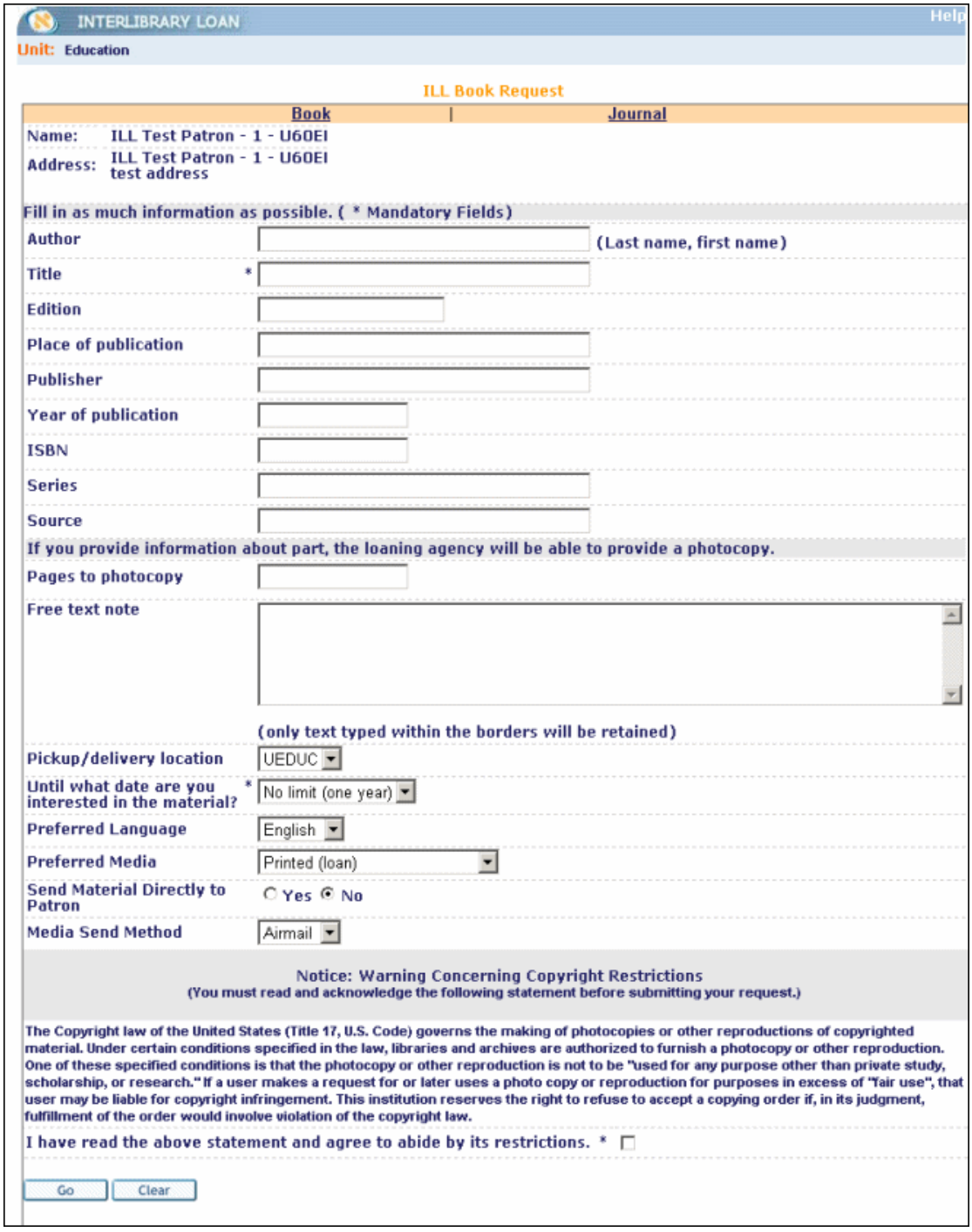
- Author of title (Z38-AUTHOR) –720##, subfield a. (All occurrences of 720, up to the limit of the field size).
- Title of article (Z38-TITLE) – 245##, subfield a + 246, subfield a
- Pages (Z38-PAGES) – 773##, subfield **g** + 300##, subfield a (there is a one byte space delimiter between 773##, subfield g and 300##, subfield a.
- Z38-NOTE-2 will be populated with 001##

For all these fields, fields that do not have $$9 subfields are used. Only for Author of title (Z38-AUTHOR), is a field that has $$9A used.
If the field does not exist in the document, the photocopy form field is left empty.

Note: The TYP field is populated using the `expand_doc_type` routine. For more information, refer to the *Aleph 21 System Librarian's Guide – Indexing*).

4.12 ILL Screens

There are several kinds of ILL screens. Each one is tailored to a specific type of item. Following is part of the screen for a Book/Thesis ILL Request:



The HTML code for the Author field of the ill-book-request file is given below:

```
<tr>
  <td class=td2 width=10% nowrap>Author</td>
<td class=td2> </td>
  <td class=td2>
      <input size=40 maxlength=180 name="BIB___1001_a_1500"
value="$1500">
            <span class="text1">(Last name, first name) </span>
  </td>
 </tr>
```

Pay special attention to "BIB___1001_a_1500".

100 is the tag for author;
1 is the first indicator
a is the subfield.
1500 is the placeholder.

All of the fields follow this same format. Make sure that the tags used in any forms you create are ones that are listed in tab01.lng in the ILL library.

If you want a particular field to be mandatory, make sure that it has an "M" in the "BIB...." code, as in the following example:

```
<tr>
  <td class=td2>Title</td>
  <td class=td2 width=1%>
      <span class=text2>*</span>
  </td>
  <td class=td2>
      <input size=40 maxlength=120 name="BIB_M_2451_a_1700"
value="$1700">
  </td>
 </tr>
```

4.13 Setting up the Special Request Function

The Special Request function allows a user to place Special Requests for items that are in the library's collection, although they are not listed in the computerized catalog. The Special Request form lets the user create a request for these items. The link for making a special request only displays in the menu bar if the patron's profile includes permission for placing a special request. This procedure creates three ALEPH records for an item:

- Item Record
- BIB Record
- Hold Request

To facilitate the Special Request function, the following must be defined in the new-record HTML file:

For an Item Record

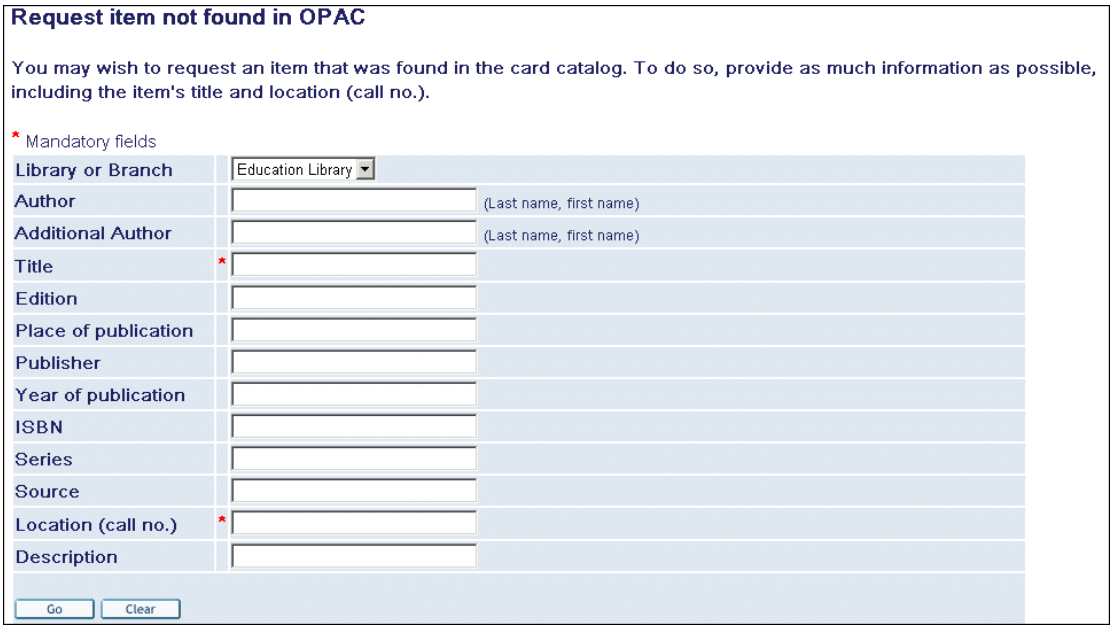
- The sublibrary the item record will be written in.
- The ADM library that the sublibrary is part of.
- The BIB library that the ADM library is related to,
- The item collection (optional), status, material type, and location type (optional).

For a BIB Record

- The BIB database that the ADM database and sublibrary are related to. This is the definition which determines where the BIB records will be written to.
- Field tags for BIB information.

To set up the Special Request function, follow these steps:

1. Define the sublibrary in which the item record will be written. To do so, edit the **new-record** file from the `alephe/www_f_lng` directory which defines the Special Request Form.



The following is the HTML of the Library or Branch field, which must be the uppermost field.

```
<table cellspacing=2 border=0 width="90%">
<tr>
  <td class=td2 width=20% nowrap>Library or Branch</td>
  <td class=td2 width=1%> </td>
  <td class=td2>
     <select name="LIB">
        <option value=USM01-USM50-UEDUC $$3200-S"USM01-
USM50-UEDUC">Education Library</option>
        <option value=USM01-USM50-UHLTH $$3200-S"USM01-
USM50-UHLTH">Health Library</option>
        <option value=USM01-USM50-ULAW $$3200-S"USM01-USM50-
ULAW">Law Library</option>
        <option value=USM01-USM50-ULINC $$3200-S"USM01-
USM50-ULINC">Lincoln Library</option>
      </select>
```

The default data for an item is as follows:

- Name of sublibrary - five characters in "" for example, UEDUC (mandatory)
- Name of collection - up to five characters, for example, AB-D (optional)
- Item Status - two digits, such as 70 (CLOSED STACK) (mandatory, must be in tab15.lng)
- Material type - up to five characters, must be in tab25, for example, BOOK (mandatory)
- Location type, one digit, for example, 8 (optional)

Use hyphens to fill unused positions. Let's look at the following example:

```
<input type=hidden name="UEDUC" value="GEN---70-BOOK--8">
```

The following is an example of a sublibrary definition in the new-record file from the alephe /www_f_lng directory. Each section is separated by a hyphen:

```
<option value=USM01-USM50-UEDUC $$3200-S"USM01-USM50-
UEDUC">Education Library</option>

<option value=USM01-USM50-UHLTH $$3200-S"USM01-USM50-
UHLTH">Health Library</option>
```

If these definitions are correct, a valid item is created and written in the ADM library's Z30 listing of library items.

2. Decide which bibliographic fields appear in the Special Request form and which of these fields are mandatory.

   The following is a example of definitions that appear in the new-record file from the `alephe /www_f_lng` directory.

```
<tr>
  <td class=td2>Author</td>
  <td class=td2> </td>
  <td class=td2>
      <input size=30 maxlength=180 name="BIB___1001_a_1000"
value="$1000">
       <span class="text1">(Last name, first name)  </span>
  </td>
 </tr>
```

   The parts of the following line, which is taken from the example above, are separated by an underline. The first three characters are always BIB. The fifth character can be M, meaning that data entry is mandatory. Characters 7-11 contain field tag + indicators. The last four digits are the same as the placeholder.

   `BIB___1001_a_1000`

   The following is the HTML of the mandatory Title field:

```
<tr>
  <td class=td2>Title</td>
```

```
  <td class=td2>
      <span class=text2>*</span>
  </td>
  <td class=td2>
      <input size=30 maxlength=120 name="BIB_M_2451_a_1200"
value="$1200">
  </td>
 </tr>
```

3. Define an item status for special requests. From the ADM library, use UTIL G/5/15 to edit tab15.lng which defines item statuses and various parameters connected to these statuses. Following is a sample from the table:

```
! 1   2  3  4        5                        6 7 8 9 1 1 1 1 1 1
!                                                 0 1 2 3 4 5
!!!!!-!!-!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-!-!-!-!-!-!!
.../...
15A   70 ## L Closed Stack-1                  Y N C N Y N N Y Y 00
15A   71 ## L Closed Stack-2                  Y N C N Y N N Y Y 00
.../...
```

4. In addition to creating the hold request for the user who fills in the Special Request, the system can automatically create a second hold request for a "CATALOGER" user. This feature is available for libraries that want to trap the item when it is returned, in order to send the item to cataloging. The request is generated for the user ID "CATALOGER". Therefore, there must be a user defined whose ID is "CATALOGER" in order that this feature be activated.

   Column 8 refers to Hold requests. To enable Special Requests, you must set this column to C (closed stack). If the item is in a closed stack, hold requests will always be allowed.

4.14 Support For Right-To-Left Text Display In All Formats

ALEPH contains language support for all major written languages as Unicode has been fully implemented throughout the system. This language support includes languages that are written from right to left, such as Arabic or Hebrew. In order to display data that is right-aligned and right-oriented in Arabic or Hebrew characters, follow these steps:

1. Set the environment variable "direction_support" to "Y" (in the file ./alephe/www_server.conf).

2. Add the $AL<NN> syntax, as seen in the following example:

```
<tr>
  <td class=td1
      id=bold
      width=15%
      valign=top
      nowrap>
    $0100
  </td>
  <td class=td1 $AL02>
```

```
      $0200
    </td>
  </tr>
```

Here is an example of a full format display of a bibliographic record containing Latin, Arabic and Hebrew characters:

4.15 Adding an SFX Link Button

The `tab_sfx` table, found in the BIB library's `data_tab`, defines which icon is displayed for the SFX link. The table determines what SFX services are available when a record is found in a specific base.

If one of the services is available, the table also controls which icon will be displayed. This control is done by defining, in column 3, into which index on the HTML page, the icon set in column 2 will be inserted. The index has to be used on the HTML page for it to be displayed. For example, if the `tab_sfx` rows state
:

```
USM01                www_f_sfx           11 N getAuthorEmail
USM01                www_f_sfx_em        12 N getCitedJournal
```

then if the SFX service *getAuthorEmail* is available for a record found in USM01, the `www_f_sfx` icon will be inserted into index 11. For the full-set-head HTML page to display this icon, use the following definition:

```
<td class="bar" align="center" valign="bottom" nowrap>
      $1100
```

```
      $1200
  </td>
```

instead of :

```
<td class="bar" align="center" valign="bottom" nowrap>
      <a href='javascript:open_window("&server_f?func=service-
sfx&doc_number=$0600&line_number=0000&service_type=RECORD");'>
        <img src="&icon_path/f-sfx.gif"
             border=0
             alt="SFX services"></a>
  </td>
```

If none of the services is available, a dimmed and unavailable SFX link button will be inserted into index 11 of the HTML page.

If the searched base is not found in `tab_sfx`, the `www_f_sfx` icon will be inserted into index 11 of the HTML page.

If more than one SFX service is required, columns 1-4 do not have to be repeated. For example:

```
USM01                www_f_sfx            11 N getAuthorEmail

                                               getCitedJournal
```

means that either *getAuthorEmail* or *getCitedJournal* have to be available for the `www_f_sfx` icon to be displayed.

To prevent problems of loading a Record Full View page when the SFX server is busy, make sure that the file sfx-button-data exists under $alephe_root/www_f_lng. and that the "full-set-tail" and "direct-tail" files contain the following lines at the end of the file:

```
<script>
// Function to process SFX buttons
function ProcessSFX(SFX_String) {
    var sfx_elem;
    sfx_elem = document.getElementById("sfx_button");
    sfx_elem.innerHTML = SFX_String
}
</script>
<script src="&server_f?func=load-sfx-
data&doc_number=$0900"></script>
```

You may set up tab_sfx so that the conditional SFX button is linkable if one of the following SFX services is available for the BIB title:

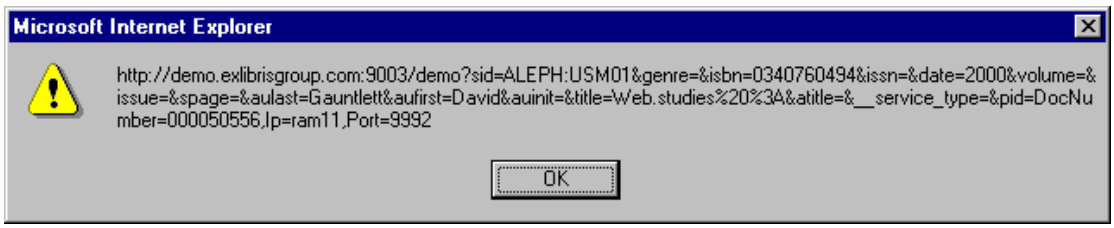
Set the following in $alephe_root/www_server.conf:

setenv sfx_kev Y

The list below is the full set of the services supported only when the environment variable sfx_kev = Y. Otherwise, only the services getFullTxt, getAbstract, getReference, getHolding and getTOC are supported.

- getWebService
- getWebSearch
- getTOC
- getSubject
- getReference
- getPayPerView
- getMessageNoFullTxt
- getHolding
- getFullTxt
- getDOI
- getDocumentDelivery
- getCitedReference
- getCitedRecord
- getCitedJournal
- getCitedGenome
- getCitedBook
- getCitedAuthor
- getBookReview
- getAuthorEmail
- getAuthor
- getAlikeRecord
- getAbstract
- getMsgNoRecomm
- getRecommendation
- getSelectedFullTxt

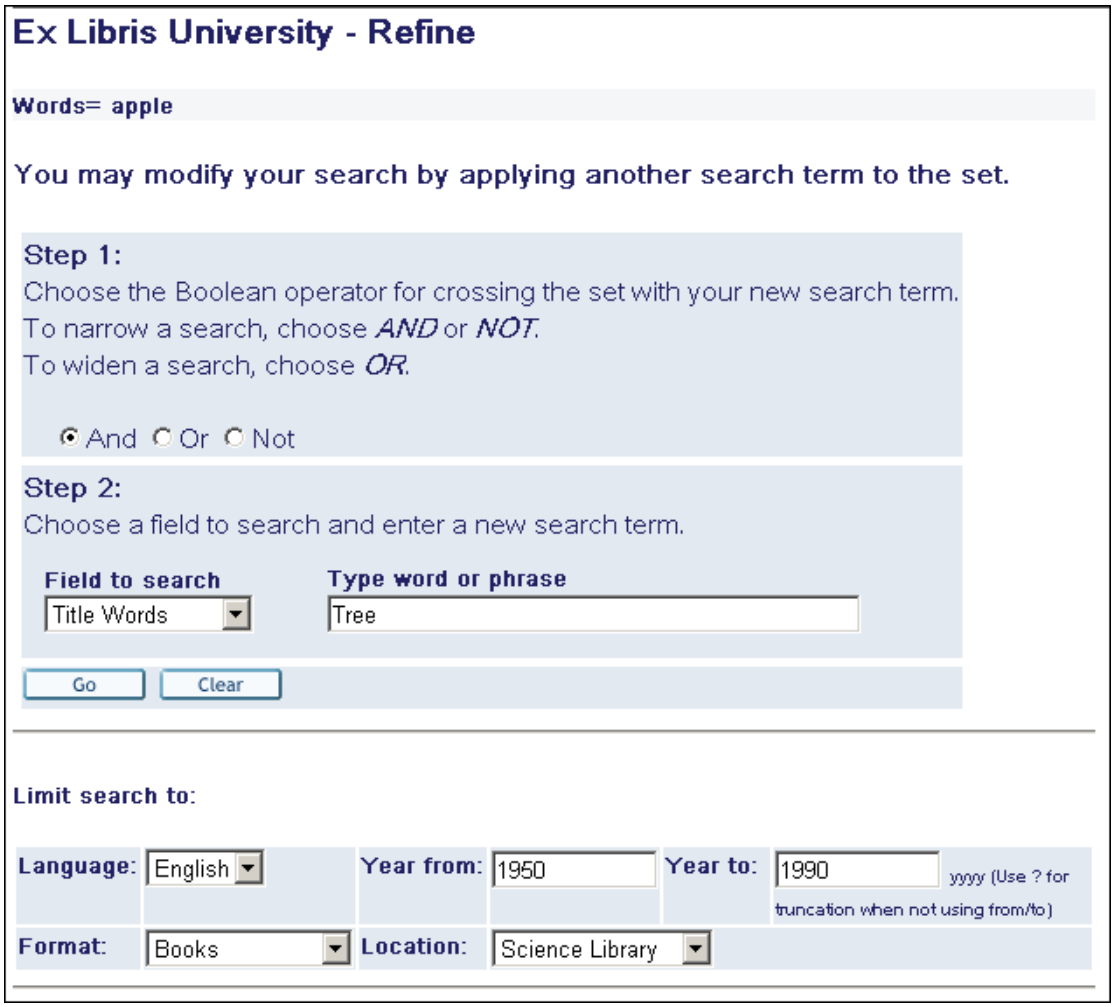
The following message, which pops up before the SFX window appears, presents the OpenURL link:



If you do not want this message to appear, change the following variable in sfx-record (in the www_f_lng directory) from `var debug =0` to `var debug =1`.

4.16 Setting Up the Refine Window

The Refine function allows users to produce a new results set that is narrower than the original one. The Refine window appears when users click **Refine** button from a Results List:



It is divided into the following sections:

- Step 1 - The user can choose one of the Boolean operators AND, OR, NOT.
- Step 2 - The user can select a field to search, such as Author, Title, Subject and so on, and enter a new search term.
- Limit search to - The user can further limit his search by filling in the fields in the Limit search to: section. For search limits to be included in the Refine window, the

following lines must be present in the `./alephe/www_f_eng/short-refine` file:

```
<hr>
<include>find-include-filter
<hr>
```

These filters include filtering by language, year, format, and location. To modify these options, edit the ./alephe/www_f_eng/find-include-filter file.

4.17 Photocopy Information

When viewing patrons photocopy requests details (in my library card), the note-1 and note-2 are concatenated into a single note field. To view the notes separately, add the following in ./alephe/www_f_lng/bor-photo-exp:

```
<tr>
<td class=td1 id=bold align=left>Note 1:</td>
<td class=td1 align=left> $1600</td>
</tr>
<tr>
<td class=td1 id=bold align=left>Note 2:</td>
<td class=td1 align=left> $1700</td>
```

5 Sort

A user can sort a results list by clicking on a link in all of the brief views in the Web OPAC. ALEPH sorts the records in the table according to the sort rules set for the column. This is defined by the predefined sort options set up in the `www_f_sort_heading.lng` table in the library's tab directory. The set remains in the last order selected, and the sort factor is displayed at the top of the screen. If the set contains more records than the sort threshold allows, the set cannot be sorted. The maximum sort limit is 1000.

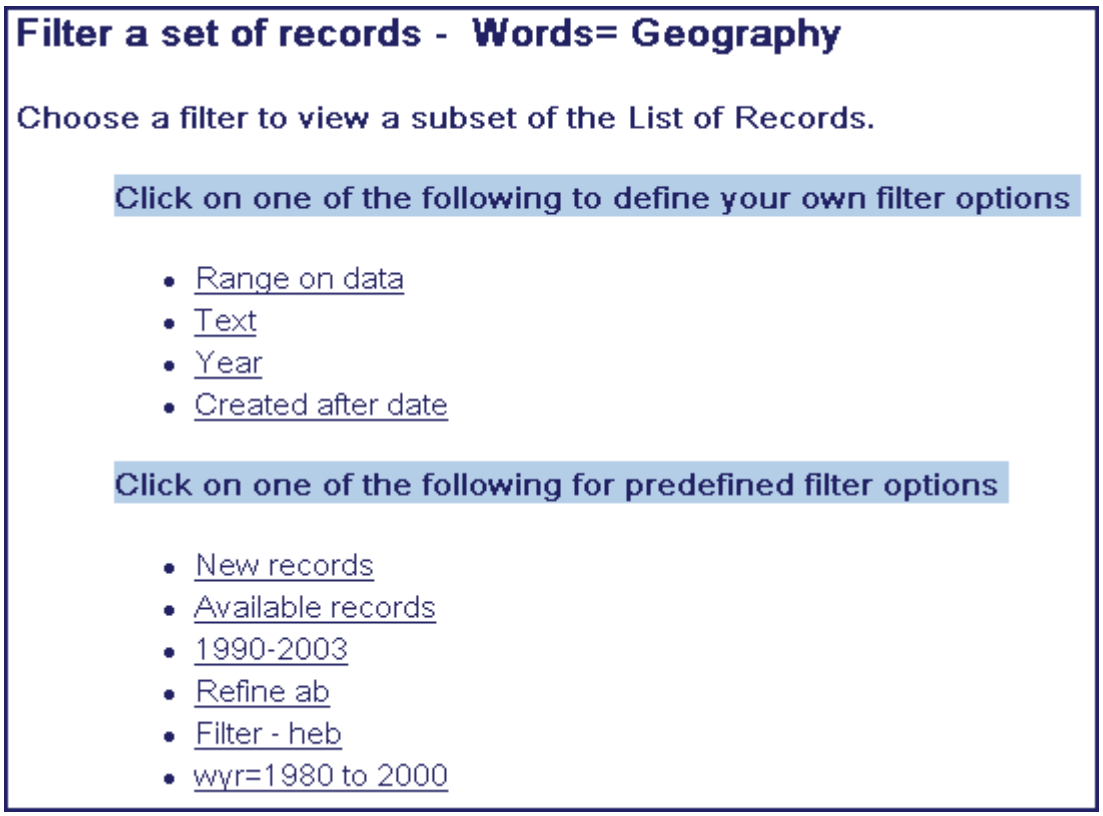
Example of the www_sort_heading.lng table:

```
!1   2 3   4       5                                  6
!!!!!!!!!!!!-!!!!!!!!!!!!
```

```
!!-!!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-!!!!!-!-!!!!!-
!!!!!-!-!!-!
!
01 95 008            260## c
08 0
4
02 01 1#### a
00 0
0
03 11 245## a     2 240##       2
00 0
0
04 22 050##          LOC## hjl
00 0
0
05 11 440## av    2
00 0
0
06    TYP##          FMT##
00 0
```

6 Filters

The Filter feature lets users define filter options after carrying out searches, creating their own filters or using predefined filter options.



When the user chooses one of the first four choices on the list, the relevant screen is displayed, as detailed below.

- Range on data - defined in short-filter-r
- Text - defined in short-filter-s
- Year - defined in short-filter-y
- Created after date - defined in short-filter-a

The other choices on the list, which appear below the line, are derived from the various files in `www_f_lng`, and further refine them. Each predefined filter option on the above screen is explained below so that you can understand the general concepts. Editing takes place in the short-filter file.

Range on data
The Range on Data filter displays records containing terms relating to the text in the range specified by the user.

As well as allowing the user to filter by an entire field's content, you can also specify a subfield by setting up the HTML page accordingly (`./alephe/www_f_eng/short-filter-r`) as shown in the following example:

```
<select name=field_code>
         <option value="245__a" $$0500-S"245__a">Title</option>
         <option value="100" $$0500-S"100">Author </option>
         <option value="110" $$0500-S"110">Corporate Author</option>
         <option value="111" $$0500-S"111">Meeting</option>
         <option value="260__a" $$0500-S"260__a">Place of
publication</option>
         <option value="260__b" $$0500-S"260__b">Publisher</option>
      </select>
```

- Use underscore ("_") instead of hash ("#") to specify "any indicator".
- If no indicators are specified (as in "100", "110", and "111" in the example above), this implies - "##" - any indicators. Note that "100__" is equivalent to "100", and may be used for improved clarity).
- If no subfield is specified, the entire field's content is checked.
- Only one subfield code is allowed in each line.

New records
This filter uses short-filter-a as its basis and specifies that records updated within the last seven days should be displayed. (The system checks the CAT field in the Catalog record to determine the date.)

```
<li><a href="&server_f?func=short-filter-a&start_date=&DATE-D007">
New records</a>
```

If you wanted to specify records for the past year, you would change "DATE-D007" to "DATE-Y001". In these examples, "D" stands for day and "Y" stands for year.

Available Records
This filter displays records containing items located on the shelf.

```
<li><a href="&server_f?func=short-filter-3">
Available records</a>
```

This option uses the short-filter file as its basis.

1980-2000
This filter displays records that were published between 1980 and 2000.

```
<li><a href="&server_f?func=short-filter-
y&from_date=1980&to_date=2000">
1980-2000</a>
```

This option uses the short-filter-y file as its basis, and further specifies that the "from" date should be 1980 and the "to" date should be 2000.

Refine ab
This filter displays all records of the current set that have words beginning with the letters *ab*.

```
<li><a href="&server_f?func=find-
c&CCL_TERM=wrd=ab?%20and%20SET=$0100">
Refine ab</a>
```

This option uses the find-c files as a basis and specifies that the system should search the "wrd" index for words that start with *ab* (The codes for the indexes are defined in `tab00.lng`).

By including **%20and%20SET=$0100"** , the Refine ab option specifies that the current set of records should be used as the basis for the filter.

wyr=1980 to 2000
The following filter displays all records of the current set that have the years 1980 to 2000 in the "wyr" (words in year field) index. (The codes for the indexes are defined in `tab00.lng`). The code for word index years used by the filter is hard coded, therefore it must be defined as WYR.

```
<li><a href="&server_f?func=find-c&CCL_TERM=wyr=1980-
>2000%20and%20SET=$0100">
wyr=1980 to 2000</a>
```

By including **%20and%20SET=$0100"** , this option specifies that the current set of records should be used as the basis for the filter.

Filter - eng

The following filter will display all records of the current set that have "eng" (for English) in the "wln" (words in language field) index.

```
<li><a href="&server_f?func=find-
c&CCL_TERM=SET=$0100%20and%20wln=eng">
Filter - eng</a>
```

6.1 Setting Up Bases and Filters

Instead of retrieving all the records in a particular physical database (for example), you may want to limit/filter the records retrieved to those in a particular location, in a particular format, in a particular language, and so on.

Such limiting can be either a pre-filter or a post-filter. With a pre-filter, the user selects a particular database before starting the search; with a post-filter, the user takes the results of the search then selects a location, format, and so on, by which to limit them.

You set up post-filters by using `alephe/www_f_lng/short-filter`. (For English, `www_f_eng/short-filter`). All the examples below can be set up in `short-filter`.

Location as Post-Filter
For example, you might have locations as post-filters (with this definition in `alephe/www_f_lng/short-filter`):

```
<tr bgcolor="#eeeeee">
            <td colspan=2>
<div class="text3">
Limit by location
</div>
<p>
<form>
<div class="text4">
Stanhope University Libraries
<br>
<select name="variable name">
<option>
<option value= "&server_opac/find-c?CCL_TERM=wcl=birk^and^SET=$0100"
target="err
or">Birks Religious Studies Reading Room
<option value= "&server_opac/find-c?CCL_TERM=wcl=bldr^and^SET=$0100"
target="err
or">Blackader-Lauterman Architecture and Art
Â…Â…Â…<and so on.>
```

The index code you use depends on the indexes you define in the tab11_word table.

Formats as Post-Filters

Alternately, you can have formats as post-filters (with this type of definition in `alephe/www_f_lng/short-filter`):

```
.../...
Material Type
<br>
<select name="variable name">
<option>
<option value="&server_opac/find-
c?CCL_TERM=SET=$0100^and^wtp=Archive" target="e
rror"> Archive
<option value="&server_opac/find-
c?CCL_TERM=SET=$0100^and^wtp=Graphic" target="e
rror"> Graphic
<option value="&server_opac/find-c?CCL_TERM=SET=$0100^and^wtp=Kit"
target="error
"> Kit
<option> - - - - - - - - - - - - - - -
<option value="&server_opac/find-c?CCL_TERM=SET=$0100^and^wtp=Map"
target="error
"> Map
Â…<and so on.>
```

The index code (WFT vs. WTP) you use depends on the indexes you define in the tab11_word table.

Language as Post-Filter

You can also set languages as post-filters (with this definition in `alephe/www_f_lng/short-filter`):

```
<form>
<div class="text4">
Language
<br>
<select name="variable name">
<option>
<option value="&server_opac/find-c?CCL_TERM=SET=$0100^and^wln=ara"
target="error
"> Arabic
<option value="&server_opac/find-c?CCL_TERM=SET=$0100^and^wln=chi"
target="error
"> Chinese
<option value="&server_opac/find-c?CCL_TERM=SET=$0100^and^wln=cze"
target="error
"> Czech
```

Other Post-Filters

The `alephe/www_f_lng/short_filter` comes with some built-in filters:

```
<form>
<div class="text4">
Define your own limit options
<br>
<select name="variable name">
<option>
<option value="&server_opac/short-filter-r">Range on data
<option value="&server_opac/short-filter-s">Text
<option value="&server_opac/short-filter-y">Year
<option value="&server_opac/short-filter-a">Recent acquisitions
<option>
```

Pre-Filters (Databases)

With a pre-filter, the user selects a particular database before starting the search.

The `alephe/tab/tab_base.lng` table is used to specify pre-filtering. Any keyword/Boolean search which can be performed through the Find/Search function can be the basis of a database. The filtering criteria are specified in column 9 of the `tab_base.lng` entry. Additional information on `tab_base.lng` can be found in the ALEPH UTIL Menus - 16.02 document, section UTIL Y/3.

The `tab_base.lng` entry for a physical database looks like this:

```
!         1          2           3            4     5     6     7   8
9
!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!-!!!!!!!!!>
USM01                  Ex Libris University                    USM01
N
```

Columns 4, 5 and 6 are used for file extensions. Column 8 defines whether or not to use Z0102 to create a logical base.

Excluding Suppressed, Deleted Records

Almost all sites will want to have the database(s) which the public uses exclude suppressed records, logically-deleted records, and possibly substandard, "fast-cat" records. This is accomplished with a specification like this:

```
!         1          2           3            4     5     6     7   8     9
!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!-
!!!!!!!!!>
USM01_PAC              USM01 PAC                                USM01 N
wrd=alldocuments not (wst=suppressed or wst=deleted or wst=circ-created)
```

wst is a word code which is specified in `xxx01/tab/tab11_word` as a word index on the STA field:

```
STA##                                          01     WRD  WST
```

(The STA field looks like this: "`STA $$aSUPPRESSED`"; "`STA $$aDELETED;`" and so on, and is found in the bibliographic record.).

Note

Bases apply only to Words and Browse searches. Direct index (number) searches are always done on the entire bibliographic file. You may want to remove or change the tag of OCLC numbers or other overlay-matching numbers if records are being logically deleted.

Locations as Pre-Filters

Here is a sample definition for `tab_base.lng` entries for databases that are filtered by sublibrary:

```
!         1          2           3            4     5     6     7   8     9
!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!-
!!!!!!!!!>
LAW                    Law Library                              USM01 Y
wsl=ulaw
```

What you specify for the code depends on which indexes you have defined in the tab11_word table. If you have defined a single WCL with both the sublibrary and collection then you can use that:

```
LOC##                      bc                     01     WCL
```

If you have defined separate sublibrary and collection indexes:

```
LOC##                      bc                     01     WSC
LOC##                      b                      01     WSL
LOC##                      c                      01     WCL
```

then you can use the WSC and/or WSL.

Note

If none of your collection codes are used as sublibrary codes, then you only need to define the single WSC (or WCL) index with LOC (or PST) subfields b and c.

You might think of specifying "EXUCAT" in column 7, that is, having LAW be a database built on EXUCAT rather than EXU01, thus making it unnecessary to repeat the EXUCAT filtering criteria. You cannot do this. The value in column 7 must be a physical library; it cannot be just another base.

Formats for Pre-Filters

The `tab_base.lng` entries for databases which are filtered by format might look like this (using the wfm index, built on the FMT field):

```
!         1          2           3            4     5     6     7   8
9
!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!-!!!!!!!!!>
SERIALS                Serial (USM01)                          USM01 N
wfm=SE
MONOGRAPHS             Monographs (USM01)                      USM01 Y
wfm=BK
```

Or like this (using the wtp index, built on the TYP field):

```
!         1          2           3            4     5     6     7   8
9
!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!-!!!!!!!!!>
 TPE                    Periodicals          USM90             USM90
Y wtp=se
```

Note

The FMT field is a physical field which is stored in the BIB record. The TYP field is a virtual field generated by `expand_doc_type`. It uses the configuration setup in a `tab_type_config.lng` table, for example using LDR pos. 06,07; 006 field pos. 00; 007 field pos. 00, 01; and 008 pos. 23 to arrive at more detailed formats than those specified in the FMT field. To index it, you need to have a:

```
WORD       expand_doc_type         tab_type_config
```

entry in the `xxx01/tab/tab_expand` table.

7 Error Messages

You can edit and translate the error messages that are displayed to users by editing the `www_f_heading` file in the `$alephe_root/error_lng` directory. Following is an example from that file:

```
9031 L There is a copy available; request not accepted.
9032 L Patron is missing an ILL library. Unable to process request.
9033 L Error in patron ILL library. Unable to process request.
9035 L Too many hits. Refine request.
```

```
9036 L Error parsing find request.
9037 L Set is too large for adjacency check. Refine request or skip
adjacency check
```

Key to table:

- **Column 1** is the message number. Do not change this number.
- **Column 2** was the code for the character set of the column heading and is now irrelevant.
- **Column 3** is the text of the message that is displayed to the user.

Note

When writing error messages with quotation marks ("), use only the HTML definition for quotation marks - *"* - to ensure that they display correctly.

8 Default Text for Links

The cataloger uses the LKR field to create links from one bibliographic record to another. The following table shows the possible types of LKR links for BIB records.

Subfield	Description	
a	Value	Use
	UP	"up" link to another BIB type record. A record may have only one link of this type.
	DN	"down" link to another BIB type record. Multiple links of this type are possible.
	PAR	parallel link from BIB record to BIB record
	ANA	analytic link between BIB records of different levels
	ITM	link from BIB record to ADM record (used for items bound together)

When no other text is available (that is, the Z103-LKR-TEXT-N field, which is the test of the up link, is empty), the default text that appears next to the LKR (link) field in the Full View of the bibliographic record is defined through the library's tab/tab05.lng table as follows:

```
! 1                       2                              3
!!!!!------!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
NONE       Click to view related record
```

NONE is reserved for both the Search functionality in GUIs and the Web OPAC.

The LKR fields can be displayed according to various sorts.

9 Rank

Web OPAC users can rank sets, using the words that appear in the Rank Set window. Web OPAC ranks the sets, using the words of the search query.

There are three factors involved in setting the ranking weight of a word. The weight is calculated by:

- the number of occurrences of the word in the record (a document gets a higher ranking if a significant word occurs several times).
- the weighting factor defined in column 9 of `tab00.lng` (adds weight according to the particular word list).
- an initial factor attributed to each of the words used for ranking (in case more than one word is used), based on the frequency of the word in the whole database. The more frequent the word is (the more results its search obtains), the lower its initial weight.

To define a weighting factor, set the last two digits in the W section of `tab00.lng` (Col.9) as appropriate. Other sections (ACC and IND) should have 00 in column 9. Here is an example of a definition:

```
!   2     3   4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H WRD   W-001      00       01       Words
H WTI   W-002      00       02       W-titles
H WAU   W-003      00       02       W-authors
H WSU   W-004      00       03       W-subjects
H WUT   W-005      00       02       W-Unif .Titles
H WSM   W-006      00       03       W-MeSH Subj.
H WNO   W-007      00       01       W-Notes
H WNC   W-008      00       02       W-Table of contents
H WPU   W-009      00       02       W-publishers
H WYR   W-010      00       02       W-year
```

Note that the system only uses the WRD index.

10 Adding a New Index

To add a new index, follow these steps:

Step 1: Define the Index Type
Decide which type of index you want. There are three types: Direct, Words (Find), or Headings (Browse).

Step 2: Add New Index to the Indexing Tables

The indexing tables are:

- `tab11_acc` (for headings indexes)
- `tab11_ind` (for direct indexes)
- `tab11_word` (for words indexes)

Include in the relevant indexing table the code you specified in `tab00.lng` for each field which you want to have sent to your new index.

If you specified "IND" in `tab00.lng`, put the entry in the `tab11_ind` table for direct indexes. If you specified "ACC", put the entry in the `tab11_acc` table for headings indexes, and if you specified "W-nnn", put it in the `tab11_word` table for word indexes. Consult the ALEPH Configuration Guide - 16.02 document for other parameters.

The following is a sample of the `tab11_ind` table:

```
! 1    2    3     4        5            6           7 8
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!-!-!
010##                    010   -bz
010##                    STIDN -bz
020                      ISBN  az
```

The following is a sample of the `tab11_acc` table:

```
! 1    2    3     4        5            6           7 8
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!-!-!
100##                    AUT   -e468
245##                    TIT   -ch68                  2
```

The following is a sample of the `tab11_word` table:

```
! 1     2   3      4            5             6  7 8  9   10   11
12
!!!!!-!!!!!-!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!-!!-!-!-!!!--!!!--!!!--
!!!->
100##                    -6                    01     WRD  WAU
245##                    c                     03     WAU
245##                    -c6                   03     WTI
```

The index codes can be up to 5 characters (in tab00.lng, tab11_acc and tab11_ind).

Step 3: Add New Index to the tab00.lng Table
Go to `xxx01/tab/tab00.lng` and define a code for the new index. Make sure the code is not in use already.

Note

Although the code can be up to five characters, by convention, three characters are used. For a Direct index, specify "IND" for the index type. For a Headings index,

specify "ACC" for the index type. For a Words index, find the next "W-nnn" value and use that for the index type. Consult the Database Management Guide for other parameters.

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 table (`tab11_ind`, `tab11_acc`, `tab11_word`), they must be defined in `tab00.lng` in order to define the index name in column 11.

Example:

```
!   2     3   4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H SYS   IND     21 00       00       System Number
H BAR   IND     21 00       00       Barcode
```

Step 4: Run the Index Job
Open the Cataloging GUI and click Services to access the **Build Indexes to the Catalog** submenu options.

- For a Direct index, run p_manage_05.
- For a Headings index, run p_manage_02,
- For a Words index, run p_manage_01.

Note
On large databases, these indexes take a number of hours to run - the larger the database, the longer it takes to run the index. This job locks the library.

Step 5: Adding New Indexes to the Browse and Search Screens

- A Direct index can be added to either the Browse or the Search screen. Add it to the Web Include files
- If you want an index to be searchable on the Web, it has to be added to an HTML file. For a Headings index, add it to `www_f_lng/scan-include-2-xxx01`.

Example:

```
!   2     3   4 5  6  7 8   9   10            11
!-!!!!!-!!!!!-!-!!-!!-!-!---!!-!!!!!-!!!!!!!!!!!!!!!!!!!
H SYS   IND     21 00       00       System Number
H BAR   IND     21 00       00       Barcode
```

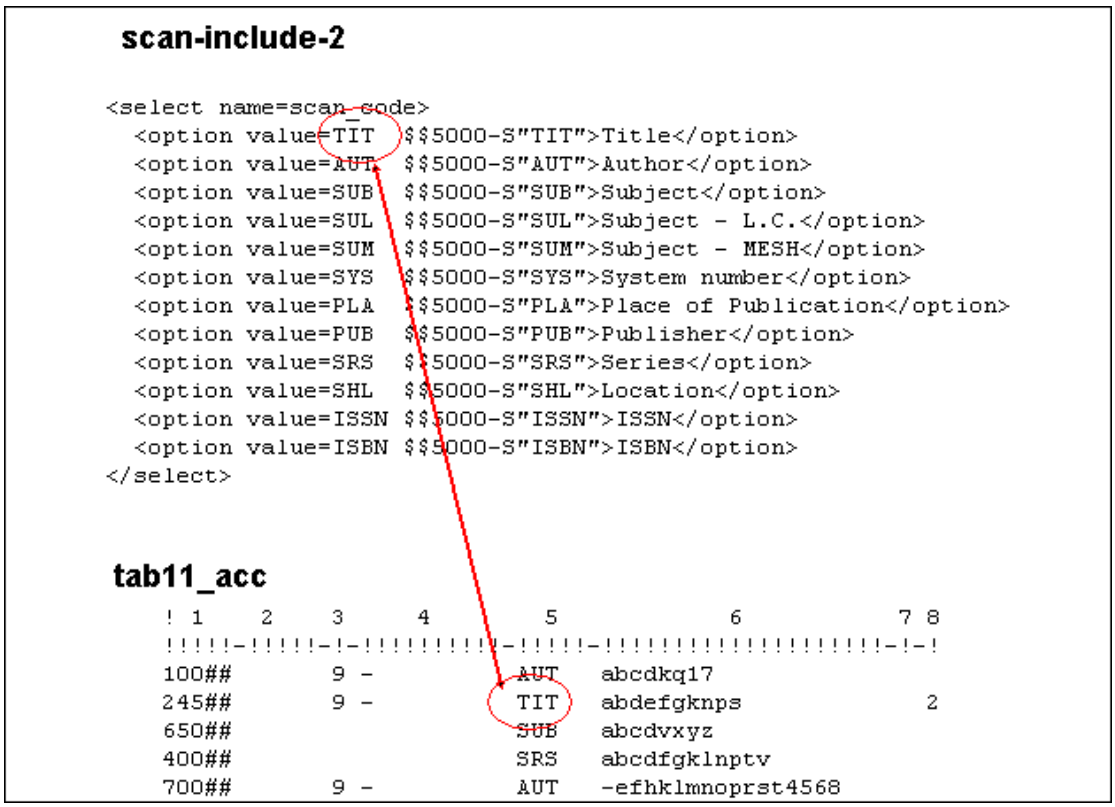
For a Words index (search), add it to `www_f_lng/find-code-include`.

Example:

```
<select name=find_code>
     <option value=WRD $$FXX-S"WRD">All Fields</option>
     <option value=WTI $$FXX-S"WTI">Title</option>
```

```
 <option value=WAU $$FXX-S"WAU">Author</option>
 <option value=WSU $$FXX-S"WSU">Subject</option>
 <option value=WSU $$FXX-S"WSM">Medical Subjects</option>
</select>
```

The following figure shows the relationship between the tables and the HTML code:



11 Setting Up the SDI Function

The SDI interface allows patrons with SDI permission to define SDI (Selective Dissemination of Information) profiles for themselves, based on their areas of interest. Patrons can access and modify their SDI profiles from the patron information (My Library Card) screen by clicking on the SDI Profile link from the menu bar. SDI profiles are stored in a special Oracle table, Z325, which is located in the USR00 library.

An SDI request defines a search query, a sublibrary to search (a patron can also set his SDI request regardless of location), a list of bases to search, a frequency with which the SDI query will be run, expiry date for the SDI request, the suspend period, and the character set. When the system runs the query (p_sdi_01 service), the records that have been added or updated in the database since the last time the query was run are retrieved. The resulting set of records is mailed to the patron. The format of the records is defined in the patron's SDI profile. If a patron clicks **Online Search** from his SDI profile window, then the Z325-last-action-date is updated and no SDI letter will be sent to the patron regarding items found in the online search.

SDI functionality in ALEPH is controlled by the setup of the `p_sdi_01` service. By using the job scheduling available in ALEPH (`job_list`), you can set this service to

run at regular intervals. The only parameter the process requires is the library in which the Z325 is stored. For example:

```
W1 23:30 Y USR00 p_sdi_01 USR00
W1 23:30 Y USR00 p_sdi_01 USR00
```

The format for the e-mail message sent to patrons is stored in `/xxx01/form_lng/bor-sdi-00`.

Patrons can also create a new SDI request via the current search results list and previous searches.

When a new record is added to the database, the system checks if an SDI-ready record (Z324) should be created for it. Part of this mechanism is sending a Z105 message to the USM01 library where Z324 records are stored. The following setup should be defined in `/usm50/tab/tab_z105`:

```
!      1        2   3     4     5     6     7     8     9    10    11    12
!!!!!!!!!!!!!!!-!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!
UPDATE-SDI      8 USM01
```

12 Logical Bases

A logical base is a collection of records which share a common denominator, such as a particular library or sublibrary, material type, works published before 1800, and so on. Logical bases are organized in a manner that facilitates regular library activities, such as the retrieval of information. A logical base consists of a subset of the records available in an entire database (for example, USM01) but it is presented to the user as though it were a separate database. In effect, logical base act as pre-filters for a Find query. The Browse (headings) lists are also filtered, so that a list includes only headings that are relevant to the logical base.

Logical bases are defined in the `tab_base.lng` table in the `alephe/tab` directory. The following is an extract from `tab_base.lng`:

```
!          1           2            3             4     5     6     7   8
9
!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!-!!!!!!!!!>
USM01                    Demo Library          USM01 USM01       USM01
EDUC                     Education Library     USM01             USM01 N
wsl=ueduc
SERIALS                  Serials               USM01             USM01 Y
wfm=SE
MUSIC                    Music                 USM01             USM01 Y
wfm=MU
```

Column 8 defines whether or not a logical base should be created.

12.1 Logical Base Counter

The Z0102 table (logical base counter) is an option which can be used to make browsing from the Web OPAC more efficient when scanning logical bases which are less than 10% of the total database.

When a logical base is being browsed, the system uses the Z0102 table to determine whether or not to display the heading (Z01) without having to retrieve the documents attached to the heading and read them.

The Z0102 table is built for each heading and for each logical base. The table includes the filing text, the access sequence of the heading and a counter of the relevant documents attached to the heading. The pointer to the documents is stored in the Z02 (ACCDOC) table.

Conversion is not applicable; the Z0102 records are generated by running the Build Counters for Logical Bases (manage-32) service from the Services menu of the Cataloging GUI. You should always run this service after building the Headings index (manage-02).

When a heading is updated or when a new heading is created, the logical base counter is automatically updated by the UE_01 (specifically by the ue_01_z0102_index sub-process) daemon process. When a heading (Z01) is updated or created, the UE_01 process updates the Z0102 table.

It is recommended to set large bases to work with the regular Z01 mechanism instead of the Z0102 mechanism. This saves space and it increases the performance of the building process. The bases that work with the Z0102 must be set to Y(es) in column 8 of `/alephe/tab/tab_base.lng`.

You can view and analyze the setup of the bases by running UTIL H/1/10. Following is an example of this type of report which includes recommended and current setup of the Z0102 column (column 8) of the `tab_base.lng` table and a column indicating that there is a conflict between them:

```
Base            | # Docs       | Recommended | Current | Conflicts
--------------------+---------+-------------+----------+-------
BOOKS           |     2700     | No          | No       |
FAIRYTALES      |NEGATIVE BASE | No          | Yes      | *** Err ***
HISTORY         |      166     | Yes         | Yes      |
INTROD          |       43     | Yes         | Yes      |
HEALTH          |        3     | No          | Yes      | *** Err ***
NOTCLASSIFIED   |NEGATIVE BASE | No          | No       |
SERIALS         |      306     | Yes         | Yes      |
USM01           | FULL BASE    | No          | No       |
```

If a base has the `wrd=alldocuments not...` command in Column 9, or if the base has no documents, then the caption in the # Docs. column in the report will be: NEGATIVE BASE.

If a base has no command in Column 9, then the caption in the *# Docs.* column in the report will be: `FULL BASE`.

Here is an example. `tab_base.lng` has the following lines:

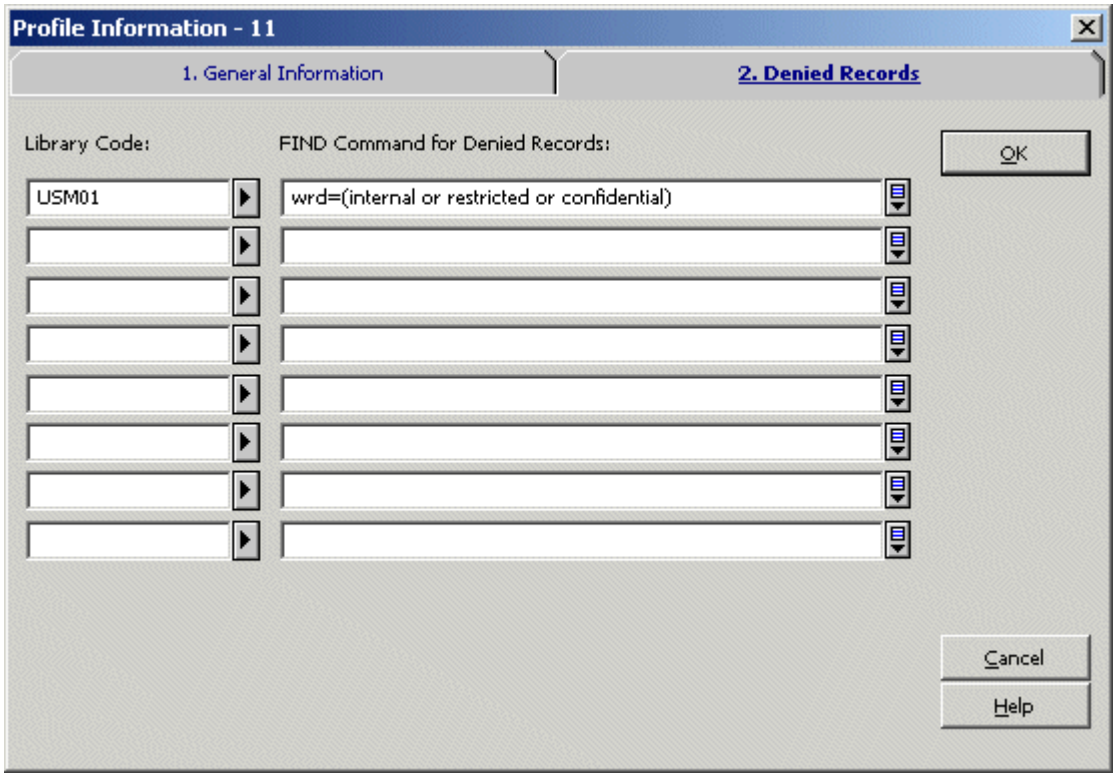
```
!         1          2           3            4     5     6     7   8
9
!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-
!-!!!!!!!!!>
NOT_ART                Not Art              USM01             USM01 Y
wrd=alldoc
uments not wrd=art
USM01                  Ex Libris University                   USM01 Y
```

The report will present the bases-related information as follows:

```
Base            | # Docs       | Recommended | Current | Conflicts
----------------+--------------+-------------+---------+----------
NOT_ART         |NEGATIVE BASE | No          | No
USM01           |FULL BASE     | No          | No
```

12.2 Denied Libraries

You can define the logical base (`tab_base.lng`) accessed by guest users. To prevent a guest user from entering a particular base, you can limit access by putting an asterisk (*) in col. 2 in `tab_base.lng` in the line where that base is defined. In addition, you can prevent a user from seeing records (in a logical base) by setting Denied Records in the Access rights tab of the Profile information window in the Circulation GUI. You do this by entering queries in the **FIND command for denied records** fields.



(See the Patron chapter in Circulation, Patron Profiles).

13 Patron ADM Library (Single/Multiple)

When patrons sign in to a Web OPAC that has multiple ADM libraries, the default library that is displayed when the patron clicks **My Library Card** is the ADM library that was chosen during sign-in. A patron can also choose *Shared Pool* from the library branch drop-down menu. In this case, the user's HOME-LIBRARY (primary sublibrary from his Global Patron Information – Z303 record) is taken to indicate to which ADM library the user belongs. If the user does not have a HOME-LIBRARY, the ADM library is determined by the first local patron information record (Z305). If no Z305 record exists for the user then TAB100-DEFAULT-BOR-LIBRARY in the BIB library is used. Here is an example:

```
DEFAULT-BOR-LIBRARY=USM50
```

In this case, the ADM environment of the user will be USM50.

If none of the abovementioned options supplies an ADM library, then error 9849 will be displayed, and the log-in action will fail.

On the *My Library Card* screen, the ADM libraries in which the patron is registered display, and he can click the link to switch to another library.

Information regarding loans and holds is managed and displayed separately for each ADM library. Information about the patron such as proxies, SDI profile, patrons. address and more, is managed by the USR00 library.

When the user places a request on an item in the OPAC, the system automatically checks the ADM library of the item, and chooses the most relevant Z305 record.

14 Synonyms Dictionary for Word Searching (Optional)

ALEPH includes a feature whereby groups of words can be set as synonyms (words having the same or nearly the same meaning as another in the language). In this case, a search for any one of the words automatically includes a retrieval of all the words that have been defined as synonyms in the group. For example, if karel, karl, carol, charles are set as synonyms, a FIND on any one of these words retrieves the records which contain any of the other words.

Synonym groups are created and maintained through the UTIL B menu of the bibliographic library, or through the Web Utilities. Web utilities are available from the Web Staff Menu; click Utilities, and choose the bibliographic library.

The following options are available for synonyms management:

- Add a group of synonyms
- Remove word(s) from a group of synonyms
- Unlink an entire group of synonyms
- View synonym groups

15 Setting up Save/Mail Encoding Options

You can set up the Save/Mail form invoked when Save/Mail is clicked to allow users to select different encoding options for records sent by e-mail.

To enable this functionality:

2. Remove the line CHARACTER_CONVERSION=NONE from all Web OPAC-related forms in the usm01/form_lng directory.

3. Set up character conversion for printouts for both Data and Form in mail-related HTML pages.

   Example (full-mail):

```
<td class=td2 id=small >
  <input type="radio"
         name="encoding"
         value="UTF_TO_WEB_MAIL_ASCI"
         $$1100-C"UTF_TO_WEB_MAIL_ASCI">ASCII (recommended /
default)
</td>
<td class=td2 id=small >
  <input type="radio"
         name="encoding"
         value="NONE"
         $$1100-C"NONE">Unicode / UTF-8 (non-Roman character
sets)
</td>
<td class=td2 id=small >
  <input type="radio"
         name="encoding"
         value="UTF_TO_WEB_MAIL"
         $$1100-C"UTF_TO_WEB_MAIL     ">ISO 8859-1 (Roman
character sets)
</td>
```

   You can define as many options as you like as long as they are also defined in the `$alephe_unicode/tab_character_conversion_line` configuration table.

   You can modify the encoding of the subject line of the e-mail message sent by the user by editing the UTF_TO_WEB_SUBJECT section in the `$alephe_unicode/tab_character_conversion_line` configuration table.

   Example:

```
UTF_TO_WEB_SUBJECT   ##### # line_utf2line_sb
unicode_to_8859_1
```

The above setting will save or send records by e-mail in the appropriate encoding but most mail programs will not show it without manual selection of the correct encoding by the recipient.

In order to avoid this, your ALEPH installation includes an example script, $alephe_tab/send_mail.example, that shows how this can also be solved.

The following section controls the encoding passed from the Web page:

```
switch ($4)
    case "NONE":
        echo "Content-Type: text/plain; charset=UTF-8" >>
$TMPDIR/send_mail.$$
        breaksw
    case "UTF_TO_WEB_MAIL_ASCI":
        echo "Content-Type: text/plain; charset=US-ASCII" >>
$TMPDIR/send_mail.$$
        breaksw
    case "UTF_TO_WEB_MAIL":
        echo "Content-Type: text/plain; charset=ISO-8859-1" >>
$TMPDIR/send_mail.$$
        breaksw
    case default:
        echo "Content-Type: text/plain; charset=UTF-8" >>
$TMPDIR/send_mail.$$
        breaksw
endsw
```

You can use this example script to create an appropriate *charset=* section inside the SMTP header. The same applies to the subject line though it can only hold one pre-defined value for charset.

If you are using `send_mail.example` as your mailing program, rename it as `$alephe_tab/send_mail` and make sure it has execution permission as follows:

```
chmod +x $alephe_tab/send_mail
```

Note

Any change in the `tab_character_conversion_line` conversion table relating to the WEB sections must also be reflected in this script.

16 Save/Mail on Multi-Apache Topology

When using a multi-Apache topology, clicking **save/mail** displays a File Download dialog box that allows the user to save the file.

To support the save/mail feature in such topology, the following setup is required:

1. Set the preview_required attribute to `Y` in all forms that use the Save/Mail option by adding the following line:

   ```
   <input type=hidden name=preview_required value="Y">
   ```

   After the line:

   ```
   <input type=hidden name=func value=myshelf-mail>
   ```

   In the following files:

   ```
   ./alephe/www_f_lng/myshelf-mail
   ./alephe/www_f_lng/myshelf-course
   ./alephe/www_f_lng/user-info-mail
   ./alephe/www_f_lng/short-mail
   ./alephe/www_f_lng/full-mail
   ```

2. The number of docs that can be downloaded is limited at user profile level by the variable Z61-MAX-SAVE-MAIL-FILES. Define for all profiles (Z61) an appropriate maximum limit of the number of save/mail files for all profiles (the default is 999999). It is recommended not to exceed the value of 20,000 for this limit, thus enabling a reasonable response time to the end user.

For more information, refer to the *Aleph Topologies* document.

Note:
With this topology, the save/mail functionality is not based on the search result, so an additional search is performed. Sort is not applied.

17 Local Own Check Procedure

When an ACQ request form is submitted, the system activates the “locate similar record in local database” function for blank ACQ requests and for titles found in an external base. This function is inactive for titles found in a local base. The base that is used for this locate action is the BIB library where the ACQ request's BIB document is about to be created.

The base used for this locate action is defined in the "www_base" environment variable of aleph_start.
The locate query is taken from ./xxx01/tab/tab_locate.

The ACQ request BIB document is created in the BIB library to which the patron’s ADM library is attached.

If an ACQ request is created based on a BIB record that is not related to the ADM library of the patron's home library, a new BIB record is created. It is possible to set a value in the OWN field of the new BIB record that indicates its connection to the ADM library. To do this, populate the ACQ-REQUEST-OWN variable in tab100 of the ADM library with a string of up to 10 characters. For example

./[ADM_library]/tab/tab100

ACQ-REQUEST-OWN=USM50

18 ACQ Request for Library with more than One BIB Library

This chapter is relevant only for library which its ADM library is set to work with more than one BIB library.

For example:

./alephe/tab/library_relation is set as in the following example:

BIB USM50 XXX01 YYY01

Such setup; relates the ADM library (USM50) to two BIB libraries (XXX01 and YYY01).
When an ACQ request is created via the Web OPAC, there is an option to configure the OPAC ACQ Confirmation page so that the patron can select the BIB library in which the BIB record will be created.
The locate procedure, which is activated during submitting ACQ Request, will be performed at the selected BIB library.
To support this, update the following OPAC HTML pages:

./alephe/www_f_eng/acq-request
./alephe/www_f_eng/acq-request-doc-tail
./alephe/www_f_eng/acq-request-confirm
./alephe/www_f_eng/acq-request-doc-tail-conf

For two BIB libraries, XXX01 and YYY01, update the above HTML pages as in the following example:

```
<tr>
  <td class=td2>BIB Library<br></td>
  <td class=td2> </td>
  <td class=td2>
    <select name="bib_lib">
      <option value="XXX01" $$3700-S"XXX01">BIB library XXX01</option>
      <option value="YYY01" $$3700-S"YYY01">BIB Library YYY01</option>
    </select>
  </td>
 </tr>
```

Note that if the HTML pages are not updated, the system will use the first BIB library as set in library_relation table (XXX01 in above example).

19 Applying ADM Filter in OPAC-ACQ Requests

The ACQ request flow appends the order that is created as a result of an ACQ request on a BIB and an ADM record.

Usually, when the system points to a BIB record for attaching an ACQ request, the ADM library that owns the title is not taken into account. In certain library structures, this might causes the system to attach an ACQ request belonging to a certain library to a BIB record owned by a different library. Such a situation can be avoided by applying a filter routine within the ACQ request flow: the ADM library filter.

There is an option to apply the ADM library filter when the system points to a BIB record for attaching an ACQ request.

The ADM library ensures that the ACQ request relies on a BIB that has an ADM record belonging to the patron's home-library. This BIB is used for the ACQ request only if the BIB has an ADM record that is related to the library that matches the patron's home library. Otherwise, a new BIB records is created.

To use the ADM library filter mechanism, the library must configure a hidden parameter in the ACQ Request confirmation HTML pages.

This parameter should be set for libraries whose structure requires activating the ADM filter, for example, a multi BIB/ADM library architecture in which there are several BIB libraries and each of the ADM libraries is connected to more than a single BIB library.

19.1 Activating the ADM Library Filter

Once the candidate BIB record is detected by the locate action, the system conducts an additional check to see if the ADM library filter action should be applied.

In order to turn-on the ADM library filter, set the following hidden parameter with:

```
<input type=hidden name="APPLY_ADM_FILTER" value="Y">
```

The value of the parameter can be:

- Y – Activate the ADM library filter

or

- N – Default value. Do not activate the ADM library filter (append the ACQ request to the candidate BIB record).

The parameter should be set in the following OPAC HTML pages:

- ./alephe/www_f_eng/acq-request
- ./alephe/www_f_eng/acq-request-doc-tail
- ./alephe/www_f_eng/acq-request-confirm
- ./alephe/www_f_eng/acq-request-doc-tail-conf

If the new hidden parameter is not set or set to "N", the located BIB record is used for ACQ requests without applying the ADM filter to the detected BIB record.

The ADM library filter conducts the following:

- The system detects the BIB library of the located BIB record. This BIB record is considered as a ‘candidate BIB record’.
- ADM library filter setup check - The system consults the parameter set in the HTML page to see if the ADM Library filter should be activated.
- If the ADM library filter parameter is turned on in the HTML page, the ADM filter is applied so that the ACQ request is appended on a BIB record that is related to the patron’s home library. If the located BIB record dose not relate to an ADM record that belongs to the patron’s ADM library, a new BIB record is created to which the ACQ request is attached.

19.2 ADM Library Filter Routine

The ADM Library filter takes place only when the hidden parameter ‘APPLY_ADM_FILTER’ is set to “Y”.

The routine works as follows:

The ACQ request flow traces a candidate BIB record for assigning the ACQ request and the ADM library filter checks to see if the BIB record has an ADM record to which the patron belongs.

- If yes – the BIB record is used for attaching an ACQ request.
- If no – a new BIB record is created by duplicating the traced non-relevant BIB record. The newly created BIB record contains bibliographic information similar to the one from which it is driven. Fix, expand fields, and OWN values are stripped. A new ADM record is attached to the new BIB record, to which the ACQ request is appended.

If the candidate BIB has no attached ADM record at all, a new BIB record is created (rather than using the existing BIB record with no ADM record).

20 Log File Expansion

The `www_server` log file can be expanded to include information on the status of the checks that were performed on a hold request. This means that the exact number of errors caused by each check configured in `tab_hold_request` can be logged. This is done by setting the environment variable DEBUG_CHECK_HOLD to Y prior to `www_server` initialization in the following manner:

```
setenv DEBUG_CHECK_HOLD Y
```

If this configuration is set, the log file will include information such as :

```
check_hold_request "CIRC      " :

After check_hold_request_a BUF-ERROR-NO-ERRORS=01

After check_hold_request_b BUF-ERROR-NO-ERRORS=02

After check_hold_request_c BUF-ERROR-NO-ERRORS=02
```

21 Setting Up BIB Request Policy

21.1 Group Definition and Availability

The following two steps are performed when a BIB request is initiated:

- Any ADM library, which has items that are connected to the BIB record, is checked with the tests defined in the HL-GRP-PRE section of the tab_hold_request configuration table. If the ADM library does not pass the tests, it is ignored and its items are not considered for placing the hold request.

- The items that are connected to the requested title are split into groups of *like items*, so that each group can be selected separately. A group of *like items* is defined as a set of items that fulfill the following conditions:

    - They belong to the same ADM library.
    - They are similar in their enumerations (item enumeration level 1, 2 and 3) and in their chronological information (item chron level 1, 2, and 3).
    - They are similar in their sublibrary, item status, item process status and collection information, depending on the `tab100` setup of
        - SUB-LIBRARY-DIVISION
        - HOLD-REQUEST-ITM-STATUS
        - HOLD-REQ-PROCESS-STATUS
        - HOLD-REQUEST-COLLECTION

Items in a group are considered requestable only if they pass the checks that are defined in `tab_hold_request` under the HL-GRP section. Therefore, if all of a group's items do not pass the HL-GRP checks, the group is considered not requestable and it will be shown in the groups list with the *Items* link instead of the *Request* link.

Note that items that are not displayable in the OPAC according to the `tab15.lng` setup are ignored when the *like item* groups are built. Also note that item request exactness (according to column 11 of `tab15.lng`) is also ignored when requests are created from the BIB level.

21.2 Pickup Location List

The available pickup locations are built according to the setup of `tab37`. The list will include all of the allowed pickup locations of all of the items in the group. You can use Column 5 (Availability of item) to filter out a sublibrary from the drop-down list of pickup locations. If Column 5 is set to Y, the line only applies to available items (that is, on the shelf). If Column 5 is set to N, the pickup location will display only if the item is not available (that is, the item is on loan or on the hold shelf).

For example, consider the following `tab37` setup:

```
MUS   ## ## ## Y MUS  HIL  WID
MUS   ## ## ## N MUS  HIL  WID
WID   ## ## ## Y HIL  MED
WID   ## ## ## N WID  HIL   MED
```

If the group has items that belong to sublibrary MUS as well as items that belong to sublibrary WID, the pickup list displays all locations, meaning: MUS,HIL,WID and MED.

However, if a WID item is available (according to the availability routine that is defined in the `tab100` variable AVAILABILITY-ROUTINE) the WID sublibrary will be removed from the pickup list. This is because WID items allow WID to be a pickup location only if a WID item is not available. The pickup list will therefore be MUS, HIL, MED. On the hold request form, a message will indicate that WID has been removed from the pickup list.

In addition, the selected pickup location sets which items are considered requested. In the example above, if MUS was selected as a pickup location, the WID item is not considered requested even though it is a *like item*. This is because MUS is not a valid pickup location for WID items.

21.3 Item Level Request

The `tab100` variable, HOLD-BY-ITEM-GROUP, is used to set the system behavior when items are requested from the item level. If the variable is set to Y, then the list of pickup locations will be built according to the `tab37` pickup locations of all items that are *like copies* of the requested item. If the variable is set to N, then the list of pickup locations is built only according to the `tab37` line of the requested item.

22 Setting Up My e-Shelf

After performing a search, patrons or guest users can select items to add to their e-Shelf (My e-Shelf) from the results list or from the full view of a record. The contents of the e-Shelf can than be sent by e-mail, printed or saved.

To delete from the database folders and records (Z129 records) that have been generated by guest users, run UTIL/A/18.

The e-Shelf is made up of the following files:

```
myyshelf-clear-basket
myshelf-course
myshelf-delete-folder
myshelf-folder-list
myshelf-folder-list-body-no-doc
myshelf-folder-list-left-body-1
myshelf-folder-list-left-body-2
```

```
myshelf-folder-list-left-body-3
myshelf-folder-list-left-body-no-folder
myshelf-folder-list-left-head
myshelf-folder-list-left-tail
myshelf-folder-list-right-body-1
myshelf-folder-list-right-head
myshelf-folder-list-right-tail
myshelf-full-head
myshelf-full-tail
myshelf-mail
myshelf-no-doc-basket
myshelf-no-fol-for-delete
myshelf-note-full
myshelf-note-short
myshelf-refresh-page
myshelf-save-as-fol
myshelf-short-body
myshelf-short-folder-list
myshelf-short-head
myshelf-short-head-2
myshelf-short-head-2-no-course
myshelf-short-select-include
```

The `www_tab_myshelf_short` table (located in the tab directory of the USR library.), defines which information will be displayed in the columns of the short display in the My e-Shelf function.

The order of the columns defined in this table determines the order of the display, in conjunction with the `myshelf-short-head`, `myshelf-short-head-2`, `myshelf-short-head-2-no-course` and `myshelf-short-body` HTML files in the `www_f_eng` directory.

22.1 My e-Shelf and Course Reading Addresses

There is an additional option for saving the e-Shelf contents onto the server (usually in `alephe_scratch` unless defined otherwise in `aleph_start` by using the `SAVE_FILE_TMPDIR` parameter) and notifying the Course Reading administration. This option is contained in the `myshelf-course` HTML file. The patron must have course permission in his profile to be able to see this option.

You can choose to specify the e-mail address of the Course Reading administration inside this HTML file. You can also use this option to build a drop-down menu of e-mail addresses where there are several course reading administrations to choose from. For example, if the following definition is present in the SEND_MAIL setting in the `myshelf-course` HTML file:

```
<input type=checkbox name="SEND_MAIL" $$0200-C"on" value="
mdewey@exlibris-usa.com ">
```

the system sends the e-mail notification message to mdewey@exlibris-usa.com:

Select this option to inform the Course Reading administration that you want these records added.

If the value in the `myshelf-course` HTML file SEND_MAIL setting is not an e-mail address, the system chooses the e-mail address defined in the `www_course_email_address` environment variable present in `www_server.conf`. For example, the following setting sends the e-mail notification message to the value defined in `www_course_email_address`:

```
<input type=checkbox name="SEND_MAIL"> Check here in order to inform
the Course Reading admini...
```

22.2 Setting the Maximum Number of Records in a Folder

Each folder (including the BASKET folder) can contain a maximum of 500 records.

You can define the maximum number of records allowed in a folder by setting the `www_myeshelf_limit` environment variable in the `www_server.conf` file.

If the variable is not defined, the default for the maximum records in each folder is 500 records.

If the number of records to be stored in a folder exceeds the maximum number as defined in the `www_myeshelf_limit` variable, the user is prompted with the following message:

There is no limit on the number of folders that a patron or guest can create and manage.

My e-Shelf and Union View

An availability column can be added to ‘My e-shelf’ screen of a Union View OPAC Interface. The column displays an indication whether the contributing library to which the user belongs already has the record.

The column must be added to www_tab_myshelf_short.lng table, for example:

```
! 2        3                  4        5  6   7     8 9    10        11    12     13
!--!-!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!-!!-!!-!!!!-!-!-!!!!!!!!!!-!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!
9 L  Availability                                  L                ##  AVAIL
```

In addition, the e-shelf HTML pages must be configured to include this column, just like any column that is setup in www_tab_myshelf_short.lng.

- myshelf-short-body
- myshelf-short-head-2
- myshelf-short-head-2-no-course

If the logged in user’s library already has a record that belongs to the set of the record that is in the e-shelf, a message such as “Record already exists in your library” will be displayed in this column

The message can be configured by the library.

The system must be able to recognize to which contributing library the logged in user belongs, based on the user’s z66_user_own_check field. The message will be

displayed if the SID $b of one of the merged records matches the content of z66_user_own_check.

Note that the following line must exist in tab_expand of the union library:

WEB-BRIEF expand_doc_merge_unio

22.3 Setting Up the Note Column in the Records Which Were Not Requested Table

To add a note that specifies the reason for a rejected hold request, add the following to ./alephe/www_f_lng/group-request-exec-head:

After:

<th class=text3>Title</th>

Add:

<th class=text3>Note</th>

In /alephe/www_f_lng/group-request-exec-body

After:

<td class="td2" nowrap>$0200</td>

Add:

<td class="td2" nowrap>$0300</td>

22.4 Setting Up the Submitted a Hold Request Column in My e-Shelf

To add a column that indicates if a record has a hold request, add the following to /alephe/www_f_lng/myshelf-short-body:

After:

<td class=td1 id=centered width="1%" valign=top>$0400</td>

Add:

<td class=td1 id=centered width="1%" valign=top>$1600</td>

In /alephe/www_f_lng/myshelf-short-head-2-no-course,

After:

Documents in My e-Shelf

Add:

<img src="&icon_path/f-check-mark-1.gif"

border=0 alt="Submitted a hold request">

submitted a hold request

After:

<th class="text3">$0300</th>

Add:

<th class="text3">$1600</th>

23 Search Keyword in Headings

A keyword search in the heading list can be implemented in the Aleph search mechanism. This chapter describes the keyword search in heading list functionality and its setup.
The Keyword search can be activated by the user via:

- WEB OPAC - Browse (./www_f_lng/scan-include-2)
- WEB OPAC – Search/Browse (./www_f_lng/find-e-head)
- GUI-Search-Browse

In order to activate keyword search in heading list functionality, the Z01x mechanism must be activated. (See details in the Setting-Up Keyword Search in headings section).

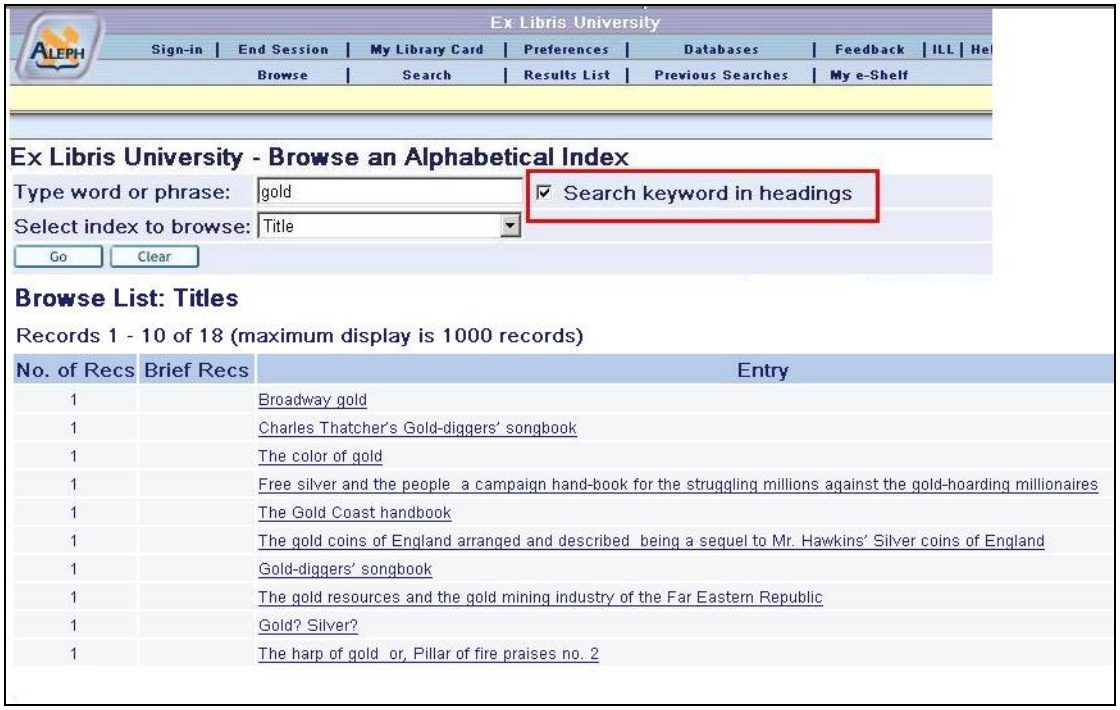
Keyword search in heading list functionality cannot be applied for external libraries (EXT libraries).

23.1 Keyword Search in Heading List Functionality Via Web OPAC-Browse

Upon implementing the keyword search in heading list functionality (see details in section 23.5 Setting Up Keyword Search in Heading List Functionality), the headings list can be searched for keywords that is contained within the heading text.

The user submits a browse query, selects an index to browse, and selects the **Search Keyword in Headings** checkbox.

In the Browse List, the system displays all headings which include the browse query.



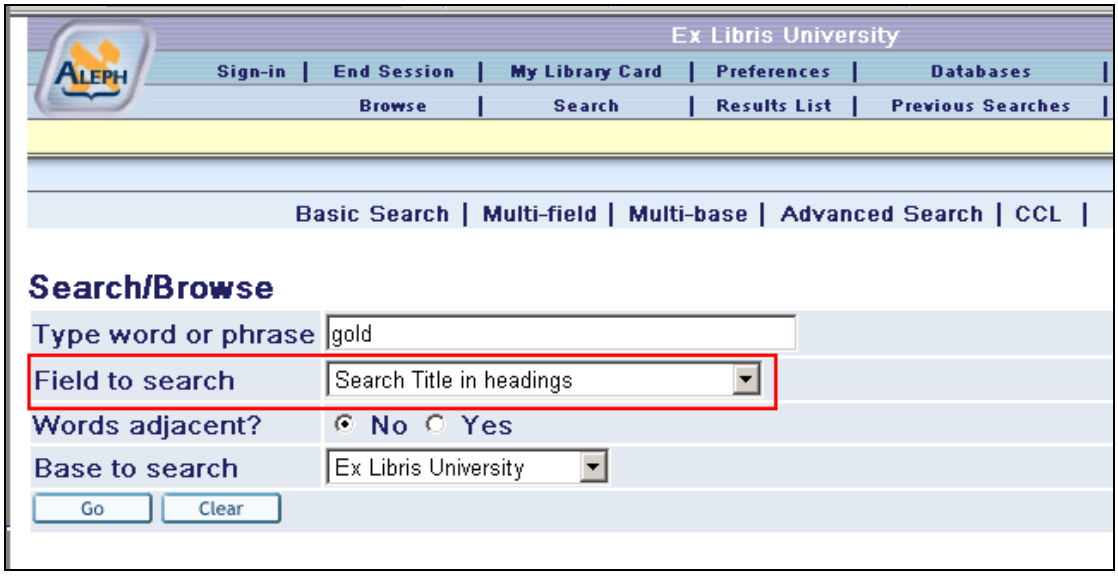
The above example shows a browse list of headings that include the word “gold” in their title index.

User may also type “gold*”. In this case, the browse list displays headings that include words that begin with “gold”, for example, “Broadway golden show”, “The Goldenberg's book”, “Silver and gold”, etc.

23.2 Search Keyword in Heading via Web OPAC-Search/Browse

For systems which are set to apply the Search/Browse mechanism (fine-e-head, see details in 23.5 Setting Up Keyword Search in Heading List Functionality); the Search Keyword in Headings can be applied by selecting heading search in the **Field to search** dialog box.

The following example shows a heading keyword search for the word “gold”.



The search results are the same as described above in the section "Search Keyword in Headings via WEB OPAC-Browse".

See setup details in 23.5 Setting Up Keyword Search in Heading List Functionality section.

23.3 Search Keyword in Heading via GUI-Search-Browse

The GUI-Search-browse screen includes a checkbox **Search Keyword in Headings** that indicate to the system to perform a keyword search in the heading list.

The search results are the same as described above in the section 23.2 Search Keyword in Heading via Web OPAC-Search/Browse.

23.4 Search Rules and Logical Operators for Keyword Search in Headings

The following section explains how the keyword search in heading list functionality works for various user's input.

Keyword search in heading list functionality supports "and", "or", and "not" Boolean operators.

"And" is assumed if no operator is present.

Operators can be typed in English or expressed by using any of the following conventions:

- AND = + (plus sign)
- AND = & (ampersand)
- OR = | (pipe sign)
- NOT = ~ (tilde)
- - (minus sign)

Note that the phrases OR, AND, and NOT are not supported as a text string but only as Boolean search operators.

When entering keywords, the user can use truncation, masking, and logical operators, as explained below.

23.4.1 Truncation

The user can type a portion of a word and add an asterisk (*)at the end. This indicates that the user wants all words that begin with that portion of the word.

Note: A truncation symbol cannot be used at the beginning or a middle of a word. The portion of the word must contain at least three characters before the truncation symbol.

For example:

```
Chloro*
```

retrieves all records containing words beginning with *chloro* in an indexed field.

23.4.2 Logical (Boolean) Operators

The user can narrow or broaden a search by using the Boolean operators: &,+, |, ~, AND, OR and NOT. The user can use these operators in a search string.

For example:

```
government and labor
```

retrieves all records containing *government* and *labor*.

```
government and lab*
```

Retrieves all records containing *government* and (*labor* or *labour* or *laboratory or* any other word beginning with *lab*).

```
government or cabinet
```

retrieves either *government* or *cabinet*.

```
government not parliament
```

retrieves all records containing the word *government* but not including the word *parliament*.

23.4.3 Parentheses

Parentheses must be used in a retrieval request, when required for correct query syntax.

For example:

```
(strategy or tactics) and (dunnigan and churchill)
```

23.4.4 Phrase Searches

In order to search for an exact phrase, the user can type in the phrase with or without double quotes.

For example, to find the exact phrase **Olympic Games**, enter: *"olympic games"* or *olympic games*.
Each of those queries retrieves all records containing the exact phrase *olympic games.*

To retrieve headings that include these words in any order not necessarily in adjacency, use the AND expression (for example, *olympic and games).* To search for a phrase which include “and”, “or” or other operators as regular words, surround the whole phrase with double quotes.

23.4.5 Using the "-" Symbol

The – (minus sign) can be used to lower the score of documents that contain unwanted terms. The ,minus sign is useful when you want to search for documents that contain one query term but want the presence of a second term to cause a document to be ranked lower.

term1-term2 returns documents that contain term 1. The system calculates the score by subtracting the score of term 2 from the score of term 1. Only documents with positive scores are displayed.

Example:

Suppose a query for the term `cars` returns documents about *Ford cars*. You can lower the scoring of the *Ford* documents by using the following expression:

```
cars – Ford
```

This expression returns documents that contain the term `cars` and possibly `Ford`. However, the score for a returned document is the score of `cars` minus the score of `Ford`.

23.5 Setting Up Keyword Search in Heading List Functionality

In order to enable keyword search in heading list functionality, Z01x must be created for Z01- Access Heading records. Enabling the keyword search in headings functionality consumes extra Oracle database space. The extra database space needed is about 70% of the Z01 table size.

Note that indexing Z01x requires an oracle user with “JOB CREATE” enabled.

The following are the required steps for implementing Z01x records:

Step 1:

Define Z01x in the `file_list` of the BIB Library (XXX01) as in the following example:

```
Sample of . /usm01/file_list:
-1-|--2---------|------3-----|--------4------------|---5--|---6--→
TAB z01x              2M              0K              ts0
IND z01x_id           1M              0K              ts1
INX z01x_id1          1M              0K              ts1
```

After Z01x is defined in `file_list`, create the Z01x table by running util/a/17/1 in the BIB library.

Step 2:

In order to create Z01x for current Z01 heading records, in BIB library run the script `p_fill_z01x`.

The following is an example of how to run the script in the USM01 library:

```
>ap

> csh -f p_fill_z01x USM01
```

This script performs the following actions:

- Lock library
- Drop indexes on z01x (if exist)
- For each Z01 record, create z01x records
- Create indexes for z01x
- Unlock library

At this stage Z01x records are created and indexed and keyword search in heading is enabled via OPAC and GUI.

Note that after updating/loading headings using a bulk mode (batch services), the Z01x script should be run.

This means that after submitting the below batch services, the p_fill_z01x script should be run so that z01X records will be updated/created and indexed.

- Update Headings Index (manage-02)
- Pre-Enrich Bibliographic Headings Based on the Authority Database (manage-102)
- Alphabetize Long Headings (manage-17)

Step 3

To enable Search By Keyword in headings, add the following lines to alephe/unicode tab_character_conversion_line:

```
!!!!!!!!!!!!!!!!!!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!-!
WORD-FIX         ##### # line_utf2line_utf       unicode_to_word_gen
```

```
HEADING-KEYWORD      ##### # line_utf2line_utf    unicode_to_word_gen
```

Note that the HEADING-KEYWORD definition must be the same as the WORD-FIX definition unless the naco_diacritics table (FILING-KEY-10) is not used for normalized text in headings. In this case the HEADING-KEYWORD definition must correspond to the character conversion procedure used for normalized text in headings.

Step 4:

To ensure that the Z01x mechanism is optimized, the optimize procedure should be set in `job_list`.

Following is a sample of setting Z01x optimizes procedure for USM01 BIB library:

```
Sample of ./alephe/tab/job_list:


W0 00:00:00 Y              USM01 optimize_z01x_context_index    USM01
```

This procedure removes duplicate entries from the "z01x_id1" index in order to reduce its size and help the Z01x work faster.

It works similar to UTIL A/17/17/6 "Optimize Context Index".

Step 5:
In the Web OPAC-Browse HTML page, add the checkbox for **Search keyword in headings**:

The checkbox should be located next to the **Type word or phrase** edit field:

```
Relevant lines in ./alephe/www_f_lng/scan-include-2:

  <td class=td2 valign=top>
        <input size=29
               name="scan_start"
               maxlength=100
               value="$0100">
        <input type="checkbox"
               name="scan_include"
               value= "Y"
               $$0800-C"Y">
        Search keyword in headings
  </td>
```

Step 6 - Relevant only for libraries that applied the Web OPAC-Search/Browse option (find-e-head)

The **Field to Search** drop-down list in `./www_f_lng/find-e-head` should be updated with new lines that point to the Search in Keyword functionality.

The list of **Field to Search** is defined in `./www_f_lng/find-e-code-include`.

Add the desired index to `find-e-code-include`. The string "`FACC_`" must precede the index code as in the following example.

```
For example, to set keyword search for title (TIT index) in headings
list, set the following line in ./www_f_eng/find-e-code-include :

       <option  value=FACC_TIT    $$FXX-S"FACC_TIT">Search  Title  in
headings</option>
```

24 Web OPAC and OpenSearch™ Compatibility

If you want to make your site available for searches using the A9 search interface (the search is initiated from http://a9.com/) you need to register your site in A9 and also prepare a number of configuration files required by the OpenSearch protocol. When registering your ALEPH database at the A9 site, you have to define a number of keywords that describe the areas of knowledge covered in your database, institutional affiliations - or any other keywords that would allow an end user to make a decision as to whether include your site in the list of sites s/he wants to search simultaneously.

OpenSearch requires the following mandatory information:

- The search query syntax to be used when searching your database
- The search query itself.

You should provide some of this information when you register at the A9 site, and there are also three configuration files that have to be created and stored on the machine hosting your ALEPH database. These files should reside under `alephe/jboss_conf` directory. These files include:

- Query syntax rules and keywords as mentioned above
- Parameters for the JBoss to construct the X-service
- The XSL file with definitions controlling the output display at A9.

The `QuerySyntax.xml` file includes XML tags required by A9 and information about the server to be searched. The full syntax is described in www.A9.com. A valid example might look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
<ShortName>ALEPH Search</ShortName>
<LongName>ALEPH Search</LongName>
<Developer>ALEPH Development Team</Developer>
<Description>Search ALEPH Catalog</Description>
<Tags>exlibris aleph catalog library</Tags>
<Contact>contact.us@my.library</Contact>
<Url type="application/rss+xml"
template="http://alephex.exlibris.co.il:1891/AlephWebApi/OpenSearch?r
equest={searchTerms}&amp;start_page={startPage?}"/>
</OpenSearchDescription>
```

The following URL points to the above XML, and it is also the one to be registered at A9:

```
http://alephex.exlibris.co.il:1891/AlephWebApi/OpenSearch?action=QuerySyntax
```

Note that the port number specified in the template should correspond to the number defined for the JBoss.

The `QuerySyntax.xml.tml` template file is available in the `alephe/jboss_conf` directory.

The `present2rss.xsl` file defines the fields of the data that will be presented once the search results are fetched and defines the display features - how many hits will be displayed per page, which fields of the resulting records will be clickable, and so on. The present2rss.xsl.tml template file is available in the `alephe/jboss_conf/xsl` directory. The current setting turns the title section of the `present2rss.xsl` file into a clickable link to the document.

The `main.properties` file resides under the `alephe/jboss_conf` directory. This file defines the relation between the JBoss and the ALEPH X-server. It also defines the search base that is searched when a query is submitted in A9. Any physical or logical base defined in `alephe/tab/tab_base.lng` can be specified here. The `main.properties.tml` template file is available in the `alephe/jboss_conf` directory.

25 Web OPAC Security

25.1 XSS Validation

The Web OPAC contains free text fields that allow the insertion of HTML text that can modify the page displayed. Without proper security care, the site may be abused. An attacker can create a program that pretends to be a real browser and submit JavaScript that may open security holes.

If you have encountered this problem, activate the HTML validation variable (XSS-VALIDATION).

Note that even without the activation of this flag, the malicious insertion of HTML scripts does not corrupt the database.

25.2 CSRF Valuation

CSRF (Cross Site Request Forgery) is a type of internet identity theft where a malicious Web site attempts to access information from a second Web site open on a user's computer. If the user has logged on to the second Web site, the malicious Web site may be able to access sensitive information.

In order to prevent this violation from occurring in Aleph, when a user logs on to the Aleph Web OPAC, a token ID is created and saved in the Aleph database as part of the session entity. This token ID is validated by the OPAC before any actions can be performed on the database. Any actions without the token ID are blocked.

In order to apply the token validation, set $alephe_tab/tab100:
TOKEN-VALIDATION = Y

In order to support the token parameter, configure the $9900 place holder as hidden in the following HTML files:

- item-photo-request-head
- bor-include-4
- collect-data-js
- bor-update
- bor-update-without-date
- new-ill-book-request
- new-ill-journal-request
- item-hold-request-head
- booking-page-form-head
- booking-req-conf-2
- title-request-form-head
- myshelf-note-short
- myshelf-note-full
- acq-request-doc-head
- acq-request-doc-head-conf
- acq-request
- new-ill-r-head
- new-ill-req-conf-tail
- group-request-head
- myshelf-short-head-2
- myshelf-short-head-2-no-course
- new-record
- bor-note-new
- item-hold-request-details
- item-photo-request-details

In the./alephe/www_f_lng/bor_note_new file, change:

<a href="&server_f?func=bor-note-new&clear_level=2">

to:

<a href="&server_f?func=bor-note-new&clear_level=2&token_id=$9900">

25.3 Session Fixation

Authenticating a user without invalidating any existing session identifier gives an attacker the opportunity to steal authenticated sessions: An attacker is able to force a known session ID on a user so that after the user authenticates his identity, the attacker has access to the authenticated session.

In a typical example of session fixation vulnerabilities exploitation, an attacker creates a new session on a Web application and records the associated session identifier. The attacker then causes the victim to authenticate his identity on the server using the

same session identifier, giving the attacker access to the user's account through the active session.

In order to prevent this violation from occurring in Aleph, the session ID is changed when the user logs on to Aleph.

To have the session ID change at log on, set the following in alephe_tab_tab100:

SESSION-FIXATION = Y

25.4 Apache Security Vulnerability Patch

The Apache server contains a security vulnerability. To avoid the security vulnerability:

1. Add the following line to the./alephe/apache/conf/httpd.conf file, just before the Section 2 line:

```
RequestHeader unset Range
```

2. Restart the Apache server.

3. Restart the Web server.

26 Web OPAC Accessibility

Web accessibility allows people with disabilities to use the World Wide Web.

This is important because the World Wide Web is an important resource in many aspects of life, such as education, employment, government, commerce, and health care. It is essential that the Web be accessible in order to provide equal access and equal opportunity to people with disabilities. Having the Web accessible can also help people with disabilities more actively participate in society.

This chapter covers a set of accessibility recommendations. Complying with them assures that the HTML code complies with international standards of code writing, accessibility, and usability to make your content more accessible to a wider range of people including users with disabilities.

The goal of this chapter is to enhance the Aleph Web OPAC to comply with the W3C Web Content Accessibility Guidelines 1.0, level A Section 508 of the Rehabilitation Act (29 U.S.C. 794d) – the leading international accessibility and industry standard. (For exceptions, see Exceptions on page 110).

26.1 Adding Descriptions to Images

Use the alt tag to provide a brief description of the content and function of images, so that a screen reader can describe the image. It should be as short as possible - 1024 characters maximum. If the image is used for cosmetic purposes (a blank space, dot, etc.) leave the alt attribute empty.

For example, change:

```
<img          style="display:none;" id="google_book"/ >
```

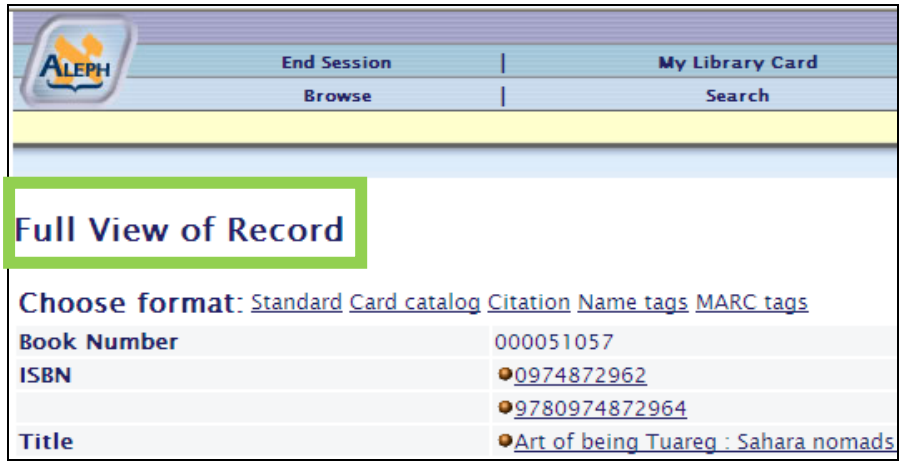
To:

```
<img           style="display:none;" alt="Book Front Cover"
id="google_book"/ >
```

26.2 Defining Titles with the Heading Attribute

Define the main title of a page with the <h1> attribute.

For example:



Change:

```
<div class="title">
  Full View of Record
</div>
```

to:

```
<div><h1>
  Full View of Record
</h1></div>
```

In order to keep the title display consistent with the previous title view, define the <h1> attribute in the Web OPAC CSS file. See the following example:

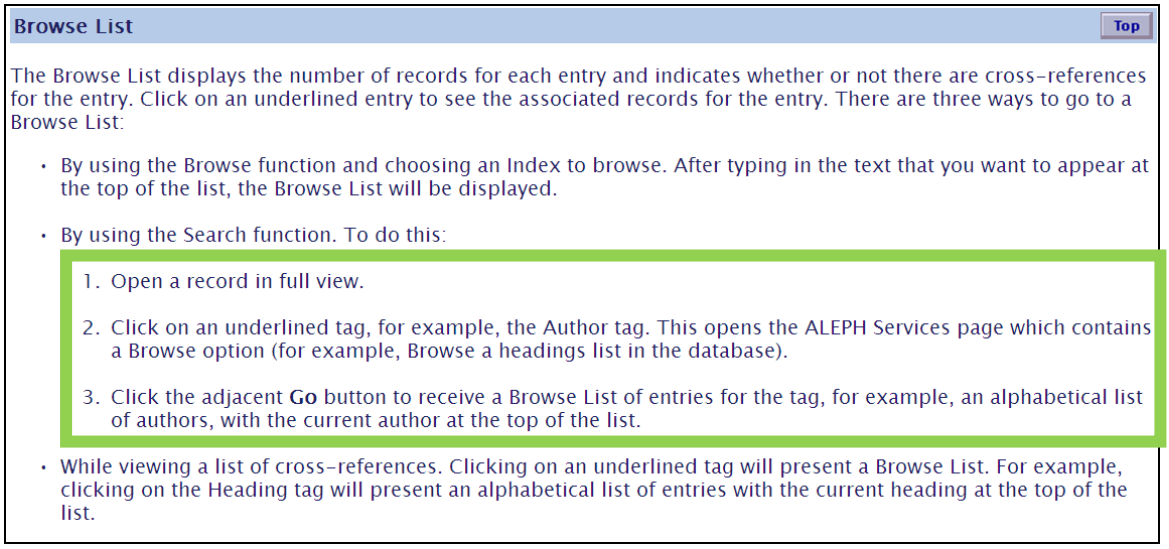
```
h1
{
font-size:120%;
font-family:'Lucida Sans Unicode','Arial Unicode MS',Arial,
Helvetica, serif;
font-weight:bold;
color:#212063;
}
```

26.3 Bulleted and Numbered Lists

HTML offers several mechanisms for specifying lists of information. All lists must contain one or more list elements such as bullets and numbers.

Update all lists in the Web OPAC to have open and close <ol> or <ul> tags and each item of the list to have an <li> tag.

For example:



The following is an example of the code:

```
<ol>
<li>Open a record in full view.
<br><br>
<li>Click on an underlined tag, for example, the Author tag. This opens the ALEPH Services page which contains a Browse option (for example, Browse a headings list in the database).
<br><br>
<li>Click the adjacent <b>Go</b> button to receive a Browse List of entries for the tag,for example, an alphabetical list of authors, with the current author at the top of the list.
</ol>
```

26.4 Indicating Table Headers

In order for a data table to be accessible, indicate row and column headers with the <th> tag, so that screen readers can navigate through data table one cell at a time. Do this only for data tables and not for visual layout tables.

For example:

No.		Author	Description
1	☐	Laufer, Paul.	Au pays du Christ /
2	☐	Galron, Nurit.	Nurit Galron
3	☐		She lun p'ing lun hsüan.
4	☐	Viets, Henry R. (Henry Rouse), 1890 –	A brief history of medicine in Massachusetts /
5	☐	Danielis ludus	The play of Daniel :

The code is:

```
  <th class="text3">No.</th>
  <th class=text3><br></th>
  <th class="text3">Author</th>
```

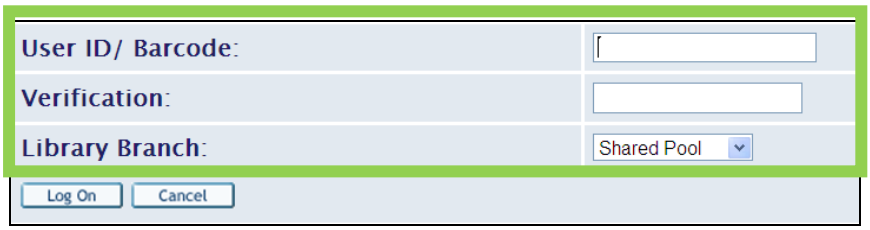
```
  <th class="text3">Description</th>
```

26.5 Assigning Labels to Input Form Elements

HTML forms are used to pass data to a server. A form can contain input elements such as: text fields, checkboxes, radio-buttons, and submit buttons. Assign all input elements a label with the <label> tag.

26.5.1 Text Fields

The following is an example of a text field:



Change the code from:

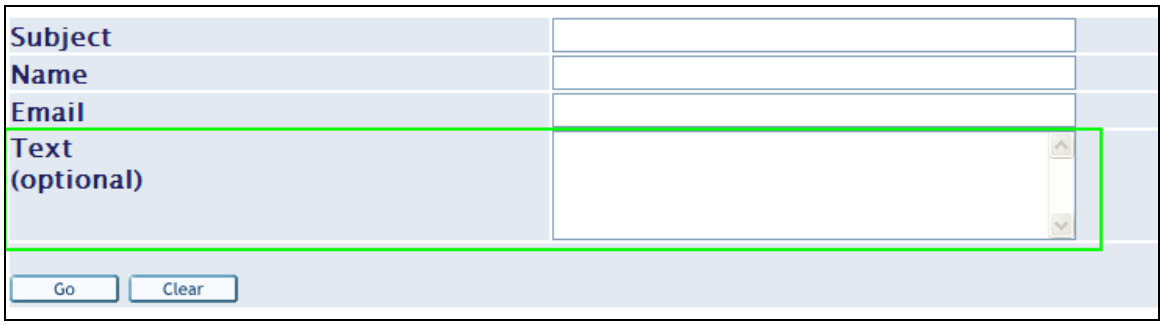
```
<Tr>
<td class=td2> Verification:</td>
  <td class=td2>
  <input type=password
    size=21
    maxlength=20
    name=bor_verification
    value ""=>
 </td>
 </tr>
```

To:

```
<Tr>
<td class=td2><label for="pat_password"> Verification:</label></td>
  <td class=td2>
  <input id="pat_password"
    type=password
    size=21
    maxlength=20
    name=bor_verification
    value ""=>
 </td>
 </tr>
```

26.5.2 Text Area Fields

The following is an example of a text area field:



Change the code from:

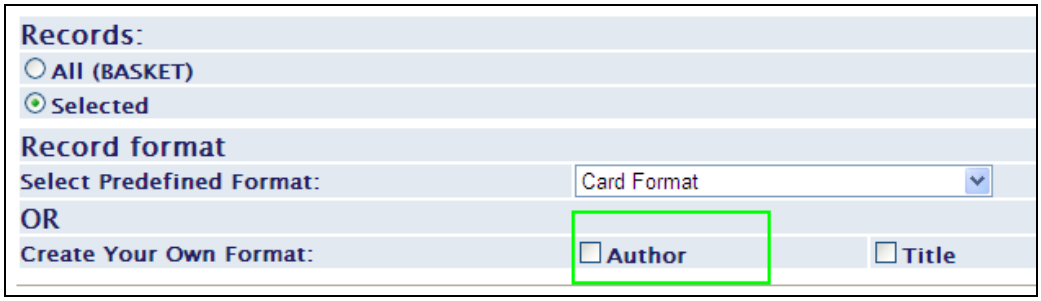
```
>tr<
                <td valign=top class=td2>Text
>  br>(optional)</td<
> td class=td2<
>  textarea name="text" cols=39 rows=4>$0800</textarea<
/> td<
              </td>
             </tr>
```

To:

```
>tr<
                <td valign=top class=td2><label for="text">Text
>  br>(optional)</label></td<
> td class=td2<
>  textarea id="text" name="text" cols=39 rows=4>$0800</textarea<
/> td<
              </td>
             </tr>
```

26.5.3 Checkbox Fields

The following is an example of a checkbox field:



Change the code from:

```
              <td class=td2 id=small >
                <input type="checkbox"
                  name="own_format"
                  value="1%23%23%23%23">Author
              </td>
```

To:

```
              <td class=td2 id=small >
                <input id="author" type="checkbox"
                  name="own_format"
                  value="1%23%23%23%23"><label
for="author">Author</label>
              </td>
```

26.5.4 Radio Button Fields

The following is an example of a radio button field:

Change the code from:

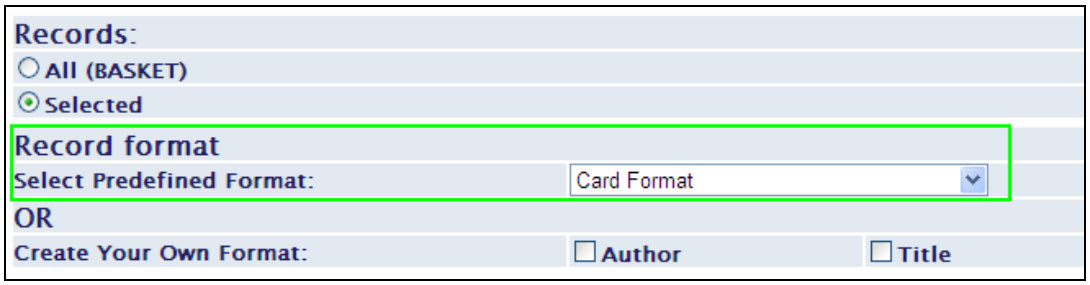
```
    <td class=td2 id=small >
      <input type="radio"
             name="encoding"
             value="UTF_TO_WEB_MAIL_ASCI"
             checked>ASCII (recommended / default)
    <td>
```

To:

```
    <td class=td2 id=small >
      <input id="ascii" type="radio"
             name="encoding"
             value="UTF_TO_WEB_MAIL_ASCI"
             checked><label for="ascii">ASCII (recommended / default)
    </label></td>
```

26.5.5 Drop-Down List Fields

The following is an example of a drop-down list field:



Change the code from:

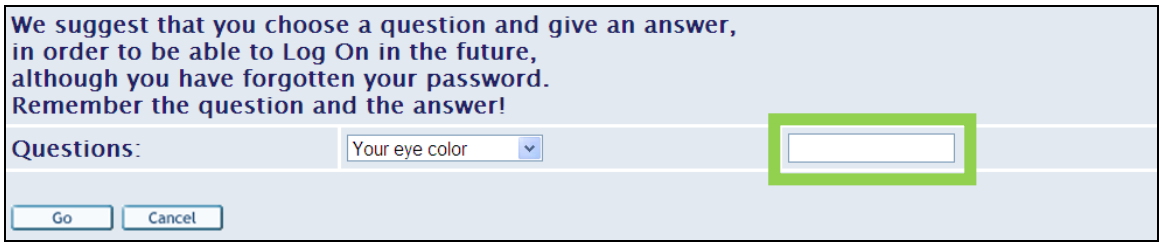
```
<tr>
  <td class=td2 id=small width=20% nowrap>
      Select Predefined Format:
  <td>
  <td class=td2 id=small colspan=7>
   <select name=format>
          <option value=037              >Card Format</option>
          <option value=001              >MARC Tags</option>
          <option value=002              >Name Tags</option>
          <option value=997              >MARC Format</option>
```

```
          <option value=998              >ALEPH Sequential</option>
          <option value=999              >OPAC Format</option>
          <option value=777                  >ISI ResearchSoft Tagged
Output Format</option>

   </select>
  </td>
 </tr>
```

To:

```
<tr>
  <td class=td2 id=small width=20% nowrap><label for="format">
      Select Predefined Format:
  </label></td>
  <td class=td2 id=small colspan=7>
   <select id="format" name=format>
          <option value=037              >Card Format</option>
          <option value=001              >MARC Tags</option>
          <option value=002              >Name Tags</option>
          <option value=997              >MARC Format</option>
          <option value=998              >ALEPH Sequential</option>
          <option value=999              >OPAC Format</option>
          <option value=777              >ISI ResearchSoft Tagged
Output Format</option>

   </select>
  </td>
 </tr>
```

If the input field does not have an associated caption, for example:



Create a hidden label. For example:

```
<td  class=td2><label  for="question_answer"  class=hidden>  Question
answer</label>
      <input type=password
            id="question_answer"
            value=""
            name=PASSWORD_VALUE
            size=16
            maxlength=15>
  </td>
```

Define the hidden class in the Web OPAC CSS file as:

```
.hidden
{
    position:absolute;
    left:-10000px;
    top:auto;
    width:1px;
    height:1px;
```

```
    overflow:hidden;
}
```

26.6 Color Indications

Information or restrictions indicated using colors are not detectable to screen readers. Therefore, make sure that the pages do not indicate information or restrictions with color. For example, red asterisks to indicate mandatory fields are not detected by screen readers.

Use the title attribute for such items.

For example:

```
<table cellspacing=2 border=0 width="60%">
 <tr>
  <td width=15% class=td2 nowrap><label for="patron_id">
    Borrower ID/Barcode :
  </label></td class=td2>
  <td class=td2>
      <span class=text2 title="Mandatory">*</span>
  </td>
  <td class=td2>
    <input id="patron_id" size=30 name=id maxlength=12 value="">
  </td>
 </tr>
 <tr>
```

26.7 Additional Changes for Aleph Web Files

Change the following HTML files, as follows, to support accessibility:

- **bor-loan-body**

Line 4 – change from:

```
    <td  class=td1  valign=top  align="center"><input  type="checkbox"
name="$1100"></td>
```

To:

```
    <td       class=td1       valign=top       align="center"><label
for="select_row_$2200" class=hidden>Select     Row</label>
    <input id="select_row_$2200" type="checkbox" name="$1100"></td>
```

- **booking-matrix-body**

Change from:

```
                <td class="$3100" align="center" nowrap><br></td>
                <td class="$3200" align="center" nowrap><br></td>
          <td class="$3300" align="center" nowrap><br></td>
                <td class="$3400" align="center" nowrap><br></td>
                <td class="$3500" align="center" nowrap><br></td>
          <td class="$3600" align="center" nowrap><br></td>
```

```
                <td class="$3700" align="center" nowrap><br></td>
                <td class="$3800" align="center" nowrap><br></td>
          <td class="$3900" align="center" nowrap><br></td>
                <td class="$4000" align="center" nowrap><br></td>
          <td class="$4100" align="center" nowrap><br></td>
          <td class="$4200" align="center" nowrap><br></td>
          <td class="$4300" align="center" nowrap><br></td>
         <td class="$4400" align="center" nowrap><br></td>
         <td class="$4500" align="center" nowrap><br></td>
         <td class="$4600" align="center" nowrap><br></td>
         <td class="$4700" align="center" nowrap><br></td>
         <td class="$4800" align="center" nowrap><br></td>
         <td class="$4900" align="center" nowrap><br></td>
         <td class="$5000" align="center" nowrap><br></td>
         <td class="$5100" align="center" nowrap><br></td>
         <td class="$5200" align="center" nowrap><br></td>
         <td class="$5300" align="center" nowrap><br></td>
         <td class="$5400" align="center" nowrap><br></td>
```

To:

```
          <td     class="$3100"     title="$5500"     align="center"
nowrap><br></td>
         <td      class="$3200"      title="$5600"      align="center"
nowrap><br></td>
         <td      class="$3300"      title="$5700"      align="center"
nowrap><br></td>
         <td      class="$3400"      title="$5800"      align="center"
nowrap><br></td>
         <td      class="$3500"      title="$5900"      align="center"
nowrap><br></td>
         <td      class="$3600"      title="$6000"      align="center"
nowrap><br></td>
         <td      class="$3700"      title="$6100"      align="center"
nowrap><br></td>
         <td      class="$3800"      title="$6200"      align="center"
nowrap><br></td>
         <td      class="$3900"      title="$6300"      align="center"
nowrap><br></td>
         <td      class="$4000"      title="$6400"      align="center"
nowrap><br></td>
         <td      class="$4100"      title="$6500"      align="center"
nowrap><br></td>
         <td      class="$4200"      title="$6600"      align="center"
nowrap><br></td>
         <td      class="$4300"      title="$6700"      align="center"
nowrap><br></td>
         <td      class="$4400"      title="$6800"      align="center"
nowrap><br></td>
         <td      class="$4500"      title="$6900"      align="center"
nowrap><br></td>
         <td      class="$4600"      title="$7000"      align="center"
nowrap><br></td>
         <td      class="$4700"      title="$7100"      align="center"
nowrap><br></td>
         <td      class="$4800"      title="$7200"      align="center"
nowrap><br></td>
              <td    class="$4900"    title="$7300"    align="center"
nowrap><br></td>
         <td      class="$5000"      title="$7400"      align="center"
nowrap><br></td>
```

```
        <td      class="$5100"      title="$7500"      align="center"
nowrap><br></td>
        <td      class="$5200"      title="$7600"      align="center"
nowrap><br></td>
        <td      class="$5300"      title="$7700"      align="center"
nowrap><br></td>
        <td      class="$5400"      title="$7800"      align="center"
nowrap><br></td>
```

- **item-hold-request-head**

The pickup options list comes from the code along with the ID of the label (id=”pickup”). Therefore, define the label in the HTML page as “pickup”.

Change from:

```
<td class="td2" width=20% nowrap>Pickup/delivery location :</td>
```

To:

```
<td class="td2" width=20% nowrap><label for="pickup">Pickup/delivery
location :</label></td>
```

- **group-request-head**

The pickup options list comes from the code along with the ID of the label (id=”pickup”). Therefore, define the label in the HTML page as “pickup”.

Change from:

```
<td class="td2" width=20% nowrap>Pickup location :</td>
```

To:

```
<td class="td2" width=20% nowrap><label for="pickup">Pickup location
:</label></td>
```

26.8 Skipping Repetitive Links

Give pages an option to skip repetitive links in order to allow users with a screen reader to avoid hearing the same links in each page to which they browse.

Add the following lines in each place you want to skip.

Add the highlighted line at the beginning of the code:

```
<A HREF="#skip-section" class="hidden">Skip Section</A>
<include>include-window
<table   width="100%"   border="0"   cellspacing="0"   cellpadding="0"
align="center">
  <tr class="topbar">
    <td valign="middle" height="20" width="103">
      <a  href="http://www.exlibris.co.il"  class="blue"  target=new
title="Ex Libris, developer of the ALEPH Integrated Library System">
      <img      src="&icon_path/f-aleph-logo-1.jpg"      width="103"
height="20" border="0" alt="Aleph logo"></a></td>
    <td   class="topbar"   valign="middle"   height="20"   width="*"
align="center" colspan="15">
       &f-base
    </td>
  </tr>
```

And the following line at the end:

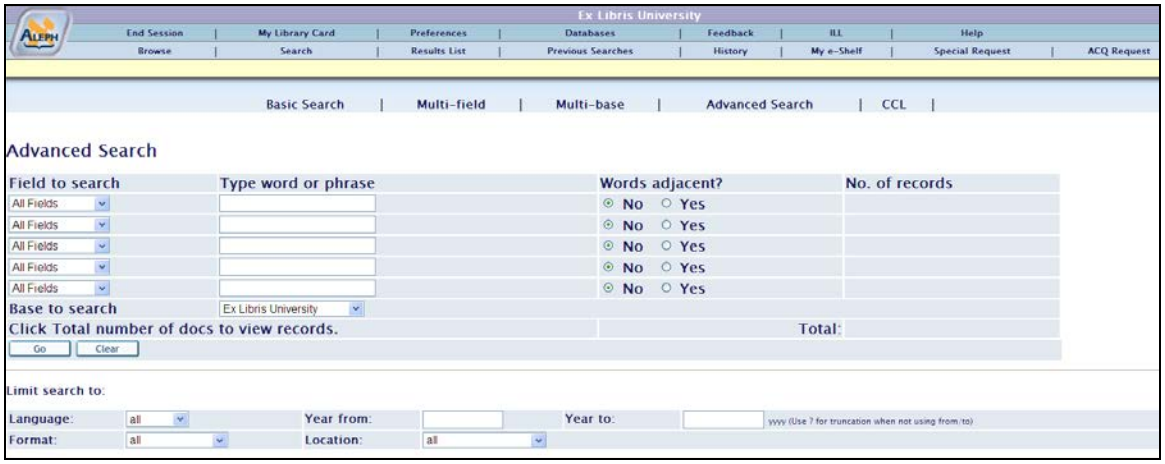
```
  <tr>
```

```
    <td    valign="middle"    height="20"    background="&icon_path/f-
spacer.jpg" colspan="17" nowrap>
       
    </td>
  </tr>
</table>
<A NAME="skip-section"></A>
```

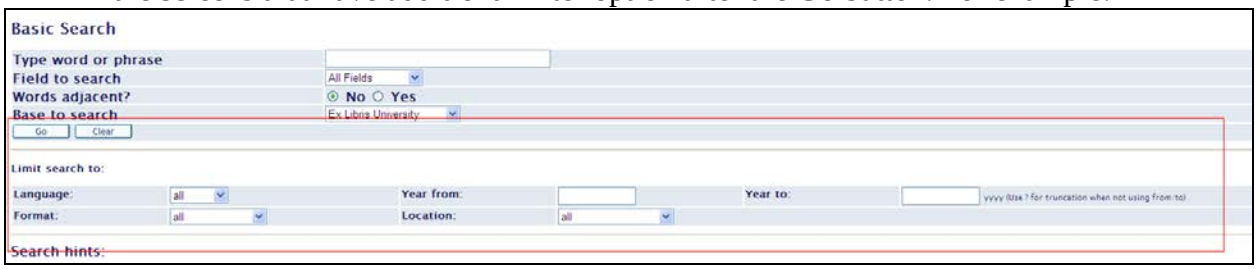
26.9 Exceptions

The following pages have a limited level of accessibility:

- Advanced search



- All the screens that have additional filter option after the Go button. For example:



Auto refresh is not accessible according to section 508; however, in the Aleph Web OPAC, auto refresh can be made accessible using the following tag in:

```
<META HTTP-EQUIV="REFRESH" CONTENT="1200; URL=http://il-
aleph09.corp.exlibrisgroup.com:8991/F/Y3Q77Y6GMASJVXBGJ7FPLE88YVIX5DH
T6FU6T3NK627BFY2SCD-00001?func=logout">
```

Note: Since most screens reader support java script functionality, the <non script> tag has not been added to the Aleph Web OPAC.

27 Link to Google Books Service

It is possible to link a record's full view, which is displayed after a search query, to the Google Books Service. To implement this option, add the following to the full-set-tail HTML page after the "Choose format" table:

```
<a  id="$0800" style="display:none;">
<img           style="display:none;" id="google_book"/>
About this book
</a>
<script>
// Function to process GBS info and update the dom.
function ProcessGBSBookInfo(booksInfo) {
  var url_elem = document.getElementById("$0800");
  var pic_elem = document.getElementById("google_book");
  for  (var  items_idx  =  0;  items_idx  <  booksInfo.totalItems  ;
items_idx++)
  {
        var item = booksInfo.items[items_idx];
        if(!item.volumeInfo  ||  !item.volumeInfo.industryIdentifiers)
{
                continue;
        }
        url_elem.href = item.volumeInfo.infoLink;
        url_elem.style.display = '';
        if (item.volumeInfo.imageLinks.smallThumbnail)
        {
           pic_elem.src = item.volumeInfo.imageLinks.smallThumbnail;
           pic_elem.style.display = '';
        }
   }
</script>
<script src="$0700"></script>
```

28 Forgot Password

Patrons can reset their password to the OPAC by activating the Forgot Password? link in the Web OPAC login page. Based on the library setup, the link can provide two options for resetting a forgotten password:

- Answering a security question whose answer was previously defined

or

- Sending an email to the patron's email account that provides the ability to reset the password.

The functionality of the Forgot Password flow is set in the following login pages:

./alephe/www_f_lng/login

./alephe/www_f_lng/login-acq-session

./alephe/www_f_lng/login-ill

./alephe/www_f_lng/login-session

28.1 Reset Password by Answering a Predefined Question

In order to implement the question-answer option, set the following line in the login pages (listed above).

```
<a href="&server_f?func=file&file_name=bypass-passwd">
```

28.2 Reset Password by Getting an Email

In order to implement the reset password option by getting an email, set the following line in the login pages (listed above):

```
<a href="&server_f?func=file&file_name=forgot-password">
```

- The following files should be set in to ./alephe/www_f_lng:

```
forgot-password
forgot-password-rst
```

- Make sure the following lines are not set in ./alephe/www_f_lng/bor-new:

```
           <td colspan=3>
             <hr>
           </td>
          </tr>
          <tr>
           <td class="text9" colspan=3>
              If you forgot your password, you can use the "Password
              Bypass"  <br>  mechanism  by  selecting  one  of  the
questions and answering it.
           </td>
          </tr>
          <tr>
           <td        class=td2        width=20%        nowrap><label
for="question">Question Database:</label></td>
           <td class=td2>
               <select id="question" name=N_PASSWORD_CODE>
                 <option      value=01      $$2500-S"01">Your      eye
color</option>
                 <option     value=02     $$2500-S"02">Your     lucky
number</option>
                 <option      value=03      $$2500-S"03">Size      of
shoes</option>
                 <option    value=04    $$2500-S"04">Your    father's
name</option>
                 <option    value=05    $$2500-S"05">Your    mother's
name</option>
               </select>
           </td>
           <td  class=td2><label  for="question_answer"  class=hidden>
Question answer</label>
               <input       id="question_answer"       type=password
name=N_PASSWORD_VALUE size=16 maxlength=15 value="$2600">
           </td>
          </tr>
          <tr>
```

- Make sure the following lines are not set in./alephe/www_f_lng/ bor-update-password

```
  <td colspan=3>
    <hr>
  </td>
 </tr>
 <tr>
  <td class=td2 colspan=3>
     We suggest that you choose a question and give an answer,
     <br> in order to be able to Log On in the future,
```

```
     <br> although you have forgotten your password.
     <br> Remember the question and the answer!
  </td>
 </tr>
 <tr>
  <td class=td2 width=20% nowrap><label
for="questions">Questions:</label></td>
  <td class=td2>
      <select id="questions" name=PASSWORD_CODE>
        <option value=01 $$0100-S"01">Your eye color</option>
        <option value=02 $$0100-S"02">Your lucky number</option>
        <option value=03 $$0100-S"03">Shoe size</option>
        <option value=04 $$0100-S"04">Your father's name</option>
        <option value=05 $$0100-S"05">Your mother's name</option>
      </select>
  </td>
  <td class=td2><label for="question_answer" class=hidden> Question
answer</label>
      <input type=password
            id="question_answer"
            value=""
          name=PASSWORD_VALUE
          size=16
          maxlength=15>
  </td>
 </tr>
 <tr>
```

- Define the varible `www_max_hours_reset_pw` in ./alephe/www_server.conf

This variable defines the maximum number of hours in which the URL that is sent by mail is active.

The variable can be from 01 to 99 (two-digits)

Default value: 24

For example:

```
setenv www_max_hours_reset_pw       12
```

- Set the following files in /aleph/error_lng/www_f_heading:

```
0390 L Fail to submit the Reset Password form. Please apply to
library Staff!

0391 L The Password verification fields do not match. Please retype
again

0392 L Your request to reset password has been received. A message
has been sent to your Email address.
```

- Set the following files in ./[library]/tab/form_description.lng

```
reset-password-mail               Resetting Library account password

reset-password-confirm          Successful reset of your library
account password
```

- Set the following files in ./[library]/form_lng/ :

```
reset-password-mail-00

reset-password-confirm-00
```

29 Web OPAC-Related Tables

General

Table Extensions

You can use different tables for Web OPAC by using the file extensions: *filename*. *sublibrary, base* and so on.

Example:

edit_doc_999.lng.*ulinc*

Library Tables

The following is a list of some of the configuration tables that are related to Web OPAC. They are located in the bibliographic library's tab directory unless otherwise stated (for example `xxx01/tab/tab05.lng`).

DISPLAY TABLES

edit_doc.lng

edit_doc.lng is used to define the display of document information. It defines a concatenation of a number of paragraphs. It is used in conjunction with `edit_field.lng` and `edit_paragraph.lng`.

edit_doc_777

This table defines tag conversion routines and tags to be included as the output of the 777 format. This format is intended for the export of bibliographic records in "ISI ResearchSoft Tagged Output Format". This format can be used to import records into reference managers such as Endnote.

edit_doc_999.lng

`edit_doc_999.lng` defines the linked display format for documents for viewing in the Web OPAC. The 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 or not links to other records are displayed.

edit_doc_999_aut_xxx10.lng

edit_doc_999_aut_xxx10.lng defines the display of AUT records for cross- reference display. In the Web OPAC, clicking on a reference allows for the display of the linked Authority Record. The display of the record is defined in the `edit_doc_999_aut_xxx10.lng` table where "xxx10" is the code of the AUT library (for example, USM10).

edit_field.lng

edit_field.lng defines the display of a tag. It is used in conjunction with `edit_paragraph.lng` and `edit_doc.lng`. If a tag is not defined in `edit_field.lng`, it will be displayed as is. Therefore, tags entered in the database,

without punctuation, need to be defined here. You can define up to 10 filters for each subfield per tag name.

edit_paragraph.lng
edit_paragraph.lng defines the display of a paragraph of data from the bibliographic record. The `edit_paragraph.lng` table defines a concatenation of a number of fields. It is used in conjunction with `edit_field.lng` and `edit_doc.lng`.

Note that formats 001-099 are used by the BIB_FORMAT program for direct display of specific bibliographic information (for example, on User List of Loans, OPAC List of Items, and so on).

tab05.lng
This table defines the caption to be displayed in the Web OPAC before $$n and $$m.

tab06
Defines non-highlighted words in the Full View format in the Web OPAC.

tab22
This table defines the fields that are included in the "short-doc" (z13) - and also defines the fields in the My e-Shelf Brief format.

tab_buf_z403
This table defines URI links and the way in which they are displayed.

tab_events.lng
`tab_events.lng` defines which transactions will write a record to the Z35 or Z69 Oracle tables. While `tab_event.lng` in the administrative library defines which transactions write a record to the Z35 Oracle table, the `tab_event.lng` table in the bibliographic library defines which transactions write a record to the Z35 Oracle table (types 10,11,12) or Z69 oracle table (types 20-35). While the transactions in the Administrative library relate to Circulation, the transactions in the bibliographic library relate to the Web OPAC and to the z39 server.

If the `tab_event.lng` table in the bibliographic library does not exist in the database, then the events will not be created. This does not cause a problem.

tab_item_list_order
The table defines the display order of the items in the Web and in the Search functions in the ALEPH GUIs according to the sublibraries for each base.

tab_service
This table defines the Web OPAC services that are available for a particular field. The services currently available are:

- SCAN (ACC or IND) which activates display of a browse list.
- FIND (ACC) which activates the find command.
- AUT which activates display of the authority record to which the field heading is linked.

- ENGINE which activates an external search engine, using details defined in Z121.

tab_sfx
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 are available for a record that was found on a specific base, and which icon is displayed to signify that the services are available.

tab_sort
tab_sort defines fields for sorting. This table is used in conjunction with:

- the `option-sort` HTML screen
- `www_server.conf`
- `tab01.lng` - document record fields (for correct filing of z101 sort keys).
- `www_f_sort_heading.lng`

tab_sub_lib_sort
This table contains definitions of the sort order of a sublibraries list in various ALEPH functions.

tab_z30_sort
`tab_z30_sort` defines items (z30 sort types) in various ALEPH functions. It is located in the library's `data_tab` directory.

tab_z121
This table contains definitions for the Web service search engine. These definitions include the search engine name, and the URL.

www_f_tab_short_999
`www_f_tab_short_999` defines the short display of bibliographic data in custom formats (but not in table format) in the Web OPAC. You can set up different formats, varying the field codes used, and the order in which they appear. You must use format numbers from 951-998 for your custom formats.

Note that a format with the format number 999 must be present in the table. The 999 format is used by the system if a format that does not exist in the table is used. Format number 950 is reserved for the full `edit_doc_999.lng` view.

www_f_sort_heading.lng
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.lng` defines the sort options and can be edited using UTIL J/2.

www_tab_myshelf_short.lng
This table defines which information is displayed in the columns of the short display in the My e-Shelf function.

The order of the columns defined in this table determines the order of the display, in conjunction with the `myshelf-short-head-2`, `myshelf-short-head-2-no-course` and `myshelf-short-body` HTML files in the www_f_lng directory.

www_tab_short.lng
`www_tab_short.lng` defines the short display of bibliographic data in the Web OPAC. The short display of bibliographic information in table format will be the default in the online Web OPAC when:

- "option-display-table" is chosen (From the Preferences screen, option display format, option brief view in table);

  or

- If the library has "000" set as the default display format in www_server.conf.

INDEXING TABLES

tab00.lng
`tab00.lng` defines the system index files. There should be one such table for each language defined.

tab01.lng
`tab01.lng` defines per language the type and name of the library and tag definitions for the bibliographic records.

tab11_acc
The `tab11_acc` table is used to assign fields to headings indexes.

tab11_ind
The `tab11_ind` table is used to assign fields to direct indexes.

tab11_word
The `tab11_word` table is used to assign fields to word indexes.

tab20
This table defines the headings (including see references) that are built in the ACC table of the bibliographic base, based on the fields of the authority record that "matches" the BIB ACC Headings.

tab_filing
The `tab_filing` table defines which filing procedures are used when building the filing key for Heading (Z01), Index (Z11) entries and Sort keys (Z101). The filing procedures identifier of these is set in col. 5 of `tab00.lng`.

tab_word_breaking
This table defines routines for word breaking procedures.

path_convert

The path_convert tables under the libraries tab directory redirect files and tables. Every library has its own set of tables and files that can be maintained separately. However, it is possible for libraries to share the same table or direction by setting up redirection in the path_convert tables. For example, two libraries can use the same Web screens, forms, tab directories and so on.

Note that you can define general Web directories under alephe.

Other Tables

library_relation
alephe/tab/library_relation defines the connections between libraries.

Note

If changes are made to the table, the servers must be reactivated in order to have the changes reflected in the clients.

tab100
tab100 is the central configuration table for system-level, server-level and library-level variables.

There are a number of Web OPAC-related tab100 variables. They are:

1. ACC-COUNT-LIMIT. This variable controls the maximum number of records shown in the "Number of records" column in the headings list in Web OPAC after a browse.

   The default value is 50. If there are more records than the number set in this variable, the system leaves off counting, and ++ displays, rather than the number itself. The upper limit for this counter is 998. Note that a high number affects system performance so it is suggested to set this number no higher than 50. When the user clicks on the heading to display the list of records, the actual count is clearly displayed.

2. ACC-COUNT-STYLE This variable determines how the counter of a number of related BIB records will be displayed in a browse list when the heading is a "see" reference.

   Possible values are: S or R.

   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.

   The default value is S.

3. AUTO-TRUNCATE-Z01-FIND. This variable controls the automatic truncation option in Find.

   Possible values are: Y or N.

   Y = when a FIND search is performed, the search term is automatically set as truncated. For example, the search term *TIT=Introduction to demography* retrieves all titles beginning with *Introduction to demography.....*

   The default value is N. If the value is N, then the same query will only retrieve titles containing the exact wording *Introduction to demography* and will not retrieve titles containing any additional text.

4. BASE-DEMO-LIB This variable is used for the tables' synchronization report (UTIL H/1), and for UTIL G/2 values.

   Possible values are: 1 or 2.

   1 = the system's demo database is USMXX.

   2 = the system's demo database is MABXX.

   The default value is 1.

5. BROWSE-NO-MATCH This variable determines whether or not a message is displayed in the browse results list. This message states that an exact match for the browsed string should have appeared here and was not found.

   If the variable is set to Y, code 9982 that resides in $alephe_root/error_lng directory under www_f_heading will appear immediately preceding the next best match in the list.

   If the parameter is set to N, the message will not appear.

   The default value is N.

6. HOLD-BY-ITEM-GROUP This variable controls the pickup location list policy when hold requests are placed from the full view.

   Possible values are: Y or N.

   Y = When a hold request is created from the Web OPAC, the list of pickup locations will be built according to the tab37 pickup locations of all items that are like copies of the requested item.

   N = The list of pickup locations is built only according to the tab37 line on the requested item.

7. FILTER-LIBRARY-CARD This variable controls the behavior of the library card with regard to shared ADM libraries in which the patron is not active.

   Possible values are: Y or N.

   Default : N

   Y = When showing the library card, show only ADM libraries which share this patron and in which the patron is active, meaning that there are loans, hold requests, booking requests, photocopy requests or cash transactions that belong to this patron.

   N = When showing the library card, show all ADM libraries which share this patron, regardless of whether the patron is active in them or not.

Note that in either case, the ADM to which the patron's home library belongs will be displayed first.

8. ITMG-DISPLAY This variable lets you control the links of full-999, so that links are only displayed for sublibraries that have real z30 and not just HOL records.

   Possible values are: Y or N.

   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. The default value is N.

9. PDS-AWARE

   This variable controls the access method to the Web OPAC, i.e. whether or not PDS will be used to manage the OPAC authentication and authorization process.

   Note that the variable is configurable only in the alephe/tab tab100 table.

   Possible values are Y or N. The default value is N.

   Y = Use PDS for user authentication with the Web OPAC.

   N = Do not use PDS for user authentication with the Web OPAC.

10. PDS-KEY-TYPE

    This variable defines the Z308-KEY-TYPE against which all PDS authentications are verified. Multiple key types may be defined by using a comma separator, for example, 01,02,03.

    Note that the variable is configurable only in the alephe/tab tab100 table.

Note: If you are working with PDS in Oracle mode, the identifiers can only be up to 100 characters in length. This is a PDS limitation.

11. UNION-LIBRARY Determines how the system will display equivalent BIB records.

    Possible values are: 0, 1 or 2.

    0 = BIB database does not use "union" features

    1 = Union View Catalog: BIB database uses Z120 and Z127 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 and Z127 to display equivalent BIB records as one record. Holdings information is embedded in the BIB record, and there is no item display.

    The default value is 0.

12. X852-HOLDING This variable is used for union catalogs and defines whether or not to use a special program for holdings from the 852 tag.

    Possible values are: Y or N.

    Y = whether to use a special program that presents a list of holdings taken from an 852 tag. For union catalogs.

    The default value is N.

13. Z39-LOGIN-BASE-CHECK This variable restricts the non-signed users connecting via Z39.50 accessing limited bases marked with * in tab_base.lng.

    Possible values are: Y, N

    The default value is N.

    Y= non-signed users connecting via Z39.50 have access to all the bases that have user Z39 defined except access to limited bases marked with * in tab_base.lng

    N= non-signed users connecting via Z39.50 have access to all the bases that have user Z39 defined

14. XSS-VALIDATION This variable determines whether to activate the HTML validation added to prevent an XSS breakthrough (a malicious insertion of HTML script by a user).

    Possible values are: Y, N

    The Default value is N

    Y= Active the HTML validation

    N= Non Active HTML validation

tab_base.lng_count

This table contains definitions of the Web databases displayed in the browse list.

tab_base.lng

tab_base.lng defines the logical and physical databases that can be accessed by the user. Logical bases are defined by setting a FIND command that serves as a pre-filter or scope. In order to set a logical base to include everything **except** a group of records, use "alldocuments" to define "everything", together With "not". for example all not wsp=suppressed table format.

You can set a mandatory sign-in in order to access a base (this can be a library or a logical base). This is done in `alephe/tab/tab_base.lng`, by placing an asterisk in column 2. There is no check for specific user names; any valid user name will gain access to the database.

tab_ccl_boolean_operators

The `alephe/tab/tab_ccl_boolean_operators` 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.

tab_ill_unit.lng

The `./alephe/tab/tab_ill_unit.lng` table defines which ILL Centers are displayed in the drop-down list when the ILL Request link is clicked. The server address of each ILL Center is defined in this table.

Tab_z36_sort

The [adm_lib]/tab/tab_z36_sort table defines which sort routine will be present for the loan list. The loan list in the WEB is sorted using the entry WWW-USER.

tab_mime_type

`alephe/tab/tab_mime_type` assigns a special file extension to any document format. This file extension is used automatically when documents are saved on the computer in Web OPAC. If a format is not assigned an extension, the *.sav* extension is used.

www_server.conf

This table defines parameters for the Web server. For more information see Setting Up the System on page 8.

30 Cancelling Hold Requests

Cancel a hold request in the Web OPAC by clicking the **Delete** button in the following Library Card pages:

- The hold request full display
- The title request list
- The title requests full display

The Web OPAC may be configured to allow cancelling a hold request only before the library started processing the hold request (i.e. no call slip has been printed yet). It may also be configured to allow cancelling a hold request after the requested item has already been placed on the hold shelf.

Configuring these options is done by manipulating the following HTML files:

- bor-hold-exp (for the hold request full display)
  - Use the $0700 place-holder to allow cancelling only unprocessed requests.
  - Use the $2300 place-holder to allow cancelling on hold shelf requests.
- bor-title-exp (for the title request list)
  - Use the $0700 place-holder to allow cancelling only unprocessed requests.

- Use the $2100 place-holder to allow cancelling on hold shelf requests.

- bor-title-body (for the title request full display)
  - Use the $2000 place-holder to allow cancelling only unprocessed requests.
  - Use the $2100 place-holder to allow cancelling on hold shelf requests.

Library Card pages, to enable cancelling the request both before it has been processed and after the item is already on the hold shelf, place both place-holders on the relevant HTML page.

31 Staff Request and Requester Name Field

The Central Library is a library that offers services to subscribed libraries. Subscribed libraries may request material from the central library on behalf of their patrons. The subscribed libraries are identified in the central library as "pseudo patrons", but their patrons are not identified in the central library.

When a patron of the subscribed library wants to request material from the central library, the staff of the subscribed library creates the request for the patron. The Requester name field is filled with information about the requesting patron, for tracking purposes.

31.1 Requester Name Field on Request Creation Forms

1. Display the requester name field on hold and photocopy request creation forms:

   In the following files:

   - alephe/www_f_lng/item-hold-request-tail
   - alephe/www_f_lng/item-photo-request-tail
   - alephe/www_f_lng/item-photo-request-tail-an (if you are using this file)

   Insert the following lines:

```
<tr>
 <td class="td2">Requester Name </td>
 <td class="td2">
   <input size=30 maxlength=100 name="requestername"  value="$3800"></td>
</tr>
```

2. Apply the "clear" function also to requester name field:

   In the following files:
   - alephe/www_f_lng/item-hold-request-head
   - alephe/www_f_lng/item-photo-request-head

Insert the bolded line:

```
<script language="Javascript">
<!--
 function my_reset()
 {
  for (var i = 0; i< document.form1.elements.length; i++) {
   if (document.form1.elements[i].name == "author" ||
      document.form1.elements[i].name == "title" ||
      document.form1.elements[i].name == "pages" ||
      document.form1.elements[i].name == "note" ||
      document.form1.elements[i].name == "note2" ||
      document.form1.elements[i].name == "addlinfo" ||
      document.form1.elements[i].name == "requestername") {
     document.form1.elements[i].value = "";
    }

  }
 }
// -->
</script>
```

3. Apply the "back" option also to requester name field:
   In the following files:
   - alephe/www_f_lng/item-hold-request-details
   - alephe/www_f_lng/item-photo-request-details

Insert the following bolded line:

```
<a href="&server_f?func=item-photo-
request&doc_library=$0100&adm_doc_number=$0200&item_sequence=$0300&yea
r=$2500&volume=$2600&sub_library=$2700&type=$2800&no_loaned=$2900&star
t_rec_key=$3000&end_rec_key=$3100&bib_doc_num=$0600&bib_library=$0500&
source_form=PHOTO_DETAILS&author=$4200&title=$4300&pages=$4400&note1
=$4500&note2=$4600&pickup=$4100&addlinfo=$4700&requestername=$3800" >
<img src="&icon_path/f-back.gif" border=0 alt="back to Photocopy Request form">
```

- In order to display the requester name field on request confirmation screen:
  In the following files:
  - alephe/www_f_lng/item-hold-request-details
  - alephe/www_f_lng/item-photo-request-details

Insert the following line:

```
<input type=hidden name="requestername" value="$3800">
```

When the form is submitted, the request is linked to the subscribed library's user with the Requester Name saved in the new Z37-REQUESTER-NAME / Z38-REQUESTER-NAME field.

If cash charges are created, they are charged for the subscribed library's user, and the Z31-REQUESTER-NAME field is populated with the input requester name.

Note: Hold and Photocopy creation forms might be defined with different suffixes for different patrons.

In this case, add the Requester Name field to request forms with the relevant suffix.

For more information, see the **Different Hold Requests forms** and **Different Photocopy Requests forms** sections of the *Aleph 21 System Librarian's Guide-Circulation*.

31.2 Requester Name Information in My Library Card

- In order to display requester name information in cash list:
  Insert the following line to alephe/www_f_lng/bor-cash-head:
  <th class="text3">Requester Name</th>
  Insert the following line to alephe/www_f_lng/bor-cash-body:
  <td class=td1 valign=top>$1100</td>

- In order to display requester name information in requests list:
  Insert the following line to alephe/www_f_lng/bor-hold-head:

<th class="text3">Requester Name</th>

  Insert the following line to alephe/www_f_lng/bor-hold-body:

<td class=td1 valign=top>$1900</td>

- In order to display requester name information in requests list:
  Insert the following line to alephe/www_f_lng/bor-photo-head:

<th class="text3">
Requester Name
</th>

  Insert the following line to alephe/www_f_lng/bor-photo-body:

<td class=td1 valign=top>$1100 </td>

- In order to display requester name information in request expanded information:
  Insert the following line to alephe/www_f_lng/bor-hold-exp:

<tr>
<td class=td1 id=bold align=left>Requester Name:</td>
<td class=td1> $2400</td>
</tr>

- In order to display requester name information in request expanded information:
  Insert the following line to alephe/www_f_lng/bor-photo-exp:

<tr>
<td class=td1 id=bold align=left>Requester Name:</td>
<td class=td1 align=left> $1500</td>
</tr>

31.3 Sort Based on Requester Name Information

- In order to sort my library card → hold requests list according to open date and requester name, set 05 in www_user_z37_sort_routine and environment variable (defined in `www_server.conf`).

- In order to sort my library card → photocopy requests list according to open date and requester name, set 08 in www_user_z38_sort_routine and environment variable (defined in `www_server.conf`).

- In order to sort my library card → cash list according to transfer number, date and requester name, use 09 sort routine in ./<ADM library>/tab/tab_z31_sort, and define matching entries in
./<ADM library>/tab/pc_tab_exp_field.lng under CASH-SORT-TYPE.

- In order to sort my library card → cash list according to requester name, use 10 sort routine in ./<ADM library>/tab/tab_z31_sort, and define matching entries in
./<ADM library>/tab/pc_tab_exp_field.lng under CASH-SORT-TYPE.

Refer to the **tab_z31_sort** chapter of the **Aleph 21 Configuration Guide** for more details.

32 Server Side Scripts in the Aleph OPAC

The OPAC HTML pages can be updated with the tag <program> that enables a call to a server-side script. In order to add content from an external script into an HTML page, add a line in the following format to the desired place in the HTML page:

<program> external script name

Place the external script in a directory specially created for this purpose, under the u tree.

For example: create a new ext directory under `$alephe_root`.

In `aleph_start`, define the environment variable `opac_ext_prog` as the path of this directory. For example:

```
setenv    opac_ext_prog   $alephe_root/ext
```

The application identifies the tag <program> as a request to execute a script and calls the program defined as external script name from the `$opac_ext_prog` directory.

Example from an HTML page:

```
<include>bor-include-4
<program>add_more_bor_info.pl
<div class=title>
```

Place the `script add_more_bor_info.pl` in the `$opac_ext_prog` directory.

The external scripts are written and maintained by the libraries only. The performance and the content of the scripts are the responsibility of the libraries and should be validated before usage.

32.1 Script Input

The script must be based on the data available in the Web session information table: Z63. The Aleph OPAC programs pass all the relevant information to the external script as parameters. The information available from Z63 is detailed in the following table. Note that the parameters must be read by the external script according to the order defined in the table.

The Aleph OPAC programs generate the output file name (specific to the current session) and send to the script this name as a parameter. The external script uses this name if it has HTML lines to write as an output.

Parameter Number	Parameter	Z63 field
1	Output file	---
2	Session ID	Z63-SESSION-ID
3	Base	Z63-BASE
4	Patron ID	Z63-BOR-ID
5	Patron ADM library	Z63-BOR-LIBRARY
6	Current language	Z63-CON-LNG
7	Search result set number	Z63-SET-NUMBER

32.2 Script Output

The script manages the data retrieval and displays it in HTML format. The output of the script (HTML lines) is written to an output file whose name is received as a parameter by the external script.

The Aleph OPAC programs replace in the HTML page the following line with the output of the external script, which must be valid HTML code.

<program>script name

33 List of Web OPAC Files

This section includes the list of Web files for the ALEPH Web OPAC interface that reside in the `../alephe/www_f_lng` directory. The list includes HTML files that support the following functions and subfunctions:

- ADAM on page 133
- Advance Booking Information on page 147

- Advanced Search on page 131
- Basic Search on page 130
- BIB Request on page 137
- Booking Error Page on page 143
- Availability on page 143
- Browse on page 131
- Browse list - ACC on page 131
- Calendar on page 151
- Cash Information on page 148
- CCL Search on page 131
- Confirmation Page on page 143
- Cross references on page 145
- Databases on page 151
- Error on page 151
- Feedback on page 151
- Filter on page 140
- Full on page 133
- Help on page 151
- History on page 138
- History-Loan Information on page 146
- Hold Information on page 146
- History-Hold Request Information on page 147
- ILL on page 141
- Ill Center on page 142
- ILL Information on page 147
- Information display from Holdings Schema records, element set B1 on page 135
- Information display from Holdings Schema records, element set B2 on page 135
- Information display from Holdings Schema records, element set C2 on page 136
- Item Booking on page 142
- Items on page 134
- Loan Information on page 145
- Locate on page 133

- Login on page 144
- Mail on page 150
- Multi-base Search on page 130
- Multi-field Search on page 130
- My e-Shelf on page 138
- Note for borrower on page 150
- Password on page 150
- Photo Information on page 147
- Preferences on page 145
- Previous Searches on page 138
- Profile on page 149
- Proxy Information on page 149
- Rank on page 145
- Refine on page 140
- Results lists on page 132
- Save File on Server on page 140
- SDI on page 149
- Search on page 130
- Services on page 134
- SFX on page 150
- Special Request on page 140
- Update Address on page 149
- User on page 145
- User Information on page 145
- Z39.50 Item on page 135

You can check which files a particular page is made from by using the source viewing option on your browser (for example, right-click and select **View Source** in Internet Explorer 5.0 and later).

The file names are written at the top of the file in a HTML tag. For example:

```
<!-- filename: find-b -->
```

Note

Files that are present in the installed system, but not listed in this section (List of Web OPAC Files), can be ignored, but should not be removed. They are not relevant for the configuration of your Web OPAC.

For information on customizing Web OPAC pages, refer to the document *Web OPAC Screens: Structure and Customization.*

33.1 List of Web OPAC Files

Function	Subfunction	Filename	Include Files
Search			
	Basic Search	Find-b	meta-tags head-1-bor/nobor find-include-buttons find-b-head find-code-include find-include-filter copyrights
		Find-b-permute-head	meta-tags head-1-bor/nobor find-code-include
		Find-b-permute-head-top (for CJK)	meta-tags head-1-bor/nobor find-code-include
		find-b-permute-body	
		find-b-permute-tail	copyrights
		find-b-list-head	meta-tags head-1-bor/nobor find-include-buttons find-b-head
		find-b-list-body	
		find-b-list-tail	copyrights
	Multi-field Search	find-a	meta-tags head-1-bor/nobor find-include-buttons find-include-filter copyrights
	Multi-base Search	find-m	meta-tags head-1-bor/nobor find-include-buttons find-code-include find-include-filter copyrights
		find-m-results-head	meta-tags head-1-bor/nobor
		find-m-results-body	

Function	Subfunction	Filename	Include Files
		find-m-results-tail	copyrights
	Advanced Search	find-d	meta-tags head-1-bor/nobor find-include-buttons find-code-include find-d-code-include (usm90) find-include-filter copyrights
	CCL Search	find-c	meta-tags head-1-bor/nobor find-include-buttons find-include-filter copyrights
	Search/Browse	find-e-head	find-e-code-include base-list-include find-include-filter
	Floating Keyboard	floating-keyboard	floating-keyboard.js meta-tags
	Refined Search	find-r	meta-tags head-1-bor/nobor find-include-buttons find-code-include find-include-filter copyrights
Browse			
	Browse	scan	meta-tags head-1-bor/nobor scan-include-2 copyrights
	Browse list - ACC	scan-acc-head	include-window meta-tags head-1-bor/nobor scan-include-2
		scan-acc-body	
		scan-acc-tail	copyrights
		scan-ind-head	meta-tags head-1-bor/nobor scan-include-2
		scan-ind-body	
		scan-ind-tail	copyrights
		scan-word-head	meta-tags head-1-bor/nobor scan-include-2

Function	Subfunction	Filename	Include Files
		scan-word-body	
		scan-word-tail	copyrights
		scan-sys-head	meta-tags head-1-bor/nobor scan-include-2
		scan-sys-body	
		scan-sys-tail	copyrights
		scan-acc-ug-head	meta-tags head-1-bor/nobor scan-include-2
		scan-acc-ug-body	
		scan-acc-ug-tail	copyrights
	Browse headings list	heading-acc-head	meta-tags head-1-bor/nobor
		heading-acc-head-1	
		heading-acc-head-2	
		heading-acc-body	
		heading-acc-tail	copyrights
Results lists			
		short-1-head	submit-form-include meta-tags include-window short-select-include head-1-bor/nobor short-include-buttons short-include-2
		short-2-head	meta-tags include-window submit-form-include short-select-include short-include-buttons short-include-2
		short-4-head	submit-form-include meta-tags include-window short-select-include head-1-bor/nobor short-include-buttons short-include-2
		short-tail	copyrights
		short-a-head	
		short-a-body	

Function	Subfunction	Filename	Include Files
		short-a-tail	
		short-b-head	
		short-b-body	
		short-b-tail	
		short-c-head	
		short-c-body	
		short-b-body-separator	
		short-c-tail	
Full			
		full-set-head	include-window meta-tags head-1-bor/nobor
		full-set-head-bor	include-window meta-tags head-1-bor/nobor
		full-set-head-nobor	include-window meta-tags head-1-bor/nobor
		full-set-head-bor-ext-ill	include-window meta-tags head-1-bor/nobor
		full-set-head-nobor-ext-ill	include-window meta-tags head-1-bor/nobor
		full-999-body	
		full-xxx-body	
		full-000-body	
		full-999-separator	
		full-set-tail	tail-bar-include copyrights
		direct-head	meta-tags include-window head-1-bor/nobor
		direct-tail	tail-bar-include copyrights
	External Location	media-permission	meta-tags include-window
		media-no-permission	meta-tags copyrights
	Locate	Locate-list	meta-tags copyrights
	ADAM	full-999-body-media	
		full-head-media	
	Down links (view)	dn-link-view-head	meta-tags include-window head-1-bor/nobor
		dn-link-view-body	
		dn-link-view-tail	tail-bar-include copyrights

Function	Subfunction	Filename	Include Files
Services			
		service-head-tag	meta-tags
		service-find-acc-head	
		service-find-acc-tail	
		service-scan-acc-head	
		service-find-m-acc	
		service-find-m-acc-tail	
		service-scan-acc	
		service-scan-acc-tail	
		service-engine	
		service-scan-ind	
		service-aut	
		service-tail	copyrights
		service-aut-bib	
		service-gen-exec	meta-tags
		service-sfx-record	
		service-no-service	meta-tags copyrights
		www-ext-copyrights	meta-tags
		www-ext-copyrights-9	
		media-display-exec	meta-tags
Items			
		item-global-head-1	meta-tags head-1-bor/nobor full-xxx-body
		item-global-head-2	
		item-global-body	
		item-global-body-head	
		item-global-tail	copyrights
		item-global-tail-1	
		item-global-tail-2	
		item-global-body-no-item	
		item-global-body-tail	
		item-global-dropdown-menus-a	

Function	Subfunction	Filename	Include Files
		item-global-dropdown-menus-b	
		item-global-dropdown-menus-b1	
		item-global-dropdown-menus-b2	
		item-global-dropdown-menus-b3	
		item-global-dropdown-menus-b4	
		item-global-dropdown-menus-b5	
		item-global-dropdown-menus-c	
		item-global-prev-next	
		login-item	meta-tags head-1-bor/nobor copyrights
		item-global-exp	include-window meta-tags head-1-bor/nobor copyrights
	Z39.50 Item		
		item-global-ug-head-1	meta-tags include-window head-1-bor/nobor
		item-global-ug-head-2	
		item-loan	meta-tags head-1-bor/nobor copyrights
		item-hold-request-head	meta-tags head-1-bor/nobor
		item-hold-request-body	
		item-hold-request-tail	copyrights
	Information display from Holdings Schema records, element set B1	item-ug-2-head-bb1	
		item-ug-2-body-bb1	
	Information display from Holdings Schema records, element set B2	item-ug-2-head-bb2	
		item-ug-2-body-bb2	

Function	Subfunction	Filename	Include Files
	Information display from Holdings Schema records, element set C2	item-ug-2-head-bc2	
		item-ug-2-body-bc2	
	Information display from Holdings, remote library items, with no circulation, information (ACC central database).	item-cen-head-1	meta-tags head-1-bor/nobor
		full-999-body	
		item-global-tail-2	
		item-cen-head-2-a	
		item-cen-head-2	
		item-cen-body	
		item-cen-body-no-item	
		item-cen-tail	copyrights
		item-cen-tail-2	copyrights
ACQ Request	Acquisition Request - New BIB record	acq-request	meta-tags head-1-bor acq-order-group-include copyrights acq-request-doc-head-conf
	Confirmation for Acquisition Request	acq-request-confirm	meta-tags head-1-bor acq-order-group-include
		acq-request-doc-tail-conf	acq-order-group-include copyrights
	Acquisition Request - existing BIB record	acq-request-doc-head	
		acq-request-doc-tail	acq-order-group-include
Photocopy Request			
	Item Photocopy Request	item-photo-request-head	meta-tags head-1-bor/nobor
		item-photo-request-body	

Function	Subfunction	Filename	Include Files
		item-photo-request-tail	copyrights
BIB Request		bib-request-item-list-head	include-window meta-tags head-1
		bib-request-item-list-head-1	
		bib-request-item-list-head-1-r	
		bib-request-item-list-head-1-v	
		bib-request-item-list-head-2	
		bib-request-item-list-head-3-begin	
		bib-request-item-list-head-3-coll	
		bib-request-item-list-head-3-p-stat	
		bib-request-item-list-head-3-stat	
		bib-request-item-list-head-3-sublib	
		bib-request-item-list-head-3-tr	
		bib-request-item-list-head-filter	
		bib-request-item-list-body-begin	
		bib-request-item-list-body-repeat	
		bib-request-item-list-body-tr	
		bib-request-item-list-tail	
		bib-request-verify	meta-tags head-1 copyrights
		bib-request-no-items	meta-tags head-1 copyrights
		bib-request-1	
		bib-request-2	

Function	Subfunction	Filename	Include Files
		item-hold-request-details	meta-tags head-1 copyrights
Previous Searches			
		history-head	meta-tags submit-form-include head-1-bor
		history-body	
		history-tail	copyrights
		history-cross	meta-tags head-1-bor
		history-cross-body	
		history-cross-tail	copyrights
History			
		save-history-head	meta-tags submit-form-include head-1-bor/nobor
		save-history-body	
		save-history-tail	copyrights
My e-Shelf		myshelf-folder-list	meta-tags head-1
		myshelf-course	replace-textarea-js include-replace-hash replace-textarea-hash-js meta-tags head-1 include-select-format copyrights
		myshelf-mail	replace-textarea-js include-replace-hash replace-textarea-hash-js meta-tags head-1 include-select-format copyrights
		myshelf-short-head	meta-tags myshelf-short-select-include head-1
		myshelf-short-head-2	

Function	Subfunction	Filename	Include Files
		myshelf-short-head-2-no-course	
		myshelf-short-body	
		myshelf-short-tail	copyrights
		myshelf-short-folder-list	
		myshelf-short-select-include	
		myshelf-full-head	meta-tags include-window head-1
		myshelf-full-tail	tail-bar-include copyrights
		myshelf-folder-list-body-no-doc	
		myshelf-folder-list-left-head	include-window myshelf-short-select-include
		myshelf-folder-list-left-body-1	
		myshelf-folder-list-left-body-2	
		myshelf-folder-list-left-body-3	
		myshelf-folder-list-left-body-no-folder	
		myshelf-folder-list-left-tail	
		myshelf-folder-list-right-head	include-window myshelf-short-select-include
		myshelf-folder-list-right-body-1	
		myshelf-folder-list-right-tail	
		myshelf-note-full	meta-tags head-1 copyrights
		myshelf-note-short	meta-tags head-1 copyrights
		myshelf-refresh-page	include-window myshelf-short-select-include
		myshelf-save-as-fol	include-window myshelf-short-select-include
		myshelf-no-fol-for-delete	include-window myshelf-short-select-include
		myshelf-delete-folder	meta-tags myshelf-short-select-include

Function	Subfunction	Filename	Include Files
		myshelf-no-doc-basket	include-window myshelf-short-select-include
		myshelf-clear-basket	meta-tags myshelf-short-select-include
		Save-file	head-1-bor/nobor copyrights
	My e-shelf - Request	group-request-head	meta-tags head-1-bor/nobor
		group-request-body	
		group-request-tail	copyrights
		group-request-exec-head	meta-tags head-1-bor/nobor
		group-request-exec-body	
		group-request-exec-tail	copyrights
Special Request			
		new-record	meta-tags head-1-bor/nobor copyrights
		new-book	meta-tags head-1-bor/nobor copyrights
Save File on Server			
		save-server	meta-tags head-1-bor/nobor copyrights
	save-server-d	meta-tags head-1-bor/nobor copyrights	
Refine			
		short-refine	meta-tags head-1-bor/nobor find-code-include copyrights
Filter			
		short-filter	meta-tags head-1-bor/nobor copyrights
	Text Filter	short-filter-s	meta-tags head-1-bor/nobor copyrights
	Range Filter	short-filter-r	meta-tags head-1-bor/nobor copyrights
		short-filter-y	meta-tags head-1-bor/nobor copyrights

Function	Subfunction	Filename	Include Files
	Filter by date	short-filter-a	meta-tags head-1-bor/nobor copyrights
ILL			
		ill-book-request	replace-textarea-js meta-tags head-1-bor/nobor ill-include-buttons ill-request-main-include copyrights
		new-ill-book-request	popcalendar-js meta-tags head-1-bor/nobor new-ill-r-mn ill-request-main-include copyrights
		ill-journal-request	replace-textarea-js meta-tags head-1-bor/nobor ill-include-buttons ill-request-main-include copyrights
		new-ill-journal-request	popcalendar-js meta-tags head-1-bor/nobor new-ill-r-ser ill-request-main-include copyrights
		ill-conference-request	replace-textarea-js meta-tags head-1-bor/nobor ill-include-buttons ill-request-main-include copyrights
		ill-report-request	replace-textarea-js meta-tags head-1-bor/nobor ill-include-buttons ill-request-main-include copyrights
		ill-request-confirm	meta-tags head-1-bor/nobor copyrights
		ill-request-doc-head	meta-tags head-1-bor/nobor

Function	Subfunction	Filename	Include Files
		ill-request-doc-tail	ill-request-main-include copyrights
	ILL Request - Alternate Unit Selection	ill-alternate-unit-h	meta-tags include-window head-1-bor/nobor
		ill-alternate-unit-b	
		ill-alternate-unit-t	copyrights
	ILL Center	ill-unit-head	meta-tags
		ill-unit-body	
		ill-unit-tail	
		close-window	
	Error message - ILL	message-page	meta-tags head-1-bor/nobor copyrights
Item Booking	Booking Request Form	booking-page-form-head	
		booking-page-form-body	
		booking-page-form-tail	
		booking-page-form-head-no-pickup	
		booking-page-form-body-no-pickup	
		booking-page-form-tail-no-pickup	
		bib-info-table-head	
		booking-return-loc-head	
		booking-page-head	include-window meta-tags js-calendar head-1
		booking-page-tail	
		booking-page-item-info	
		booking-page-delivery-form-head	
		booking-page-form-body	
		booking-page-delivery-form-tail	

Function	Subfunction	Filename	Include Files
		booking-page-pickup-form-head	
		booking-page-form-body	
		booking-page-pickup-form-tail	
		booking-page-return-pickup-head	
		booking-page-return-pickup-tail	
	Availability	booking-matrix-head	
		booking-matrix-body	
		booking-matrix-tail	
		booking-req-list-head	meta-tags replace-page-js head-replace bor-include-3
		booking-req-list-body	
		booking-req-list-tail	copyrights
		booking-req-no-booking	meta-tags replace-page-js head-replace bor-include-1 copyrights
		booking-req-expand-head	meta-tags replace-page-js head-replace
		booking-req-expand	copyrights
		item-avail-page-head	include-window meta-tags
		item-avail-page-tail	copyrights
		item-avail-head	
	Confirmation Page	booking-req-conf-1	meta-tags replace-page-js head-replace
		booking-req-conf-2	copyrights
	Booking Error Page	booking-error-head	include-window meta-tags head-1
		booking-error-body	

Function	Subfunction	Filename	Include Files
		booking-error-tail	copyrights
		booking-error-page	include-window meta-tags head-1
	Login from Booking Request	item-booking-request-1	
Advance Booking	Advance Booking - Time slots	item-schedule-head	meta-tags head-1-bor/nobor
		item-schedule-body	
		item-schedule-tail	
Login			
		login	meta-tags copyrights
		login-session	meta-tags head-1-bor/nobor copyrights
		login-request	meta-tags head-1-bor/nobor copyrights
		login-acq-session	meta-tags head-1-bor/nobor copyrights
		login-ill	meta-tags copyrights
		patron-notice	meta-tags head-1-bor/nobor copyrights
		logout	meta-tags head-1-bor/nobor copyrights
		logout-1	
		bor-new	meta-tags replace-page-js head-replace head-1-bor/nobor sub-library-select-usm50 sub-library-select-usm51 copyrights
		bor-new-message	meta-tags replace-page-js head-replace copyrights
		bypass-passwd	meta-tags copyrights
		pds-login	
		pds-login-first	
		pds-sso	

Function	Subfunction	Filename	Include Files
		pds-down	
		pds-logout	
Preferences			
		option-display	meta-tags head-1-bor copyrights
		option-display-session	meta-tags head-1-bor/nobor copyrights
		option-lng	meta-tags head-1-bor copyrights
		option-lng-session	meta-tags head-1-bor/nobor copyrights
Cross references			
		accref-head	meta-tags
		accref-body	
		accref-tail	copyrights
Rank			
		short-rank	meta-tags head-1-bor/nobor copyrights
User			
	User Information	bor-info Or: bor-info-single-adm (in case of single ADM)	meta-tags replace-page-js head-replace copyrights
	Loan Information	bor-loan-head	meta-tags replace-page-js collect-data-js head-replace bor-include-4
		bor-loan-body	
		bor-loan-tail	Copyrights
		bor-loan-exp-head	meta-tags replace-page-js head-replace
		bor-loan-exp	copyrights
		bor-loan-no-loan	meta-tags replace-page-js head-replace bor-include-1 copyrights
		bor-renew-all-head	meta-tags replace-page-js head-replace
		bor-renew-all-head-2	
		bor-renew-all-head-3	

Function	Subfunction	Filename	Include Files
		bor-renew-all-head-renewed	
		bor-renew-all-head-not-renewed	
		bor-renew-all-gbl-error	meta-tags replace-page-js head-replace copyrights
		bor-renew-all-body	
		bor-renew-all-tail	Copyrights
	History-Loan Information	bor-history-loan-head	meta-tags replace-page-js head-replace bor-include-3
		bor-history-loan-body	
		bor-history-loan-tail	Copyrights
		bor-history-loan-exp-head	meta-tags replace-page-js head-replace
		bor-history-loan-exp	
		bor-history-loan-no-loan	meta-tags replace-page-js head-replace bor-include-1 copyrights
	History-Loan Information Expanded details for a loan	bor-history-loan-exp-head	meta-tags
		bor-history-loan-exp	
		bor-history-loan-exp-1	
		bor-history-loan-exp-2	copyrights
	Hold Information	bor-hold-head	meta-tags replace-page-js head-replace bor-include-3
		bor-hold-body	
		bor-hold-tail	copyrights
		bor-hold-exp-head	meta-tags replace-page-js head-replace
		bor-hold-exp	copyrights
		bor-hold-no-hold	meta-tags replace-page-js head-replace bor-include-1 copyrights

Function	Subfunction	Filename	Include Files
	History-Hold Request Information	bor-history-hold-head	bor-include-3
		bor-history-hold-no-hold	bor-include-1
		bor-history-hold-tail	
		bor-history-hold-exp	
		bor-history-hold-exp-head	
	Photo Information	bor-photo-head	meta-tags replace-page-js head-replace bor-include-3
		bor-photo-body	
		bor-photo-tail	copyrights
		bor-photo-exp-head	meta-tags replace-page-js head-replace
		bor-photo-exp	copyrights
		bor-photo-no-photo	meta-tags replace-page-js head-replace bor-include-1 copyrights
	Advance Booking Information	bor-item-schedule-head	meta-tags replace-page-js head-replace bor-include-1
		bor-item-schedule-body	
		bor-item-schedule-tail	copyrights
		bor-item-schedule-exp	meta-tags replace-page-js head-replace copyrights
		bor-item-schedule-no-hold	meta-tags replace-page-js head-replace bor-include-1 copyrights
	ILL Information - List of ILL Requests	bor-ill-head	meta-tags replace-page-js head-replace bor-include-1
		bor-ill-body	
		bor-ill-tail	copyrights
		bor-ill-exp	meta-tags replace-page-js head-replace copyrights

Function	Subfunction	Filename	Include Files
		bor-ill-no-ill	meta-tags replace-page-js head-replace bor-include-1 copyrights
	ILL request details	bor-ill-exp-ext	meta-tags replace-page-js head-replace
		bor-ill-exp-ext-tail	copyrights
	Cash Information	bor-cash-sub-head	meta-tags replace-page-js head-replace bor-include-3
		bor-cash-sub-body	
		bor-cash-head	
		bor-cash-body	
		bor-cash-tail	copyrights
		bor-cash-exp-head	meta-tags replace-page-js head-replace
		bor-cash-exp	
		bor-cash-exp-1	
		bor-cash-exp-tail	Copyrights
		bor-cash-no-cash	meta-tags replace-page-js head-replace bor-include-1 copyrights
	ACQ Requests for patron	bor-acq-no-acq	
		bor-acq-head	meta-tags replace-page-js head-replace bor-include-1
		bor-acq-body	
		bor-acq-tail	copyrights
	Title Request Information	bor-title-head	meta-tags replace-page-js head-replace bor-include-1
		bor-title-body	
		bor-title-tail	copyrights
		bor-title-exp-head	meta-tags replace-page-js head-replace
		bor-title-exp	copyrights

Function	Subfunction	Filename	Include Files
		bor-title-no-hold	meta-tags replace-page-js head-replace bor-include-1 copyrights
	Proxy Information	bor-proxy-head	meta-tags replace-page-js head-replace bor-include-1
		bor-proxy-body	
		bor-proxy-tail	Copyrights
		bor-proxy-no-proxy	meta-tags replace-page-js head-replace bor-include-1 copyrights
	Update Address	bor-update	meta-tags replace-page-js head-replace copyrights
		bor-update-without-date	meta-tags replace-page-js head-replace copyrights
	Profile	bor-profile-display	meta-tags replace-page-js head-replace sub-library-select-usm50 copyrights
	SDI	bor-sdi-head	meta-tags submit-form-include replace-page-js head-replace
		bor-sdi-body	
		bor-sdi-tail	Copyrights
		bor-sdi-update	meta-tags replace-page-js js-calendar head-replace adm-library-select sub-library-select-usm50 sub-library-select-usm51 copyrights
		bor-sdi-delete	meta-tags replace-page-js head-replace copyrights
		bor-sdi-no-sdi	meta-tags replace-page-js head-replace copyrights

Function	Subfunction	Filename	Include Files
		sdi-refine	meta-tags head-1 find-code-include find-include-filter copyrights
	Password	bor-update-password	meta-tags replace-page-js head-replace copyrights
	Note for borrower	bor-note-new	meta-tags replace-page-js head-replace sub-library-select-usm50 copyrights
		bor-note-display-head	meta-tags replace-page-js head-replace
		bor-note-display-body	
		bor-note-display-tail	Copyrights
		bor-info-no-z305	meta-tags replace-page-js head-replace copyrights
	Mail from Activities	user-info-mail	replace-page-js include-replace-hash replace-textarea-hash-js meta-tags head-1 copyrights
Mail			
		short-mail	meta-tags replace-textarea-js head-1-bor/nobor include-select-format copyrights
		full-mail	meta-tags replace-textarea-js head-1-bor/nobor include-select-format copyrights
SFX			
		sfx-title	
		sfx-subject	
		sfx-au	

Function	Subfunction	Filename	Include Files
		sfx-record	
		sfx-empty	
		sfx-admin	
		sfx-record-ver-0.1	
		sfx-record-ver-1.0	
		sfx-button-data	
Databases		base-list	meta-tags head-1-bor/nobor copyrights
Help		help-1	meta-tags copyrights
Error		forbidden-access	
Calendar		calendar	
Feedback		feedback	meta-tags replace-textarea-js head-1-bor/nobor copyrights
User Limit Error		license-limit	