System Librarian's Guide - Printing

Version 22

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.

Document released: November 12, 2019

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

Table of Contents

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, 4=browse HTML).

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
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!
item-copy-label            00 EXECUTE BIAF_LABEL_PRINT
item-copy-label            07 EXECUTE BIAF_LABEL_PRINT
item-issue-label           00 EXECUTE BIAF_LABEL_PRINT
item-issue-label           20 EXECUTE BIAF_LABEL_PRINT
serial-item-label          00 EXECUTE BIAF_LABEL_PRINT
serial-item-label          01 EXECUTE BIAF_LABEL_PRINT
```

2. **Install the BIAF Print Label Software**

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.