



ALEPH VERSION 16.02

How to use the Patron Loader Interface Format (PLIF)

Ex Libris

© Ex Libris Ltd., 2006
Last Update: January 25, 2006
Document Version 1.2
Code: A-ver16-HTPLIF-1.2

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.

Any references in this document to non-Ex Libris Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Ex Libris product and Ex Libris has no liability for materials on those Web sites.

Copyright Ex Libris Limited, 2006. All rights reserved.

Documentation produced January 2006

Document version 1.2

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

Table of Contents:

Overview.....	4
Determining the Match Point.....	4
Data elements in the user section.....	5
Data elements in the address section	5
Data elements in the borrower section.....	5
Assuring Uniqueness of Z308 IDs.....	6
Actions	7
Using the ignore and spaces characters	7
Post-PLIF	8
Sample line	8
Record structure for the PLIF input file	9
Notes (on Record Structure):	12
Running p_file_20 from the command line.....	12
From the Version 16.02 Help (New Info added):.....	13

Overview

[Note: The yellow highlighting indicates features new/changed in version 16.02.]

Ex Libris provides a utility, `p_file_20`, to load patron records in machine-readable format into ALEPH. Most commonly, the records are student and faculty/staff records that come from the campus bursar's office on campus. The `p_file_20` job expects the patron records to be in a pre-defined format, referred to as the Patron Loader Interface Format (PLIF).

The system librarian should work with their Campus IT department to rearrange the bursar export data into the PLIF format. In almost all cases, there are some mappings that need to be determined by the system librarian based on ALEPH configuration tables.

Determining the Match Point

The system librarian should consult the conversion specifications to find what the Patron ID (Z308) types are. A typical arrangement of Z308 ID types might be

- Type 00 – ALEPH, system assigned
- Type 01 – Patron's barcode number
- Type 02 – Campus ID number
- Type 03 – Social Security number

The IT department and the system librarian need to determine what number ALEPH and the Bursar system have in common to provide a match point. Most likely the match point will be a campus ID number. Type 00 is internal to ALEPH, and the bursar would have no idea what that number was. The barcode might change if the student replaces his ID card. Some campuses prohibit the use of SSN to identify a person.

Alternatively, the system librarian may look in the online GUI client at the "Additional IDs" of a sample converted record to determine what numbers they have in common with the bursar.

Once the IT staff and library have determined what ID they have in common with the ALEPH patron record, they will use that as the match point to overlay a patron record in ALEPH with the incoming PLIF data.

Note that The MATCH-ID is used only for matching - the loader program does not add this ID automatically. If you want to be able to match on an ID later (for updating purposes), you need to add the ID using the ID section (described below). IDs of type 00 will automatically be created, as they are the unique ALEPH system number.

Data elements in the user section

The User section of the PLIF is mostly standard field translations (name, birth date) or default values (ILL borrowing limit). The library should give the default values for ILL borrowing, given that there might be different values for faculty and for students -- for example 1000 limit, but students can have only 100.

USER-PROFILE. The PROFILE is what determines default settings when a patron logs into the Web OPAC. Profiles are set in the ADMIN client. A typical example might be that faculty members have a profile of FACULTY, which allows them to save course reserve files to the server – this is a privilege set in the ADMIN client for that profile.

USER-DELINQ. The bursar can only send one delinquency (block) to ALEPH. The library might choose to do this if the student is in bad academic or financial standing and the library wants to know so they can limit their borrowing privileges. There are three delinquency fields in the ALEPH patron record, so if the bursar is sending one of them, they need to specify which of the fields to populate. If they want to populate the first delinquency field, then USER-REC-DELINQ-INDEX would be '1', the USER-REC-DELINQ would be the delinquency code (taken from xxx50/tab/tab_delinq.eng), and the USER-REQ-DELINQ-NOTE would be the textual description of this block.

USER-FIELD fields are similar in structure to the USER-DELINQ fields.

USER-REC-PROXY-FOR-ID and USER-REC-PRIMARY-ID are used in the creating proxy borrowers. The ALEPH functionality behind this is that a student may be given privileges to check out materials for a faculty research professor – without actually carrying the professor's card to the library, and without having the checked-out items appear on the student's record.

Data elements in the address section

Address of type 01 is for permanent (parent's home) and type 02 is for campus (dorm/office) address. The Address expiration dates should be checked with library staff, but it would be based on the student's registration – for example, if they are registered for fall term, the expiration date might be mid-January. Then, in the January loads, the date would be again updated to mid-June if they registered for Spring term. Faculty would probably have expiration dates far into the future.

Data elements in the borrower section

The Borrower record section also requires discussion between the IT department and the Librarian as well. The IT department needs to translate their statuses (Faculty, Staff, Undergrad, Grad, and so on) into the values in ALEPH. The patron values are listed in tab31, and the descriptions are in pc_tab_exp_field.eng:

BOR-STATUS	L Undergrad Student	01
BOR-STATUS	L Grad Student	02
BOR-STATUS	L Faculty - Staff	03
BOR-STATUS	L Community	05
BOR-STATUS	L CMLE	19
BOR-STATUS	L ILL	20
BOR-STATUS	L Retired Fac - Staff	26

So probably if you have a faculty member, they would have an ALEPH record of 03 and a XXX50 record of 03. Undergrads would have an ALEPH record of 01 and an XXX50 record of 03. Consortia may have different values for ALEPH and XXX50 records, and if the librarian is unable to determine the mapping, they should consult their Ex Libris project librarian. Expiration dates in the Borrower section should be determined in a similar manner to address expiration dates.

Records that are updated or inserted will take Borrower defaults from Tab31, based on the Borrower Status and Sublibrary, with the exception of the expiration date.

Assuring Uniqueness of Z308 IDs

The z308_rec_key (and z303_rec_key) needs to be unique. If it is not, you will get an Oracle "Unique constraint violated" error.

p_file_20 matches on the USER-REC-MATCH-ID (usually the university ID/SSN). If a barcode (or other ID) in the PLIF record already exists for another user, p_file_20 detects this condition and issues an error message.

The Oracle Z308 unique constraint error occurs when p_file_20 has **generated** an ID -- which it expects to be unique -- but it is not.

There are two cases where p_file_20 generates an ID:

- (1) it **always** generates a Z303 / Z308 type '00' ALEPH user number;
- (2) if there is no ID type '01' (barcode) in the input file -- or if its ID-NO field is blank -- p_file_20 will generate Z308 type '01' barcode.

The value for the ALEPH user number comes from the util g/2 last-bor-id counter for the ADM library.

The value for the barcode (if not in the input record) comes from the util g/2 last-bor-id-1 counter for the ADM library.

Thus, this error would occur when one of these counters is giving a value which duplicates an existing value.

Prior to version 16 or in a single ADM situation in version 16, one can use this SQL to locate the highest value for the Z308:

```
SQL> select max(z308_rec_key) from z308 where substr(z308_rec_key,1,2) = '01';
```

<'01' would be for the barcode; '00' for the ALEPH user ID>

Note: in version 16, the z308 is in the \$usr_library.

But with a multi-ADM situation in version 16-up, the last-bor-id/last-bor-id-1 counters in the various ADM libraries need to be synchronized with each other to make sure that their values do not overlap. There are two ways to do this: (1) specify a different prefix for each ADM; or (2) specify a different starting key number for each ADM which will not collide with the others.

Actions

Each section above has an action associated with it, and that action is used to direct the p_file_20 job what to do with the data in ALEPH.

A – Update if a match is found, and if a match is not found, add a new record with the same information. (Combination of I and U) This is the most common action.

U - Update all fields in input record. If a record is sent with a U and there is no existing record, an error will be reported.

I - Insert the record. If the record exists already, the insert will fail and the record will not be updated.

D - Delete the section of the record.

X - Do Nothing

There is an action associated with every sub-section of an input line. It would be possible to start with a user record that has only a matching ID and action "X" and follow that with address lines for update. This allows selective update of records associated with a patron.

Using the ignore and spaces characters

The PLIF format allows for two different characters to either a) overwrite a data field in ALEPH with spaces; or b) ignore the data field, that is, leave the data in the record in ALEPH as it is. For example, the PLIF spaces character could be '%' and the ignore character could be '+', which would be specified in the submission of the job p_file_20. So use the single character '%' in the input file to indicate that the ALEPH field in the database should be blanked out, no matter what is in it. Use the single character '+' in the PLIF input file to indicate that we should leave the ALEPH field in the database alone. The ignore character is useful if you know that librarians routinely update a field – like the user's e-mail address – when we trust the data in ALEPH more than we trust the data in from the bursar's office.

The spaces character and the ignore character cannot be any of the following: *, ? or ^ . They have other meanings in Unix. Either the spaces character or the ignore character can be the actual space character (' ') if desired.

A CLARIFICATION about how parameters 5 and 6 work: “If I have “%” as the spaces character and “ ” as the ignore character, does that mean that someone's name in the user-rec-name field should appear as:

Smith,%John%% (up to 200)? ”

Answer: Parameters 5 and 6 are operating at the *field* level, not at the individual character level. For parameter 5 the idea is that the field would have just a single occurrence of this character. If the user had previously had an email address and that address was now invalid and there was no replacement, then you would put a single % in the email-address field. (It may also be OK if the field had *all* %%%%%%%%%%'s , but I'm not sure. I would not do it that way....)

Similarly, if you have blank for the parameter 6 ignore character, p_file_20 interprets a field which has blank as the first character and no other data as an ignore character. (But in that case I'm pretty sure that having a field with all blanks is OK.)

The user-rec-name should appear as “Smith, John ” (up to 200).

Post-PLIF

No indexing jobs need to be run after execution of p_file_20. The only index involved (the Z111 User record Keywords) is updated automatically.

Sample line

The following sample record uses the ‘A’ action to update someone by matching on ID type 03, but if that ID type 03 is not found, a new patron record will be added. The whole line would be structured as below – with spaces and line breaks added for readability, but the whole record would in fact be one line. Also, spaces have been compacted for readability.

User section:

```
A 03 12345 Swanson, Kristin 1000 1000 Y 010202 <--- 1 ID
record follows (01), 2 address records follow (02), and 2 borrower records follow (02)
```

ID section:

```
A 03 12345 AC Y
```

Address section:

```
A 01 01 Kristin Swanson 123 Elm Lane Little Falls, MN 02 Kristin
Swanson 345 Crownhart Hall Campus Dorm
```

Borrower Section (Kris Swanson is an undergrad registered for fall term so her expiration date is two weeks after semester ends)

```
A LIB50 01 20040115
A ALEPH 01 20040115
```

When a student drops, you will probably just update the Borrower section so instead of expiring in January, they expire on the current date. Or when a faculty member quits, you would change their expiration date to the current date as well.

Record structure for the PLIF input file

The PLIF input file contains one line per one patron record. Fields are strung together with nothing between them, and there is an end of line character at the end of each record. The line is composed of four sections:

USER Section (not repeatable) – Corresponds to the Z303 Oracle table. Contains information about the user that is only stored once – name, birth date, ILL library, etc. At the end of the user record section, the loader specifies how many ID, ADDRESS and BORROWER sections will follow.

- ID Section (repeatable) – Corresponds to the Z308 Oracle table. There can be up to 99 types of User IDs.
- ADDRESS Section (repeatable) – Add one section for each address. Corresponds to the Z304 table.
- BOR Section (repeatable) – Determines a user's borrowing privileges. Corresponds to the Z305 table.
- Lengths: USER: 1000 chars | ID : 100 chars | ADDR: 500 chars | BOR : 200 chars

The FILLER entries are to allow additional fields to be added without changing the structure of the input file.

USER SECTION:

Field name	Char/Num	Length	Notes
USER-REC-ACTION	CHAR	1	U, I, A, D or X (see Actions below)
USER-REC-MATCH-ID-TYPE	CHAR	2	01, 02, and so on. See note in Notes section below. Most commonly, this would be the SSN or university ID.
USER-REC-MATCH-ID**	CHAR	20	
FILLER	CHAR	40	For future use
USER-REC-VERIFICATION-1	CHAR	20	Verification for Z308 type 00; appears as "PINCODE" in GUI. Any letters need to be UPPERCASE.
FILLER	CHAR	40	For future use
USER-REC-NAME-TITLE	CHAR	10	
USER-REC-NAME	CHAR	200	For the alphabetized list in the Circulation client, in the form Last name, First name.
USER-REC-BIRTH-DATE	NUM	8	YYYYMMDD format.
USER-REC-BUDGET	CHAR	20	
USER-REC-EXPORT-CONSENT	CHAR	1	Set to "N" unless you are exporting records to other members of a consortium. ALEPH does not use; used only by user-written SQL.
USER-REC-DELINQ-	NUM	1	

Field name	Char/Num	Length	Notes
INDEX*			
USER-REC-DELINQ*	NUM	2	
USER-REC-DELINQ-N	CHAR	200	
USER-REC-FIELD-INDEX*	NUM	1	See note below.
USER-REC-FIELD*	CHAR	100	See note below.
FILLER	CHAR	100	
USER-REC-PROFILE	CHAR	12	
USER-REC-ILL-LIB	CHAR	5	Leave blank unless you are using the ILL module.
USER-REC-HOME-LIB	CHAR	5	
USER-REC-ILL-TOTAL-LIMIT	NUM	4	Zeroes, unless you are using the ILL module.
USER-REC-ILL-ACTIVE-LIMIT	NUM	4	Zeroes, unless you are using the ILL module.
USER-REC-SEND-ALL-LETT	CHAR	1	Y = all letters will be sent to user. (Normal setting is "Y".) N= Hold, photocopy and ILL requests will not be sent to user.
USER-REC-PROXY-FOR-ID	CHAR	12	
USER-REC-PRIMARY-ID	CHAR	12	
USER-REC-CON-LNG	CHAR	3	[New in v. 16.02] This corresponds to the user's language field (Z303-CON-LNG). For North American sites normal setting is "ENG"
FILLER	CHAR	170	
USER-REC-NO-ID	NUM	2	
USER-REC-NO-ADDRESS	NUM	2	
USER-REC-NO-BOR	NUM	2	

The last three fields of the USER section specify the number of additional sections following the user section on the same line (for example, five IDs, three addresses and one borrower record)

ID SECTION:

Field name	Char/ Numeric	Length	Notes
ID-REC-ACTION	CHAR	1	U, I, A, D or X (see actions below)
ID-TYPE	CHAR	2	Use this type to identify what to match on later. Can be 01 (barcode), 02, 03 --> 99. You should not include a type 00 ID.
ID-NO	CHAR	20	This is the ID itself (barcode, SSN, university ID, and so on). If this is blank, an ID will be automatically created if tab10 line 39= yes. Should not be padded with zeroes; should be left-aligned and right-padded with spaces.
ID-VERIFICATION	CHAR	20	A pincode or other secondary id; used primarily for Web access. Any

Field name	Char/ Numeric	Length	Notes
			letters need to be UPPERCASE.
ID-VERIFICATION-TYPE	CHAR	2	This should always be "00".
ID-STATUS	CHAR	2	AC = ACTIVE; NA = NOT ACTIVE
ID-ENCRYPTION	CHAR	1	Must be set to "Y" (for all new records)
FILLER	CHAR	52	For future use

ADDRESS SECTION:

Field name	Char/Num	Length	Notes
ADDR-REC-ACTION	CHAR	1	U, I, A, D or X (see actions below)
ADDR-REC-SEQUENCE	NUM	2	Increment for each section added. (The first address section would be 01, the second, 02, and so on.)
ADDR-REC-TYPE	NUM	2	Type of address (By convention, the permanent address is 01 and the campus/mailling address is 02)
ADDR-REC-ADDR-1	CHAR	50	First address line is name for notices – for example, John Smith
ADDR-REC-ADDR-2	CHAR	50	
ADDR-REC-ADDR-3	CHAR	50	
ADDR-REC-ADDR-4	CHAR	50	
ADDR-REC-ADDR-5	CHAR	50	
ADDR-REC-ZIP	CHAR	9	
FILLER	CHAR	1	
ADDR-REC-PHONE	CHAR	30	
ADDR-REC-PHONE-2	CHAR	30	
ADDR-REC-PHONE-3	CHAR	30	
ADDR-REC-PHONE-4	CHAR	30	
ADDR-REC-E-MAIL	CHAR	60	
ADDR-REC-START-DATE	NUM	8	YYYYMMDD format. When the Z304 address record created from this section will become active. (Note: xxx50/tab/tab/bor/address is an alternative way to control start and stop dates.)
ADDR-REC-STOP-DATE	NUM	8	YYYYMMDD format. When the Z304 address record created from this section will expire. (Note that the BOR-REC-EXPIRY-DATE controls when the borrower record will expire.)
FILLER	CHAR	39	

BOR SECTION:

Field name	Char/Numeric	Length	Notes
BOR-REC-ACTION	CHAR	1	U, I, A, D or X (see actions below)
BOR-REC-SUB-LIBRARY	CHAR	5	For North American libraries, usually the xxx50 ADM library code (but could also be "ALEPH" or a valid sublibrary code)
BOR-REC-TYPE	CHAR	2	the z305_bor_type
BOR-REC-STATUS	CHAR	2	the z305_bor_status
BOR-REC-EXPIRY-DATE	NUM	8	YYYYMMDD format. If blank, will take tab31 value
FILLER	CHAR	182	

Notes (on Record Structure):

- * = There are 3 note fields in the z303:
- 02 Z303-FIELD-1 PICTURE X(100).
 - 02 Z303-FIELD-2 PICTURE X(100).
 - 02 Z303-FIELD-3 PICTURE X(100).

If you specify "1" for the USER-REC-FIELD-INDEX, Z303-FIELD-1 will be updated;

If you specify "2", Z303-FIELD-2 will be updated;

If you specify "3", Z303-FIELD-3 will be updated.

(The same principle is used for the Delinquency field.)

** = The MATCH-ID is used **only** for matching – the loader program will not add this ID automatically. If you want to be able to match on an ID later (for updating purposes), you need to add the ID using the ID section. IDs of type 00 will be automatically created, because they are the unique ALEPH system number.

Records that are updated or inserted will take Borrower defaults from tab31, based on the Borrower Status.

Running p_file_20 from the command line

For the purpose of these instructions, assume that the input PLIF file is called "patrons.plif".

1. Copy the PLIF file to the \$data_scratch directory of the ADM library whose patrons you want to update (for example, USM50). The name of the file must be in lowercase. For example,

```
dlib USM50
ds
cp /wherever/patrons.plif .
```

2. Enter the following command to run p_file_20:

```
csh -f $aleph_proc/p_file_20  
USM50,patrons.plif,patron_update.rpt,Y,%,+ > & patron_update.log &
```

The job takes seven parameters, separated by commas.

- 1st - library to do patron update in (for example, USM50)
- 2nd - name of PLIF file, which should be in the \$data_scratch directory (for example, patrons.plif)
- 3rd - report file for update, which will be left in the \$data_scratch directory (for example, patron_update.rpt)
- 4th - Y to update the database, N to not change the database and only produce a report
- 5th - spaces character ("overwrite-with-blanks character"), if this character is in the input file, a space will be written to the ALEPH field
- 6th - ignore character ("null character"), if this character is in the input file, the ALEPH field will be protected.
- 7th - Note: In version 16 and later, there is an optional seventh parameter: Character conversion routine.

From the Version 16.02 Help (New Info added):

ALEPH requires that each patron Z303 record has a matching Z308 record with a type '00' (internal system ID) and a type '01' (barcode). Therefore, when patrons are loaded using this service:

1. If there is no Z308 type '00' reported in the input file, or already present in the record, the system generates a Z308 type '00' with KEY and VERIFICATION values the same as the Z303 ID. If only the verification is missing, it is generated with the same value as the Z308 KEY.
2. If there is no Z308 type '01' reported in the input file, or if it is already present in the record, the system generates a Z308 type '01' with KEY and VERIFICATION values the same as the Z308 type '00'. If only the verification is missing, it is generated with the same value as the Z308 KEY.

Workflow of the Job

- For each line of the input file (which contains all the relevant records for a specific patron):
 - Read all sections into ALEPH Znnn tables.
 - Find the Patron Internal ID, using the matching data given in the input.
 - Update/Insert Mode
 - In cases of "U"pdate mode, merge existing ALEPH Znnn table data with the input data (without updating the database yet.)

- In cases of "I"nsert mode, assign default values to the incoming Znnn record from ALEPH configuration tables.
- Check the validity of the desired actions:
 - Check each Znnn for record validity.
 - Check if the action flags set are consistent. For example, "D"eleting a global-user together with "I"nserting address, or "U"pdating a non-existent record are not valid sets of action flags.
 - Check permissions on actions (using tab100 permission flags and Z303-PLIF-MODIFICATION flags).
 - If no error is detected, update the database with the input data. In case of error, reject the input line and update the report-file.

Permissions - Additional Information

For permission checks the following variables in tab100 are considered:

USER-ADDR-PERMISSION

USER-PERMISSION

These permission variables can be used to limit the user permission for updating the patron's address and global records, by limiting permission to the ADM library of the patron's home library. This requires that the home library be registered in the global record. If either of the permission variables are inconsistent with the PLIF action, the input-line with all its records is rejected.

In addition, Z303-PLIF-MODIFICATION is used for blocking undesired modification. This block does not cause the input-line to be rejected; it simply causes it to be ignored for update purposes.

"D"eleting Users - Additional Information

When "D"eleting a patron, there is no need to specify all of the Znnn records to be deleted. Only the Z303 (the global-user information record) should be included. In other words, the input-line for the patron to be deleted should include only the matching data and the global user record (required only because it has the "D" flag in it). If other records are included in the input file they should appear with the action flag X or D, otherwise they will cause the input-line to be rejected.

Special checks are applied when deleting a patron, as follows:

6. "get_balance" - A patron cannot be deleted when his/her cash transactions are not balanced (that is, they are not equal to zero, meaning that there must not be a debt to the library or vice versa).

7. "get_transferred_balance" - patron cannot be deleted when his/her transferred cash transactions are not balanced (meaning, they have a debt to the library or vice versa, waiting to be handled by an accounts receivable agency).

8. "get_buf_z36" - A patron cannot be deleted if there are loan records.

"I"nserting Users - Additional Information

When a new record does not have the Z308 records that are required for a new patron (that is, the "00" type, which is the login-by-internal-id, and the "01" type, which is the login-by-barcode), tab100 DEFAULT-BOR-ID is consulted. If the value is "N", the "00" type record must be supplied in the input-file. If the value is "Y", the system will automatically generate the necessary Z308 records for the new patron. Note that the matching data is not considered login information, since it is lacking a verification field.

Barcodes

Each patron can create a hold on one barcode which is z308-key-type 01 (or login-rec-type 01). The system should not allow insertion of two barcodes, or deletion of existing barcodes since ALEPH assumes each patron has one and only one barcode. To change an existing barcode, the input file must contain the new barcode in "I"nsert mode. P-file-20 should recognize this action as "switch old barcode with the input-file barcode". In this case, if the input file does not contain any verification, the old verification (from the old barcode record) will be copied to the new record and the old record will be deleted completely. This deletion will be shown in the report as well.

Addresses

When updating an address, the input-file must contain the address-type. The system, assuming the input-file address contains an active address, should merge the input-record with the patron active address of the same address-type. In other words, the existing Z304 record is overwritten only if it is active and the input-record is active as well. Otherwise, the system considers the case as a "no-match", meaning it produces an error if the action is an "U"pdate or creates a new record if the action is "A"ppend.

Multi-ADM Environment

When using the "delete_bor_total" routine in a multi-ADM environment, the patron records will be deleted, but the global Z303 record will remain in the database if the current library is "shared" and there are local (Z305) records in the other shared library/ies.