USER DOCUMENTATION (ALEPHINO 5.0)

Patron Load Interface Format (PLIF)

The bridge to knowledge

TABLE OF CONTENTS

1 Introduction

This document describes the file format PLIF for the exchange of borrower data between Alephino and other programs.

PLIF stands for „Patron Load Interface Format“. Originally developed for Aleph 500 its structure is closely related to the internal representation of borrower data in Aleph. Hence many elements of the PLIF structure are not used in Alephino.

The character set used for PLIF is the ISO Latin 1 character set (ISO-8859-1).

2 Structure of the PLIF file

Alephino 5.0 does support PLIF based on the “classic” line-oriented plain text format and in addition XML format is available for both data export and import.

The data of a borrower in turn is subdivided in certain "sections“ called:

USER-REC
LINK-SECTION
LOGIN-REC
ADDRESS-REC
BOR-REC

The different sections contain 0 to n corresponding records; an exception is the USER-REC: it contains exactly one record. A PLIF borrower record therefore consists of one USER-REC plus LINK-SECTION, optionally followed by LOGIN-, ADDRESS- and BOR- records.

3 The PLIF plain text format

Such a sequential text file consists of lines each representing the data of one borrower:

<Data borrower 1> <nl>
<Data borrower 2> <nl>
<Data borrower 3> <nl>
...

The various sections or rather records are of fixed length. Positions of records and their fields within a line are fixed. That means in particular that:

- fields that are not filled in full length have to be filled up with blanks to their maximum length,
- record fields which are empty must be filled completely with blanks,
- field contents which are too long must be reduced to the maximum lenght,

to define the beginning of the next record at a specific position in the line. This does not apply to the last record in a borrower line: it may end after the last information and need not be filled up with blanks to a specific length.

For the different records you must meet the following lengths:

USER REC+LINK-SECTION	1000 characters
LOGIN-REC	100 characters
ADDRESS-REC	500 characters
BOR-REC	200 characters

For example a record for a borrower with 2 necessary LOGIN-Records, with 2 addresses and loan permissions for 3 sublibraries therefore would have a length of
1000 + 200 + 1000 + 600 = 2800 characters.

4 The PLIF XML format

A plain structuring and hence better readability in combination with simplified processing of borrower data is provided by PLIF data exchange in XML format. The subsequentially described logical structure of PLIF records is valid for both the text- as well as the XML format.

Pls. note that the naming of PLIF sections and their fields corresponds exactly to element names used for XML formatting (XML tag names). In contrast to the text format restrictions to field- and section lengths are meaningless. Empty fields or those marked as “not used” can be discarded completely.

An Alephino PLIF/XML file always needs to cover the following XML elements:

<PLIF-SET>	Root-Element
<UPDATE-BOR>	PLIF-Record (per user)
<USER-REC>...</USER-REC>	Master data of the user
<NO-ID-REC>x</NO-ID-REC>	Number of LOGIN-records (Identity records) per user
<NO-ADDR-REC>y</NO-ADDR-REC>	Number of ADDR-records (Addresses) per user
<NO-BOR-REC>z</NO-BOR-REC>	Number of BOR-records (Sub-library permissions) per user
<LOGIN-REC>...</LOGIN-REC> …	LOGIN-record (x times repeatable)
<ADDR-REC>...</ADDR-REC> ...	ADDR-record (y times repeatable)
<BOR-REC>...</BOR-REC> ...	BOR-record (z times repeatable)
</UPDATE-BOR> **...**	
</PLIF-SET>	

5 Structure of the PLIF records

For the structuring of the different record types into fields as well as their allocation to Alephino master files and fields please consult the following tables.
Fields that have not been labelled blue are irrelevant for Alephino but need to be populated with blanks for the sequential format.

5.1 USER Record

PLIF field	Type (length)	Alephino master file	Alephino field	Remark
USER-REC				Section tag for PLIF XML
USER-REC-ACTION	X(1)			Import: permitted: A,D,I,U,X
USER-REC-MATCH-ID-TYPE	X(2)			Import: permitted values: 00,01 Export: used value: 00
USER-REC-MATCH-ID	X(20)			Import: depends on the value of the field USER-REC-MATCH-ID-TYPE as IDN or barcode Export: IDN of the borrower record
FILLER (for future use)	X(100)			Not used
USER-REC-NAME-TITLE	X(10)	BEN	104 (Title)	
USER-REC-NAME	X(200)	BEN	102 (Name)	
USER-REC-BIRTH-DATE	N(8)	BEN	105 (Birthdate)	Format: YYYYMMDD
USER-REC-BUDGET	X(20)			Not used
USER-REC-EXPORT-CONSENT	X(1)			Not used
USER-REC-DELINQ-INDEX	N(1)			Import: permitted values: 1,2,3 Export: used value: 1
USER-REC-DELINQ	N(2)	BEN	112,114,116 (Delinquincy codes 123)	Import: depends on the value of the field USER-REC-DELINQ-INDEX Export: BEN 112 (Delinquincy code 1)
USER-REC-DELINQ-N	X(200)	BEN	113,115,117 (Delinquincy text 123)	Import: depends on the value of the field USER-REC-DELINQ-INDEX Export: BEN 113 (Delinquincy text 1)
USER-REC-FIELD-INDEX	N(1)			Import: permitted values: 1,2,3 Export: used value: 1
USER-REC-FIELD	X(200)	BEN	121,122,123 (Note 123)	Import: depends on the value of the field USER-REC-FIELD-INDEX Export: BEN 121 (Note 1)
USER-REC-PROFILE	X(10)			Not used
USER-REC-ILL-LIB	X(5)			Not used
USER-REC-HOME-LIB	X(5)	BEN	SUB (Standard sub-library)	
USER-REC-ILL-TOTAL-LIMIT	N(4)			Not used

USER-REC-ILL-ACTIVE-LIMIT	N(4)			Not used
USER-REC-SEND-ALL-LETT	X(1)			Not used
USER-REC-CON-LNG	X(3)	BEN	111 (Language)	
FILLER	X(196)			Not used

5.2 LINK Section

PLIF field	Type (length)	Alephino master file	Alephino field	Remark
NO-ID-REC	N(2)			Number of the following LOGIN Records
NO-ADDR-REC	N(2)			Number of the following address records
NO-BOR-REC	N(2)			Number of the following permission records

5.3 LOGIN Record

PLIF field	Type (length)	Alephino master file	Alephino field	Remark
LOGIN-REC				Section tag for PLIF XML
LOGIN-REC-ACTION	X(1)			Import: not used Export: like USER-REC-ACTION
LOGIN-REC-TYPE	X(2)			Import:permitted values: 00,01,02 Export: depends on the values of the following data fields
LOGIN-REC-NO	X(20)	BEN	001 (IDN) 100 (Barcode) 101 (Registration number)	depends on the value of ID-TYPE
LOGIN-REC-VERIFICATION	X(20)	BEN	108 (PinCode) 109 (n.a.) 110 (n.a.)	depends on the value of ID-TYPE
LOGIN-REC-VERIFICATION-TYPE	X(2)			Not used
LOGIN-REC-STATUS	X(2)			Not used
LOGIN-REC-ENCRYPTION	X(1)			Not used
LOGIN-REC-FILLER	X(52)			Not used

5.4 ADDRESS Record

PLIF field	Type (length)	Alephino master file	Alephino field	Remark
ADDR-REC				Section tag for PLIF XML
ADDR-REC-ACTION	X(1)			Import: permitted values: A,D,I,U,X Export: like USER-REC-ACTION
ADDR-REC-SEQUENCE	N(2)			Sequence no. of the address record
ADDR-REC-TYPE	N(2)	ADR	110	Import: permitted values: 1_,2_,3_ (numeric character + blanks)
ADDR-REC-ADDR-1	X(50)	ADR	100	
ADDR-REC-ADDR-2	X(50)	ADR	101	
ADDR-REC-ADDR-3	X(50)	ADR	102	
ADDR-REC-ADDR-4	X(50)	ADR	103	
ADDR-REC-ADDR-5	X(50)	ADR	104	
ADDR-REC-ZIP	X(10)	ADR	105	
ADDR-REC-PHONE	X(30)	ADR	107	
ADDR-REC-PHONE-2	X(30)	ADR	111	
ADDR-REC-PHONE-3	X(30)	ADR	112	
ADDR-REC-PHONE-4	X(30)	ADR	113	
ADDR-REC-E-MAIL	X(60)	ADR	106	
ADDR-REC-START-DATE	N(8)	ADR	108	
ADDR-REC-STOP-DATE	N(8)	ADR	109	
FILLER	X(39)			Not used

5.5 BOR Record

PLIF field	Type (length)	Alephino master file	Alephino field	Remark
BOR-REC				Section tag for PLIF XML
BOR-REC-ACTION	X(1)			Import: permitted values: A,D,I,U,X Export: like USER-REC-ACTION
BOR-REC-SUB-LIBRARY	X(5)	PRM	SUB	
BOR-REC-TYPE	X(2)	PRM	200	
BOR-REC-STATUS	X(2)	PRM	201	
BOR-REC-EXPIRY-DATE	N(8)	PRM	202	
BOR-REC-FILLER	X(182)			Not used

6 Explanations on specific fields

6.1 ACTION-Field

Each record begins with an ACTION field whose content defines which actions may be carried out with the record data when importing it into Alephino.

The following information is permitted for the ACTION field:

U	Update: Overwrite the fields of the Borrower/Address/Permission record that already exists in the Alephino database with the field contents from the corresponding fields of the PLIF file. (Field contents of other fields which do not appear in the PLIF record are preserved in the database record). If the record cannot be found in the Alephino database, no new record will be created; there will be only an error message.
I	Insert: Create a new Borrower/Address/Permission record. If there already is a matching record in the Alephino database the record will not be updated; there will be only an error message.
A	Combination of U and I : If the record cannot be found in the database, it will be created; if it already exists it will be updated.
D	Delete: Delete the record from the database.
X	The record will not be changed.

The records of a borrower record may have different values in their ACTION fields, this way it is possible to direct the actions very specifically.
Example: If you only want to delete the addresses of borrowers, the USER RECORDS are assigned ACTION ‚X' and the ADDRESS records ACTION ‚D'. The corresponding USER RECORDS are then used only to identify the borrower.

For USER RECORDS with ACTION ‚D' the system tries to delete the borrower completely, i.e including linked address, permissions and cash records. If there are still loans, requests, routing lists or unpaid fines for the borrower, the borrower record will not be deleted.

When exporting PLIF records from Alephino the value of the ACTION field may be specified as parameter when starting the program. This value then applies to all exported records.

6.2 MATCH-ID and MATCH-ID-TYPE (USER RECORD)

Import:
The content of MATCH-ID will not be imported into an Alephino field but is only used to identify a BEN record in the Alephino database, e.g. for the Update action. Via the content of MATCH-ID-TYPE you may control if the content of MATCH-ID should be interpreted as IDN or as barcode when searching for the BEN record:

MATCH-ID-TYPE	MATCH-ID
00	IDN
01	Barcode

Export:
In this place always the IDN will be imported. MATCH-ID-TYPE is accordingly 00.

6.3 Selecting the Delinquincy and Note fields

Import:
In Alephino it is possible to specify up to 3 (global) delinquincies. Furthermore you may fill in up to 3 note fields. In the PLIF records though, you may only transfer only one delinquincy and one note field. With the content of the field USER-REC-DELINQ-INDEX you define into which of the 3 Delinquincy fields the information from the PLIF record should be transferred:

USER-REC-DELINQ-INDEX	Delinquincy code	Delinquincy text
1	112	113
2	114	115
3	115	116

Accordingly, the content of USER-REC-FIELD-INDEX defines which Note field will be addressed:

USER-REC-FIELD-INDEX	Note field
1	121
2	122
3	123

(If you want fill in more than one Delinquincy and/or more than one Note field via the PLIF import, you need to import further PLIF records whose USER-RECORD addresses the Delinquincy resp. Note fields in question via the INDEX fields.)

Export:
Only the first Delinquincy fields (112, 113) and the first Note field (121) will be exported. The INDEX fields obtain accordingly the value 1.

7 Handling the LOGIN Records

In Alephino there is no separate masterfile for the LOGIN Records, the various ID and VERIFICATION fields are part of the BEN record. The Alephino PLIF therefore handles the LOGIN Records as follows:

Import:
Which fields in the BEN record will be filled with the information from the LOGIN Record depends on the value of the field ID-TYPE in the LOGIN Record. Permitted are the values 0,1,2. The following correlation applies between ID-TYPE values and the fields of the BEN record to fill them with the contents of the ID-NO and ID-VERIFICATION fields:

ID-Type	Content of ID-NO goes into BEN field	Content of ID-VERIFICATION goes into
0	001 (System ID)	108 (Pincode)
1	100 (Barcode)	109 (not used)
2	101 (Alternate ID)	110 (not used)

Export:
LOGIN Records will be created according to how the Alephino fields which are listed in the table above are filled.

8 Handling the BOR records

The BOR records of the PLIF record contain only few data that is loaded into the PRM records.
Not included e.g. is the permission flag for loans, requests etc. For the import the settings which are saved in the Alephino system as standard values for that combination (BOR-REC-SUB-LIBRARY, BOR-REC-STATUS) are taken. This happens not only when new PRM records are created (ACTION='I'); also when PRM records are updated (ACTION='U') via setting a flag you decide wether the permissions should be changed to the standard values or be preserverd.

The Alephino PRM record may also contain further fields (local delinquincies, local note fields), which are not allowed in PLIF and therefore cannot be imported or exported this way.

9 Ignore character

As in the PLIF file due to the fixed file format always all fields must be filled a specific indicator, the so-called Ignore character is neccessary when you want exclude specific fields from the import during the Update function (ACTION = „U"). To do so, the first position of these fields must be filled with the ignore character. Then, contents of these fields that possibly already exist in Alephino will be preserved. If the fields in the PLIF file are empty (= fields completely filled with blanks) the corresponding fields in the Alephino record will be deleted.
When creating new records (ACTION = „I") both empty fields and fields starting with the ignore character will not be considered.

Which character in the PLIF input file is used as ignore character may be entered as optional parameter when calling up the import (s.b.).

10 Calling up the program

The PLIF import and the PLIF export are both services of the Alephino Batch Program.

PLIF_EXP **Export borrower data in PLIF format**

Name	Meaning	Default	Values
LANGUAGE	Language	Select in Web Admin	GER/ENG/...
TYPE	Data format	Sequential	<empty>, XML
POOL	Name of the database	Select in Web Admin	
DATA	Name of the output file		.
ACTION	PLIF action indicator	I	I, U, A, D, X
FILTER	Selection criteria	No filter condition	
FROM	Ident number from (0 = from beginning)	0	
TO	Ident number to (0 = to the end)	0	

PLIF_IMP **Import borrower data in PLIF format**

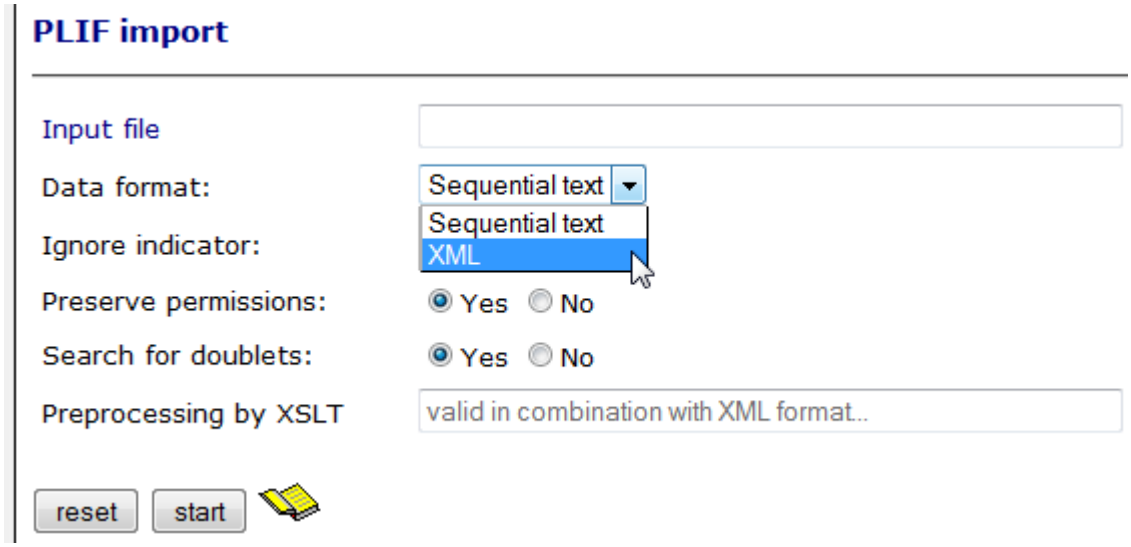
Name	Meaning	Default	Values
LANGUAGE	Language	Select in Web Admin	GER/ENG/...
TYPE	Data format	Seqential	<empty>, XML
POOL	Name of the database	Select in Web Admin	

DATA	Name of the intput file		
PRESERVE	Flag to define if the values of the permission flag will be preserved or changed to the standard values.	Y (= will be preserved)	Y/N
IGNORE	Character that causes that a PLIF field will be ignored	All fields will be imported	
DUBSRC	Search for doublets	Y = Reject doublets	Y/N

Error messages (import)

Message	ACTION	Remark
<Borrower name>: already exists	I	BEN record already exists
<BEN-Idn> - <Seq.-No.>: already exists	I	For that BEN record there is already an ADR record with that sequence number
<BEN-Idn> - <Subl>: already exists	I	For that BEN record there is already an PRM record for that sub-library
<Borrower name>: not found	U,D	BEN record not found
<BEN-Idn> - <Seq.-No.>: not found	U,D	ADR record with that sequence number not found
<BEN-Idn> - <Subl>: not found	U,D	PRM record for that sub-library not found
input formally wrong	all	Here, the program expects an ACTION information (i.e. A,D,I,U oder X); but it found another character
Unexpected end of input file	all	The program expects further information on the PLIF record.

PLIF data export and import can be handled using Alephino Web services / subsection interfaces:



Hints:

- Simply use „Unload patron data“ to get a PLIF file that can be used as a pattern for the creation of PLIF compliant borrower data.

- The "Preprocessing by XSLT" - option allows to load any data formatted in XML with a single step into Alephino. Prerequisite is provisioning of a XSL stylesheet which describes the conversion of the incoming XML data into the Alephino® PLIF/XML format.