



BENUTZERDOKUMENTATION (ALEPHINO 5.0)

Patron Load Interface Format (PLIF)



INHALT

1 EINFÜHRUNG.....	3
2 DER AUFBAU DER PLIF-DATEI.....	3
2.1 DAS PLIF TEXTFORMAT.....	3
2.2 DAS PLIF XML-FORMAT.....	4
3 STRUKTUREN DER PLIF-RECORDS.....	5
3.1 USER RECORD.....	5
3.2 LINK-SECTION.....	6
3.3 LOGIN RECORD.....	6
3.4 ADDRESS RECORD	6
3.5 BOR RECORD.....	7
4 ERLÄUTERUNGEN ZU BESTIMMTEN FELDERN.....	7
4.1 ACTION-FELD.....	7
4.2 MATCH-ID UND MATCH-ID-TYPE (USER RECORD).....	8
4.3 AUSWAHL DER SPERRGRUND- UND NOTIZ-FELDER.....	8
5 BEHANDLUNG DER LOGIN RECORDS.....	9
6 BEHANDLUNG DER BOR-RECORDS.....	9
7 IGNORE-ZEICHEN.....	9
8 PROGRAMM-AUFRUF.....	10

1 Einführung

Dieses Dokument beschreibt das Dateiformat PLIF für den Austausch von Leserdaten zwischen Alephino und Fremdsystemen.

PLIF steht für „Patron Load Interface Format“. Es handelt sich dabei um eine spezielle Datenstruktur, die einst für Aleph 500 geschaffen wurde und sich an der internen Repräsentation von Leserdaten in diesem System orientiert. Daher umfaßt PLIF eine Vielzahl von Datenelementen, die in Alephino bedeutungslos sind.

Der für PLIF verwendete Zeichensatz ist ISO-Latin-1 (ISO-8859-1).

2 Der Aufbau der PLIF-Datei

In Alephino 4.1 ist PLIF sowohl auf Basis des „klassischen“ zeilenorientierten Textformates als mittels XML-Format realisiert.

Die Daten eines Lesers sind unterteilt in bestimmte "Sektionen" genannt:

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

Die verschiedenen Sektionen enthalten 0 bis n entsprechende Datensätze (Records); Ausnahme bildet USER-REC, den es genau einmal gibt. Ein PLIF-Leser-Satz besteht demnach mindestens aus genau einem USER-REC und der LINK-SECTION, optional gefolgt von LOGIN-, ADDRESS- und BOR- Records.

2.1 Das PLIF Textformat

Eine solche sequenzielle Text-Datei besteht aus Zeilen („Leser-Sätzen“); für die Daten jeweils eines Lesers wird eine Zeile verwendet:

```
<Daten Leser 1> <nl>  
<Daten Leser 2> <nl>  
<Daten Leser 3> <nl>  
...
```

Die einzelnen Sektionen bzw. Records besitzen eine feste Länge. Positionen der einzelnen Sätze und deren Felder innerhalb einer Leserdaten-Zeile sind festgelegt. Das bedeutet im Einzelnen:

- nicht in voller Länge belegte Felder sind bis zu ihrer Maximallänge mit Leerzeichen aufzufüllen,
- nicht belegte Felder sind komplett mit Leerzeichen zu füllen,
- zu lange Feldinhalte müssen auf die Maximallänge gekürzt werden,

um den Beginn des nächsten Records an einer bestimmten Position in der Zeile festzulegen. Eine Ausnahme gibt es lediglich für den letzten Record in einer

Benutzerzeile: dieser kann bereits nach der letzten Nutzerinformation enden und muss nicht mit rechts angefügten Leerzeichen auf eine bestimmte Länge gebracht werden.

Folgende Längen sind für die verschiedenen Records einzuhalten:

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

Ein Satz für einen Leser mit 2 LOGIN-Records, 2 Adressen und Ausleihberechtigungen für 3 Zweigstellen hätte also eine Länge von $1000 + 200 + 1000 + 600 = 2800$ Zeichen.

2.2 Das PLIF XML-Format

Eine klare Strukturierung und somit bessere Lesbarkeit sowie einfachere Verarbeitung der Leserdaten ist mittels XML realisiert. Die im folgenden Kapitel beschriebene logische Struktur der PLIF-Datensätze ist hierbei unabhängig vom verwendeten Dateiformat.

Die Benennung der PLIF-Sektionen und deren Felder entspricht genau den für das XML-Format verwendeten Element-Namen (XML-Tags). Im Unterschied zum Textformat sind Längenangaben hier nicht von Bedeutung. Leere oder nachfolgend als "nicht verwendet" gekennzeichnete Felder können entfallen.

Eine PLIF/XML-Datei umfaßt stets folgende Ebenen bzw. Sektionen:

<PLIF-SET>	Root-Element
<UPDATE-BOR>	PLIF-Datensatz (pro Benutzer)
<USER-REC>...</USER-REC>	Benutzer-Stammdaten
<NO-ID-REC>x</NO-ID-REC>	Anzahl LOGIN-Sätze (Identifikation) pro Benutzer
<NO-ADDR-REC>y</NO-ADDR-REC>	Anzahl ADDR-Sätze (Adressen) pro Benutzer
<NO-BOR-REC>z</NO-BOR-REC>	Anzahl BOR-Sätze (Zweigstellen-Berechtigungen) pro Benutzer
<LOGIN-REC>...</LOGIN-REC> ...	LOGIN-Satz (x * wiederholbar)
<ADDR-REC>...</ADDR-REC> ...	ADDR-Satz (y * wiederholbar)
<BOR-REC>...</BOR-REC> ...	BOR-Satz (z * wiederholbar)
</UPDATE-BOR> ...	
</PLIF-SET>	

3 Strukturen der PLIF-Records

Die Strukturierung der verschiedenen Record-Typen in Felder sowie deren Zuordnung zu Alephino-Stammdateien und –Feldern kann den folgenden Tabellen entnommen werden. Nicht blau markierte Felder sind für Alephino bedeutungslos, müssen jedoch im sequenziellen Datenformat mit Leerstellen belegt werden.

3.1 USER Record

PLIF Feld	Typ (Länge)	Alephino Datei	Alephino Feld	Bemerkung
USER-REC				Section-Tag für PLIF-XML
USER-REC-ACTION	X(1)			Import: erlaubt: A,D,I,U,X
USER-REC-MATCH-ID-TYPE	X(2)			Import: erlaubt: 00,01,02 Export: benutzt: 00
USER-REC-MATCH-ID	X(20)			Import: abhängig vom Wert des Feldes USER-REC-MATCH-ID-TYPE als IDN, Barcode oder Matrikelnummer Export: IDN des Benutzersatzes
USER-REC-FILLER	X(100)			Wird nicht verwendet
USER-REC-NAME-TITLE	X(10)	BEN	104 (Titel)	
USER-REC-NAME	X(200)	BEN	102 (Name)	
USER-REC-BIRTH-DATE	N(8)	BEN	105 (Geb.datum)	Format: YYYYMMDD
USER-REC-BUDGET	X(20)			Wird nicht verwendet
USER-REC-EXPORT-CONSENT	X(1)			Wird nicht verwendet
USER-REC-DELINQ-INDEX	N(1)			Import: erlaubte Werte: 1,2,3 Export: benutzter Wert: 1
USER-REC-DELINQ	N(2)	BEN	112,114,116 (Sperrgrund-Codes 123)	Import: abhängig vom Wert des Feldes USER-REC-DELINQ-INDEX Export: BEN 112 (Sperrgrund-Code 1)
USER-REC-DELINQ-N	X(200)	BEN	113,115,117 (Sperrgrund-Text 123)	Import: abhängig vom Wert des Feldes USER-REC-DELINQ-INDEX Export: BEN 113 (Sperrgrund-Text 1)
USER-REC-FIELD-INDEX	N(1)			Import: erlaubte Werte: 1,2,3 Export: benutzter Wert: 1
USER-REC-FIELD	X(200)	BEN	121,122,123 (Notiz 123)	Import: abhängig vom Wert des Feldes USER-REC-FIELD-INDEX Export: BEN 121 (Notiz 1)
USER-REC-PROFILE	X(10)			Wird nicht verwendet
USER-REC-ILL-LIB	X(5)			Wird nicht verwendet
USER-REC-HOME-LIB	X(5)	BEN	SUB (Standard-Zweigstelle)	
USER-REC-ILL-TOTAL-LIMIT	N(4)			Wird nicht verwendet
USER-REC-ILL-ACTIVE-LIMIT	N(4)			Wird nicht verwendet
USER-REC-SEND-ALL-LETT	X(1)			Wird nicht verwendet
USER-REC-CON-LNG	X(3)	BEN	111	

			(Sprache)	
USER-REC-FILLER	X(196)			Wird nicht verwendet

3.2 LINK-SECTION

PLIF Feld	Typ (Länge)	Alephino Datei	Alephino Feld	Bemerkung
NO-ID-REC	N(2)			Anzahl der folgenden LOGIN-Records
NO-ADDR-REC	N(2)			Anzahl der folgenden Address-Records
NO-BOR-REC	N(2)			Anzahl der folgenden Permission-Records

3.3 LOGIN Record

PLIF Feld	Typ (Länge)	Alephino Datei	Alephino Feld	Bemerkung
LOGIN-REC				Section-Tag für PLIF-XML
LOGIN-REC-ACTION	X(1)			Import: nicht verwendet Export: wie USER-REC-ACTION
LOGIN-REC-TYPE	X(2)			Import: erlaubt: 00,01,02 Export: je nach Wert der folgenden Datenfelder
LOGIN-REC-NO	X(20)	BEN	001 (IDN) 100 (Strichcode) 101 (Matrikelnr.)	Je nach Wert von LOGIN-TYPE
LOGIN-REC-VERIFICATION	X(20)	BEN	108 (PinCode) 109 (nicht verw.) 110 (nicht verw.)	Je nach Wert von LOGIN-TYPE
LOGIN-REC-VERIFICATION-TYPE	X(2)			Wird nicht verwendet
LOGIN-REC-STATUS	X(2)			Wird nicht verwendet
LOGIN-REC-ENCRYPTION	X(1)			Wird nicht verwendet
LOGIN-REC-FILLER	X(52)			Wird nicht verwendet

3.4 ADDRESS Record

PLIF Feld	Typ (Länge)	Alephino Datei	Alephino Feld	Bemerkung
ADDR-REC				Section-Tag für PLIF XML
ADDR-REC-ACTION	X(1)			Import: erlaubt: A,D,I,U,X Export: wie USER-REC-ACTION
ADDR-REC-SEQUENCE	N(2)			Sequenznr. des Address-Record
ADDR-REC-TYPE	N(2)	ADR	110	Import: erlaubt: 1_,2_,3_ (Ziffer + Leerzeichen)
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	
ADDR-REC-FILLER	X(39)			Wird nicht verwendet

3.5 BOR Record

PLIF Feld	Typ (Länge)	Alephino Datei	Alephino Feld	Bemerkung
BOR-REC				Section-Tag für PLIF-XML
BOR-REC-ACTION	X(1)			Import: erlaubt: A,D,I,U,X Export: wie 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)			Wird nicht verwendet

4 Erläuterungen zu bestimmten Feldern

4.1 ACTION-Feld

Jeder Record beginnt mit einem ACTION-Feld, dessen Inhalt bestimmt, welche Aktionen mit den Record-Daten beim Import in Alephino durchgeführt werden.

Folgende Angaben sind zulässig für das ACTION-Feld:

U	Update: Überschreibe die Felder des im Alephino-Pool vorhandenen Leser-/Adress-/Berechtigungs-Satzes mit den Feldinhalten für die entsprechenden Felder aus der PLIF-Datei. (Feldinhalte von weiteren, nicht vom PLIF-Satz vorkommenden Feldern des Pool-Satzes bleiben erhalten.) Kann der Satz im Alephino-Pool nicht gefunden werden, wird kein neuer Satz angelegt; es wird lediglich eine Fehler-Meldung erzeugt.
I	Insert: Neuanlage eines Ben-/Adr-/Prm-Satzes. Ist ein passender Satz bereits im Alephino-Pool vorhanden, findet kein Update des Satzes statt; es wird lediglich eine Fehlermeldung erzeugt.
A	Kombination aus U und I : Wird der Satz im Pool nicht gefunden, so wird er angelegt; ist er dagegen schon im Pool vorhanden, findet ein Update statt
D	Delete: Löscht den Satz aus dem Pool.
X	Der Satz wird nicht verändert.

Die Records eines Leser-Satzes können eine unterschiedliche Belegung des ACTION-Feldes haben, so dass eine gezieltere Steuerung der durchzuführenden Aktion möglich ist.

Beispiel: Sollen nur die Adressen von Lesern gelöscht werden, so werden die USER Records mit der ACTION ‚X‘ versehen, die ADDRESS Records mit der ACTION ‚D‘. Die jeweiligen USER Records dienen dann nur zur Identifikation der Leser.

Bei mit der ACTION ‚D‘ versehenen USER Records wird versucht, den Leser komplett, d.h mit verknüpften Adress-, Berechtigungs- und Gebühren-Sätzen zu löschen. Sollten für den Leser noch Ausleihen, Vormerkungen, Umlauflisten und nicht bezahlte Gebühren vorliegen, wird die Löschung nicht durchgeführt.

Beim Export von PLIF-Sätzen aus Alephino kann die Belegung des ACTION-Feldes als Parameter beim Programmstart angegeben werden. Sie gilt dann für alle exportierten Records.

4.2 MATCH-ID und MATCH-ID-TYPE (USER Record)

Import:

Der Inhalt von MATCH-ID wird nicht in ein Alephino-Feld importiert, sondern dient lediglich zur Identifikation eines BEN-Satzes im Alephino-Pool, etwa für die Update-Aktion. Über den Inhalt von MATCH-ID-TYPE kann man steuern, ob der Inhalt von MATCH-ID bei der Suchanfrage nach dem BEN-Satz als IDN oder als Barcode interpretiert werden soll:

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

Export:

Es wird an dieser Stelle immer die IDN exportiert. MATCH-ID-TYPE ist entsprechend 00.

4.3 Auswahl der Sperrgrund- und Notiz-Felder

Import:

In Alephino ist es möglich, bis zu 3 (globale) Sperrgründe anzugeben. Zudem lassen sich 3 Notizfelder belegen. Im PLIF-Satz besteht jedoch nur die Möglichkeit, jeweils einen Sperrgrund und ein Notizfeld zu übermitteln. In welches der 3 Sperrgrundfelder die Information aus dem PLIF-Satz übernommen werden soll, wird durch den Inhalt des Feldes USER-REC-DELINQ-INDEX festgelegt:

USER-REC-DELINQ-INDEX	Sperrgrund-Code	Sperrgrund-Text
1	112	113
2	114	115
3	115	116

Entsprechend legt der Inhalt von USER-REC-FIELD-INDEX fest, welches Notizfeld angesprochen wird:

USER-REC-FIELD-INDEX	Notizfeld
1	121
2	122
3	123

(Will man mittels PLIF-Import mehr als einen Sperrgrund und/oder mehr als ein Notizfeld belegen, muss man weitere PLIF-Sätze „nachschieben“, deren USER-Record die entsprechenden Sperrgrund- bzw. Notizfelder über die INDEX-Felder anspricht.)

Export:

Es werden nur die ersten Sperrgrund-Felder (112, 113) und das erste Notizfeld (121) exportiert. Die INDEX-Felder erhalten entsprechend den Wert 1.

5 Behandlung der LOGIN Records

In Alephino gibt es keine eigene Stammdatei für die Ids, die verschiedenen ID- und VERIFICATION-Felder sind Bestandteil des BEN-Satzes. Beim Alephino-PLIF wird daher wie folgt mit den LOGIN-Records verfahren:

Import:

Welche Felder im BEN-Satz mit den Angaben im LOGIN-Record belegt werden, hängt vom Wert des LOGIN-TYPE-Feldes im LOGIN-Record ab. Zulässig sind dabei die Werte 0,1,2. Folgende Zuordnung besteht zwischen den LOGIN-TYPE-Werten und den Feldern des BEN-Satzes, um sie mit den Inhalten der LOGIN-NO- und LOGIN-VERIFICATION-Felder zu belegen:

LOGIN-Type	Inhalt von LOGIN-NO geht in BEN-Feld	Inhalt von LOGIN-VERIFICATION geht in
0	001 (System ID)	108 (Pin-Code)
1	100 (Barcode)	109 (nicht verwendet)
2	101 (Matrikelnr.)	110 (nicht verwendet)

Export:

LOGIN Records werden entsprechend der Belegung der in der Tabelle angegebenen Alephino-Felder erzeugt.

6 Behandlung der BOR-Records

Die BOR-Records des PLIF-Satzes enthalten nur einige Daten, die in die PRM-Sätze geladen werden.

Nicht enthalten sind beispielsweise die Berechtigungs-Schalter für Ausleihe, Vormerkung etc. Beim Import werden hierfür die Einstellungen ermittelt, die für das Paar (BOR-REC-SUB-LIBRARY, BOR-REC-STATUS) als Standardwerte im Alephino-System gespeichert sind. Dies geschieht nicht nur bei der Neuanlage von PRM-Sätzen (ACTION='I'), für Update der PRM-Sätze (ACTION='U') wird per Belegung eines Schalters entschieden, ob die Berechtigungen auf die Standardwerte gesetzt oder beibehalten werden sollen.

Der Alephino-PRM-Satz kann zudem weitere Felder enthalten (lokale Sperrgründe, lokale Notizfelder), die im PLIF nicht vorgesehen sind und somit auf diesem Wege nicht importiert oder exportiert werden können.

7 Ignore-Zeichen

Da in der PLIF-Datei wegen des festen Dateiformats immer alle Felder angegeben werden müssen, braucht man ein spezielles Kennzeichen, das sog. Ignore-Zeichen, wenn man bei der Update-Function (ACTION = „U“) bestimmte Felder vom Import

ausschließen will. Dazu muss die erste Position dieser Felder mit dem Ignore-Zeichen belegt sein. Im Alephino eventuell schon vorhandene Belegungen der entsprechenden Felder bleiben dann erhalten. Leere Felder in der PLIF-Datei (Felder vollständig mit Leerzeichen gefüllt) bewirken hingegen ein Löschen der entsprechenden Felder im Alephino-Satz.

Bei der Neuanlage von Sätzen (ACTION = „I“) werden sowohl leere als auch mit dem Ignore-Zeichen versehene Felder nicht berücksichtigt.

Welches Zeichen in der PLIF-Eingabedatei als Ignore-Zeichen Verwendung findet, wird bei Aufruf des Imports als optionaler Parameter angegeben (s.u.).

8 Programm-Aufruf

Der PLIF-Import und der PLIF-Export sind als Dienste des Alephino-Batch-Programms realisiert.

PLIF_EXP Export Benutzerdaten im PLIF-Format

Name	Bedeutung	Default	Werte
LANGUAGE	Sprache	Auswahl in Web-Admin	GER/ENG/...
TYPE	Datenformat	Sequentiell	<leer>, XML
POOL	Name des Datenpools	Auswahl in Web-Admin	
DATA	Name der Ausgabedatei		
ACTION	PLIF-Aktions-Indikator	I	I, U, A, D, X
FILTER	Auswahlkriterium	Keine Filterbedingung	
FROM	Identnummer ab (0 = ab Beginn)	0	
TO	Identnummer bis (0 = bis Ende)	0	

PLIF_IMP Import Benutzerdaten im PLIF-Format

Name	Bedeutung	Default	Werte
LANGUAGE	Sprache	Auswahl in Web-Admin	GER/ENG/...
TYPE	Datenformat	Sequentiell	<leer>, XML
POOL	Name des Datenpools	Auswahl in Web-Admin	
DATA	Name der Eingabedatei		
PRESERVE	Schalter, der festlegt, ob die Belegung der Berechtigungsschalter erhalten bleibt oder auf Standardwerte zurückgesetzt wird.	Y (= bleibt erhalten)	Y/N
IGNORE	Zeichen, das ein Ignorieren des PLIF-Feldes bewirkt	Alle Felder werden übernommen	
DUBSRC	Suche nach Dubletten	Y = Dubletten abweisen	Y/N

Fehlermeldungen (Import)

Meldung	ACTION	Bemerkung
<Lesername>: Satz schon vorhanden	I	BEN-Satz existiert bereits

<BEN-Idn> - <Seq-Nr.>: Satz schon vorhanden	I	Zum BEN-Satz existiert bereits ein ADR-Satz mit der Sequenz-Nummer
<BEN-Idn> - <Subl>: Satz schon vorhanden	I	Zum BEN-Satz existiert bereits ein PRM-Satz für die Zweigstelle
<Lesername>: Satz nicht gefunden	U,D	BEN-Satz nicht gefunden
<BEN-Idn> - <Seq-Nr.>: Satz nicht gefunden	U,D	ADR-Satz mit der Sequenz-Nummer nicht gefunden
<BEN-Idn> - <Subl>: Satz nicht gefunden	U,D	PRM-Satz für die Zweigstelle nicht gefunden
Formatfehler Eingabedatei	alle	An dieser Stelle wird vom Programm eine ACTION-Angabe erwartet (also A,D,I,U oder X); gefunden wurde jedoch ein anderes Zeichen
Unerwartetes Ende der Eingabedatei	alle	Vom Programm werden noch weitere Angaben zum PLIF-Satz erwartet.

Die Daten können über die WEB Services / Schnittstellen / Laden Benutzer geladen werden:

PLIF-Import

Ladedatei: * Pflichtfeld

Datenformat:

Ignore-Kennzeichen:

Erhalten Berechtigungen: Ja Nein

Suche nach Dubletten: Ja Nein

Vorverarbeitung mit XSLT:



Die Option „Vorverarbeitung mit XSLT“ erlaubt es, beliebige in XML formatierte Personendaten mit einem Arbeitsschritt in Alephino zu laden. Voraussetzung ist die Bereitstellung eines XSL-Stylesheets, welches die Konvertierung des Fremd- in das Alephino-PLIF/XML-Format beschreibt.

Tipp:

Erzeugen Sie per „Entladen Benutzer“ eine repräsentative PLIF-Datei, die Ihnen als Vorlage für die Erzeugung von PLIF-konformen Leserdaten dienen kann.