



USER DOCUMENTATION

# How to Load Records using the Generic Vendor Records Loader (file-96)

---

**Ex Libris**

© Ex Libris Ltd., 2002, 2004

Release 17

Last Update: November 14, 2004

# Table of Contents

<b>1</b>	<b>OVERVIEW .....</b>	<b>3</b>
<b>2</b>	<b>SPECIFYING OPTIONS IN TAB_96_DEF .....</b>	<b>4</b>
<b>3</b>	<b>SPECIFYING THE ORDER SEQUENCE .....</b>	<b>5</b>
<b>4</b>	<b>TAB_MATCH SECTION .....</b>	<b>6</b>
<b>5</b>	<b>CREATING THE FIX ROUTINE, AND SPECIFYING THE FIX ROUTINE IN TAB_FIX.....</b>	<b>6</b>
<b>6</b>	<b>INSTRUCTIONS FOR RUNNING THE GENERIC VENDOR RECORDS LOADER FROM THE CATALOGING MODULE SERVICES MENU .....</b>	<b>7</b>
<b>7</b>	<b>INSTRUCTIONS FOR RUNNING THE GENERIC VENDOR RECORDS LOADER FROM THE UNIX PROMPT .....</b>	<b>8</b>
<b>8</b>	<b>THE P_FILE_96 ERROR LOG .....</b>	<b>9</b>

# 1 Overview

The Generic Vendor Records Loader (p\_file\_96) batch service loads MARC records, creates associated holdings, item, order, and budget transaction records, and encumbers the appropriate fund.

The input file sent from vendors is a standard MARC record with extra fields:

049 \$\$a	-	contains the OCLC codes for local library's locations.
049 \$\$b	-	contains the item barcode (not required)
980 \$\$a	-	order date (YYYYMMDD)
980 \$\$f	-	invoice # (see below)
980 \$\$g	-	number of copies
980 \$\$b	-	list price
980 \$\$e	-	net price
980 \$\$c	-	currency
981 \$\$b	-	fund code

If there is only a 980 \$b (user-defined) field, then an encumbrance is created with the 980 \$b field. Otherwise, the encumbrance is created with the 980 \$e (user-defined) field, and the 980 \$b field is not used. Additionally, the loader program accepts prices in two forms: with or without two decimal places (for example 5995 or 59.95).

ALEPH checks for the existence of a decimal point. The default values in the default values input file (e.g. tab\_96\_def) for the Z601-ORIGINAL-SUM and the Z601-ACTIVE-SUM fields also accept values with or without decimals.

The currency code is taken from the 980 field's \$c when creating Z68 and Z601 records. If a \$c subfield is missing, the batch uses the default currency line in the default values input file.

The 981 \$d field maps to Z68-LIBRARY-NOTE, and the 980 \$f field maps to the Z68-VENDOR-REFERENCE-NO in the Z68 (acquisition order records) table. In addition, the current date is assigned to the Z68-ORDER-STATUS-DATE.

## Note:

There is no check in p-file-96 for the balance of the budget. This means that even if the budget is over-encumbered, the system will still encumber the order.

To load records with the service, carry out the following procedures:

1. Specify options in the user-defined table for default values (for example, tab\_96\_def).
2. Specifying the Order Sequence.
3. Create the fix routine, and specify the fix routine in tab\_fix.
4. Run the loader from the Load Catalog Records option of the Services menu in the Cataloging module or from the Unix prompt.
5. Carry out multiple test runs.

These procedures are explained in this document.

## 2 Specifying options in tab\_96\_def

You must specify the following default values in a user-defined configuration table in the tab/import directory of your bibliographic library. The tab\_96\_def table is a sample table of this file (XXX01/data\_tab/import/tab\_96\_def). For SUB-LIBRARY and COLLECTION, the options are the values from 049 \$\$a, in this example, XZLA and XZLL. Thus records with a holdings code of XZLA are assigned the XLINC sublibrary, and records with a holdings code of XZLL are assigned the XLAW sublibrary. Following is a sample of the table:

CATALOGER	#####	YBPSR
ORDER-SEQ	#####	mono
Z30-SUB-LIBRARY	XZLA	XLINC
Z30-SUB-LIBRARY	XZLL	XLAW
Z30-COLLECTION	XZLA	GEN
Z30-COLLECTION	XZLL	GEN
Z30-MATERIAL	#####	BOOK
Z30-ITEM-STATUS	#####	01
Z30-CATALOGER	#####	YBPSR
Z30-CALL-NO-TYPE	#####	1
Z30-ITEM-PROCESS-STATUS	#####	OR
Z68-SUB-LIBRARY	XZLA	XLINC
Z68-SUB-LIBRARY	XZLL	XLAW
Z68-ORDER-TYPE	#####	M
Z68-ORDER-STATUS	#####	SV
Z68-ARRIVAL-STATUS	#####	No
Z68-VENDOR-CODE	#####	YBP
Z68-RUSH	#####	N
Z68-DELIVERY-TYPE	#####	S
Z68-TARGET-DIRECT	#####	N
Z68-E-CURRENCY	#####	USD
Z68-METHOD-OF-ACQUISITION	#####	P
Z68-ORDER-DELIVERY-TYPE	#####	LI
Z68-MATERIAL-TYPE	#####	M
Z68-INVOICE-STATUS	#####	P
Z601-VENDOR-CODE	#####	YBP
Z601-USER-NAME	#####	LOAD
z601-CURRENCY	#####	USD
z601-TYPE	#####	ENC

### 3 Specifying the Order Sequence

From the ADM library (XXX50), go into UTIL/G/2 (Display/Update library parameters), and create or identify an existing order number sequence to use for this group of orders. If no sequence is specified, the next order number is the next in line for the global order number. To have all of the orders created from the service follow the same sequence, specify the sequence with the line:

```
ORDER-SEQ                ##### mono
```

The above order sequence corresponds to the following line in UTIL/G/2:

```
22. last-order-no-mono    12      y      S  M
```

When the above sequence is in use, all order numbers are prefixed with “M” – for example, M345.

## 4 tab\_match section

You must verify that the `tab_match` table includes the F96 match routine, the program name and the program arguments. It is located in the XXX01 library in the `tab` directory.

```
! 1                2                3
!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
F96  match_doc_uid                I-ISBN
RLIN  match_doc_uid                T-020
OCLC  match_doc_script              tab_match_script_oclc
MRCV  match_doc_uid                I-909
CAT   match_doc_uid                I-ISBN
CAT   match_doc_acc                tab_match_acc
P36   match_doc_acc                tab_match_acc
```

### Key to tab\_match

- ❑ Column 1 is the name of the match routine code. For loading records from vendors the match routine code is F96.
- ❑ Column 2 lists the match program used by the match routine. The program used by the F96 match routine is `match_doc_uid`.
- ❑ Column 3 indicates the program arguments. In this case the argument is `I-ISBN`.

## 5 Creating the fix routine, and specifying the fix routine in tab\_fix

The `96_add_own.fix` table (in the `xxx01/data_tab/import` directory) specifies the OWN field for records. All incoming records should have an OWN field. This table also moves the 001 value to the 035 value. This fix can have any name, but it must be included in the `tab_fix` section specified in the command line (see below).

The `96_add_own.fix` table looks like this:

```
1 LDR                ADD-FIELD                OWN  ,L,$$aOL
1 001##              CHANGE-FIELD            035
1 035##              CHANGE-SUBFIELD         a
```

The first line adds the OWN field with “OL” in subfield a. The last two lines move the YBP system number from 001 to 035 \$\$a.

Add the following line in `tab_fix` in the `XXX01/data_tab` directory:

```
F96    fix_doc_do_file_08                96_add_own.fix
```

## 6 Instructions for running the Generic Vendor Records Loader from the Cataloging Module Services Menu

The Generic Vendor Records Loader is run from the Services menu of the Cataloging module and is listed as Generic Vendor Records Loader or (p\_file\_96) under the Load Catalog Records sub-menu.

### Notes:

1. If the Input file is located in the \$data\_scratch directory, simply enter the name of the file.
2. If the input file is located in another directory, enter the path relative to the Scratch directory, or enter the full path name.
  - Example of a relative path: `../source/filename`
  - Example of a full path:  
`/aleph/a50_5/usm01/source/filename`

Enter the day and hour you want the procedure to run. The system uses the 24-hour clock, so if you want to run the procedure at 11 P.M., enter "23".

3. If the Create Items flag is set to Yes, the loader will always create one item only, even if the number of copies in the 980 field, subfield \$\$g, is greater than one.

## 7 Instructions for running the Generic Vendor Records Loader from the Unix prompt

Enter the following command line from XXX01 (BIB library):

```
csch -f $aleph_proc/p_file_96 USM01,96.mrc,tab_96_def,  
USM60,USM50,Y,N,Y,F96,Y
```

Note that line contains the following parameters:

- [1]: BIB library (USM01 in example)
- [2]: MARC input file. The script converts it into ALEPH sequential (96.mrc)
- [3]: defaults table - should be located in the bib data\_tab directory. (tab\_96\_def)
- [4]: HOL library - if blank then no HOL records are created. (USM60)
- [5]: ADM library - if blank then no z30 (item/adm) records are created (USM50)
- [6]: create Z30 Y/N (Item) (Y). If the Create Items flag is set to Y, the loader will always create one item only, even if the number of copies in the 980 field, subfield \$\$g, is greater than one.
- [7]: create Z68 Y/N (Order) (N)
- [8]: create Z601 Y/N (Encumbrance) (Y)
- [9]: Name of tab\_fix section to run before loading bib records (F96)
- [10]: Update database Y/N (enter N to do a test run) (Y)

To export the log to the data\_scratch directory (as opposed to the log displaying on-screen), add a ">" specification. For example:

```
csch -f $aleph_proc/p_file_96 BCT01,11100109.MRC,  
tab_96_def,,BCT50,Y,Y,Y,F96,Y & p_file_96.log.0124 &
```



## 8 The p\_file\_96 Error Log

You may see messages like this in the error log:

```
Error: problems in Z68 record:  z68-rec-key - 00004333600001
No errors: 02
000000001- 0309
000000002- 0312
```

The “No. errors” is the **number** of errors and is followed by the specific errors. The number on the left (“000000001”) is just a sequence number; the number on the right (“0309”) is the actual error. These errors are produced by the `butil/b_file_96_a.cbl` program. This program calls `check_record/check_z30.cbl` and `_z68.cbl` (which, in turn, calls `check_z68 a.c`). The last two programs pass error codes corresponding to particular fields back to the `p_file_96`. Generally, errors are due to an incorrect parameter being entered into the `xxx01/tab/tab_yankee_def` table – or a parameter being omitted. (The `butil/p_file_96_fill_z30`, `z68`, and `z601` programs read the `tab_96_def` and load values into the records.)

To view more information in the log when running the job, type the following

```
$ setenv DEBUG_P_FILE_96 Y
```

Then restart the batch queue (UTIL/C/3 then UTIL/C/2). The log shows the Z30, Z68 and Z601 Oracle tables that are being created.