

**For Internal Use Only**



**How to Use the  
MARCIVE Document Loader (b-file-99)**

**All Releases**

**Last update: April 9, 2001**

# Table of Contents

<b>1</b>	<b>LOADING MARCIVE RECORDS INTO ALEPH - OVERVIEW .....</b>	<b>3</b>
1.1	Running the MARCIVE Loader .....	5
1.2	Help on Fields .....	5
<b>2</b>	<b>CATALOG RECORD LOADING TABLES .....</b>	<b>8</b>
2.1	tab_doc_merge .....	8
2.2	Defaults configuration table (for example, tab_99_def): .....	10
2.3	tab_fix .....	11

## **1 Loading MARCIVE Records into ALEPH - Overview**

The MARCIVE document loader is a batch process designed to be run by libraries on a periodic basis as they receive MARCIVE data files. These files are normally received on a weekly basis and then updated on a monthly basis. There are two types of MARCIVE government document records files:

- weekly shipping lists -- brief cataloging of the documents the library receives from the GPO.
- monthly full cataloging -- full government document MARC cataloging, which may or may not correspond to the shipping list records.

The MARCIVE loader is run from the ALEPH Staff Services menu. It can be added to the libraries joblist (UTIL E/5) for regular automatic initiation.

There are several configuration tables associated with this process, some of which supply default values for automatically created records, others which determine how the loaded records are merged, and what fix procedures should be carried out.

**? USM01 - Load MARCIVE Records**

Input file

ADM library

HOL library

Create item records

Fix routine

Merge type

Merge section

Update database

Default values input file

Run Time  at  o'clock

**Figure 1 - The MARCIVE loader (file\_99)**

## 1.1 Running the MARCIVE Loader

To run the MARCIVE loader, carry out the following steps:

1. Convert the MARC file to ALEPH Sequential format using the Convert MARC Records - Step 1 (file\_01) and Convert MARC Records - Step 2 (file\_02) batch jobs consecutively (these batch jobs are also run from the ALEPH Staff Services Menu). This is because the batch process does not directly use the MARC file as input. Once the MARC file has been converted to ALEPH Sequential format, file\_99 can be run.

An institution can also run the Modify MARC Records (p\_file\_08) batch utility before running p\_file\_99 (after the file has been converted to ALEPH Sequential format) if global changes to the MARC records themselves are desired.

2. To open the MARCIVE loader from the ALEPH Staff Service menu, click "Load MARCIVE Records" (or file\_99) (listed under the rubric "Load Catalog Records"). The file\_99 batch service is displayed:
3. Fill in the fields according to the instructions in the Help for Each Field section of the batch service.
4. Click Send to queue.

## 1.2 Help on Fields

This is a more detailed and explanatory version of the Help for Each Field section available in the batch service.

### **Input file**

Mandatory. Enter the name of the Input file. If the Input file is located in the library's scratch directory, simply enter the name of the file as listed. If the Input file is located in another directory, enter the path relative to the scratch directory, or enter the full path name. An example of a relative path is ../source/filename  
An example of a full path is /aleph/a50\_5/exu50/source/filename

### **ADM Library**

Optional. Enter the alphanumeric code for the ADM library (for example, EXU50), for automatically generated ITEM records.

### **HOL Library**

Optional. Enter the alphanumeric code for the Holding library (for example, EXU60), for automatically generated HOL records. Only the weekly data loads include holdings information.

**Create item records**

Yes or No. Choose Yes to automatically generate ITEM records. If you do select Yes, you must supply a default values input file.

**Fix routine section** Optional. You can leave the “Fix routine” parameter blank or set it to one of the codes in the tab\_fix table of the bibliographic library. The specific bibliographic record fixes that should be applied to incoming MARCIVE records are a matter of library policy. Fixes that are specific to the incoming MARCIVE data should be added as a fix\_doc group to the tab\_fix table. For example, the code MRCV in column 1 can be used as the fix routine section to be used for the MARCIVE loader.

**Merge Type**

Mandatory. Merge type refers to how the incoming MARCIVE records are treated. There are two types of Merge available:

**Merge** – To merge matching MARCIVE records with bibliographic records which are already in the database

**Replace** - To completely replace bibliographic records which are already in the database with matching MARCIVE records.

The merge functionality is needed because subscribers to MARCIVE government document records normally get two types of shipments:

**Weekly shipping lists** -- brief cataloging of the documents the library receives from the GPO.

**Monthly full cataloging** -- full government document MARC cataloging, which may or may not correspond to the shipping list records.

If there is no record in the database with a matching 909 field, the incoming record is added. If there is a record with a matching 909 field, there are two possible outcomes, depending on the merge type:

(a) Merge - the incoming record overlays some fields in the existing record. The "merge section" parameter of p\_file\_99 controls which section in tab\_doc\_merge is used.

(b) Replace - the incoming record overwrites the existing record.

Thus, the merge functionality is there so that the monthly full cataloging records can overlay the weekly shipping list records.

**Merge section**

Only mandatory if “Merge” is the Merge type chosen. The “Merge section” parameter refers to the value of the first column of the tab\_doc\_merge table in the \$data\_tab directory of the bibliographic library that file\_99 is being run for. The tab\_doc\_merge table controls which fields are overlaid. The table’s header provides more information.

**Update database:**

Yes or No. If you choose not to update the database, the system performs all checks and prepares a log file with messages, but no records are written on the database files.

The file\_99 batch process does not produce output files, with the exception of the log file. In order to load the records to the library's catalog, it is necessary to set the update database flag to "Y". Records updated or created by file\_99 have the cataloger stamp set to "MARCIVE", unless otherwise specified in the default values table (see following).

**Default values input file**

Optional. You can set default values for fields in the BIB, HOL and ITEM records that are created using this service. The defaults are set in a user- defined table in the BIB library's /tab directory. Specify the name of the table in this field. If you choose to automatically create an item based on the bib information, you must set up default values for the item records in the default values table, and you must specify this table as the input file. tab\_99\_def in the USM01 \$data\_tab directory is an example of a default values table.

**Run Time**

Mandatory. You must enter the day and hour the batch process is to run. The system uses a 24-hour clock.

The log file for the MARCIVE loader can be found in \$alephe\_scratch. The log file will contain record numbers of documents that have been merged.

## 2 Catalog Record Loading Tables

The tables for tab\_doc\_merge, tab\_99\_def and tab\_fix are reproduced below:

### 2.1 tab\_doc\_merge

```
=====
!   tab_doc_merge
!Fields to retain when overlaying (merging) catalog records
!=====
! Example:
!
! 01 1 Y #####
! 01 1 N OWN##
! 01 1 C 245##
!
! 01 2 Y 245##
!
! In this example all fields are taken from the original document (1)
! except the OWN field. 245 is always taken from the copied record. If
! the copied record does not have a 245, the 245 of the original record
! is retained. Otherwise it is overlaid from the second to the first document.
!
! Please note that the search for the code is sequential. For example:
!
! 01 1 N OWN##
! 01 1 Y #####
!
! At first, the system will not take the OWN field (because of the N in
! column 3 for the OWN field). Then, the system "sees" the next line,
! which says to take all fields. The result is that the OWN field
! is taken, too.

!
Col. 1  Merge type
Col. 2  Merge direction
           1: defines lines for the .....
           2: defines lines for the .....
! COL 1.  2; NUM{01}; ;
!         Merge type;
!         Merge type:
!         Future use, always use 01;
! COL 2.  1; NUM{1,2}; ;
!         Merging direction;
!         Merging direction:
!         1: defines lines for the original record, i.e. the
!            document into which fields are merged/pasted.
!         2: defines lines for the document from which fields are
!            copied;
! COL 3.  1; ALPHA{Y,N,C}; ;
!         Action;
!         Action:
!         Final form of document will include fields from the original
!         record, together with fields from the copied record, depending
!         on the following values:
!         Y - For original record (1) - retains field
!         For copied record (2) - copies field
```



! N - Does not retain field  
! C - Retains field only if it does not appear in the other  
! document;  
! COL 4. 5; ALPHA\_NUM; ;  
! Tag code;  
! Tag code;  
!1 2 3 4  
!!-!-!!!!!

01 1 Y #####

01 2 C #####

01 2 N 008##

01 2 N LDR##

01 2 N 001##

!01 2 Y 590##

!01 2 Y 583##

!01 2 Y 037##

!01 2 Y 785##

!01 2 N 100##

!01 2 Y 245##

97 1 N #####

97 2 Y #####



## 2.3 tab\_fix

```
=====
=====
!   tab_fix
! List of fix_doc programs
=====
! TABLE_KEY 1,2
!
! HELP
! Routine name:
! Each line MUST be have a program name; to de-activate a routine either
! comment out the line, or delete the line.
! The following names are reserved for standard fixes:
! INS, INS2, INSFS, MERGE, HOL and LOCAT.
!
! INS is run automatically whenever a record is updated in the Catalog GUI.
!
! INS2 is run automatically whenever a record is updated in the Catalog GUI.
! The difference is that INS2 is executed just before the update in the
! database, and therefore can make use of the document's system number even
! if it's a new document. However check_doc programs can not check the outcome
! of INS2 programs !!!
!
! INSFS is run automatically whenever a fast cataloging is done from
! Circulation GUI, or when a user creates a record in the OPAC using the
! "New Record" option.
!
! MERGE is run automatically whenever the Paste record function is used
! in the Catalog GUI.
!
! LOCAT is run automatically whenever the Locate record function is used
! in the Catalog GUI.
!
! HOL is run automatically whenever a HOL record is created in Items
! or Serials GUI.
!
! Program names:
! Up to 20 program names may be assigned to each user-defined routine name.
! For tab04 use tab04-nn where nn is the entry inside tab04.
!
! To export these routines to the cataloging GUI see
! $data_root/pc_tab/catalog/fix_doc.<lng>
! END_HELP
!
! COL  1.  5; ALPHA_NUM; ;
!       Routine name;
!       Routine name;
! COL  2. 60; ALPHA_NUM; ;
!       Program name;
!       Program name;
! 1  2
!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>

INSFS fix_doc_tag_008

!TEST fix_doc_sub
INS fix_doc_sub
```

```

INS fix_doc_tag_008
INS fix_doc_tag_008_open_date
INS fix_doc_non_filing_ind
!INS fix_doc_own_1
!INS fix_doc_punctuation_usm
!INS fix_doc_usm_001
!INS fix_doc_001_sysno
!
INS2 fix_doc_001_sysno
INS2 fix_doc_005
!
LOCAT fix_doc_merge_1
!
MERGE fix_doc_merge_1
MERGE fix_doc_sort
!
! Following are user-defined fixes for the catalog GUI (fix_doc.[lng])
! and/or batch load.
04-01 fix_doc_tab04_01
!
008 fix_doc_tag_008
!
FIX1 fix_doc_tag_008
FIX2 fix_doc_tag_008_open_date
!
!UE-11 fix_doc_852
!
OCLC fix_doc_tag_008
OCLC fix_doc_oclc_2
!
RLIN fix_doc_rlin_1
!
ULLR fix_doc_rug_ullr_260_1
ULLR fix_doc_rug_ullr_del_tags
ULLR fix_doc_rug_ullr_add_head
ULLR fix_doc_rug_246
ULLR fix_doc_rug_ullr_245
ULLR fix_doc_rug_ullr_260_2
!
BIP fix_doc_rug_bip_del_hyphen
BIP fix_doc_rug_bip_260_1
BIP fix_doc_rug_bip_260_2
BIP fix_doc_rug_bip_del_tags
BIP fix_doc_rug_bip_add_head
BIP fix_doc_rug_440
BIP fix_doc_rug_bip_245
!
PUNC fix_doc_sort
PUNC fix_doc_punctuation_usm
!
B-25 fix_doc_tag_008
!
ANA fix_doc_new_ana
AUT fix_doc_new_aut_example
!
ILL-L fix_doc_merge_2
!
CVAUT fix_doc_sub

REF fix_doc_ref

```

SORT fix\_doc\_sort  
IND fix\_doc\_non\_filing\_ind  
LTI fix\_doc\_doc\_number

An example of a fix\_doc group for MARCIVE data files (taken from the tab\_fix for IOT01 and IOW01) would be :

MRCV fix\_doc\_notis\_1\_c  
MRCV fix\_doc\_notis\_3  
MRCV fix\_doc\_notis\_4