ALEPH VERSION 18.01

How to Load OCLC Records into ALEPH

Last Update: June 17, 2007
Document Version 1.3
Code: A-ver18-HTOCLC-1.3

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.

Documentation produced June 2007.
Document version 1.3
Web address: http://www.exlibrisgroup.com

Table of Contents

Note:
This document focuses on importing MARC records from the OCLC bibliographic utility into ALEPH

1 Importing OCLC records into ALEPH

In order to import bibliographic and authority records from OCLC into ALEPH, you need to carry out the following steps:

Step	✓
1. Check the `tab_oclc` configuration file setup	
2. Check the `tab_merge` setup	
3. Check the `tab_match` setup	
4. Check the `tab_match_script_oclc` setup	
5. Check the `tab_z30` configuration file setup	
6. Check the `tab_mapping` configuration file setup	
7. Check the `tab_merge_overlay` setup	
8. Check the setup of OCLC Connexion for Windows / OCLC CatME for Windows / OCLC Passport for Windows	
9. Start the OCLC server	

After these steps have been carried out, importing can be performed.

Each step in the import process is described in the following pages.

We also recommend that you check the `$TMPDIR/oclc_server_XXXX.log` file periodically for errors (where XXXX defines the port number receiving the OCLC MARC records). This port number is specified for OCLC transfers in the user's OCLC Export Gateway.

The Export option in the OCLC Passport software (on the client PC) must be configured to point to:

- ❑ the correct IP address or domain name for this library
- ❑ the appropriate port number in `tab_oclc`. The port number is listed in Column 1 of the `tab_oclc` configuration file for ALEPH. See the `tab_oclc` example given under Step 1: Check the tab_oclc Configuration File Setup.

Likewise, if you are using the OCLC Connexion software, you must make sure that it is correctly installed and configured to point to the correct IP address or domain name for the library and to the appropriate port number in `tab_oclc`.

Based on position 9 of the LDR of the incoming record, ALEPH can determine if the record is in MARC-8 or in UTF-8 (UCS/Unicode).

According to the MARC format, position 9 of the LDR identifies the character coding scheme used in the record. If it is ‘a’, then the coding scheme is UTF-8 (UCS/Unicode). If it is anything else, then the coding scheme is MARC-8.

If position 9 of the incoming record contains ‘a’, the following instance in tab_character_conversion_line is used: OCLC_UTF_TO_UTF.

If position 9 contains something other than ‘a’, the following instance in tab_character_conversion_line is used: OCLC_TO_UTF.

Both character conversion definitions can be used in tab_character_conversion_line.

The incoming record is passed through fix routines specific to OCLC. Additional institution-specific fixes for incoming OCLC data can also be applied. These routines can be library-sensitive (for example, XXX01 or XXX10). When loading OCLC bibliographic records, you can also set up the system to create item records and/or holdings records automatically upon import.

Step 1: Check the tab_oclc Configuration File Setup

The `tab_oclc` configuration table is located in the `$alephe_tab` directory:

```
! 1    2    3     4     5   6 7       8         9         10
!!!!!-!!!-!!!!!-!!!!!-!!!!!-!-!- !!!!!!!!!!- !!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!
7505  BIB XBI01       12         OCLC        OCLC
7505  AUT XBI10                  OCLC        OCLC
7505  BIB XBI01       2          OCLC        OCLC
```

Key to tab_oclc
Each of the columns in the table above indicates how many characters the length of the column will contain. For example, column 10 can accommodate up to 20 characters if needed.

- ❑ Column 1 defines the port number receiving the OCLC MARC records.
- ❑ Column 2 indicates the record type being transferred (bibliographic or authority records).
- ❑ Column 3 specifies the destination database (bibliographic or authority) for incoming records.
- ❑ Column 4 specifies fix routines to be run from the Column 1 of the `tab_fix` table (see OCLC fix routines on page 15). The OCLC fix routine is always active and does not have to be defined in Column 4 of `tab_oclc`.
- ❑ Column 5 can include up to five special fixes. Currently, two (hard-coded) fixes are provided:

    1. Inserts the data from column 2 in tab_z30 to the OWN field (`oclc_server_new_fix_1`).

    2. Modifies/creates 001 field using the data from column 2 in tab_z30 and the record system number: <*`Column 10 data`*>-<*`system number`*> (`oclc_server_new_fix_2`). For example:

```
OCLC-000037197
```

3. Modify/create 001 field using the record system number.
   Modify/Create OWN field base on column 2 in tab_z30

   If column 2 in tab_z30 is empty, no OWN field will be created. If you want more than one special fix to be run, set up Column 5 as follows:

```
! 1    2    3     4     5   6 7       8            9        10
!!!!!-!!!-!!!!!-!!!!!-!!!!!-!-!- !!!!!!!!!!- !!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!
7505  BIB XBI01       12         OCLC        OCLC
```

   This means that special fixes 1 and 2 will be run on incoming records.

- Column 6 - Not in use

- Column 7 - Not in use

- Column 8 - The value in column 8 indicates which routine from the `tab_merge` table will be used when a bibliographic or authority record in ALEPH is created or overlaid. **By default, the system performs merging according to the routines specified for the OCLC routine in the `tab_merge` table.**

- Column 9 is used to specify the match routine to be used to check the input file against the database. **Match routines are defined in the `tab_match` table of the library's `tab` directory.**

- Column 10 Not in use

Step 2: Check that OCLC routines are specified in the tab_merge table

The `tab_merge` table is located in the XXX01 library in the `tab` directory.

```
!   1                   2                               3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!>
OVERLAY-01 merge_doc_overlay              01
OVERLAY-02 merge_doc_overlay              02
OVERLAY-03 merge_doc_overlay              03
OVERLAY-04 merge_doc_overlay              04
RLIN       merge_doc_overlay              01
OCLC       merge_doc_overlay              03
TEST       merge_doc_overlay              03
HVD        merge_doc_adv_overlay          01
```

Key to tab_merge

The three columns in this table list the merge routines, the routines' program name and the programs' arguments (the section identifier of `tab_merge_overlay` and `tab_merge_adv_overlay`). The program for OCLC is `merge_doc_overlay` which calls a specific section of the `tab_merge_overlay` table (see Step 7 on page 12).

The OCLC server automatically refers OCLC records to the `tab_preferred` table. `tab_preferred` lists the table to use for setting which is the "preferred" document ("preferred" means the document into which the incoming record is merged).

In the OCLC server the preferred record is the record in the incoming load file. In order to change direction, add preferred_doc_switch to tab_preferred; it changes the direction of merge_doc_overlay (or tab_merge_adv_overlay) and switches between the preferred and the non-preferred records.

```
!1                    2                       3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!>
AD-OVERLAY preferred_doc_cdl              union_preferred
OCLC       preferred_doc_switch           union_preferred
```

If the library does not want to determine which is the preferred record (that is, the incoming record is always merged into the database record), the table referred to in Column 3 of `tab_preferred` must be set to a name such as `db_always_preferred`, and accordingly, an empty table called `db_always_preferred` must exist.

The OCLC fix routines that are listed in `tab_fix` are discussed at the end of this document.

Step 3: Check the setup of the tab_match table

Verify that the `tab_match` table includes the OCLC match routine, the program name and the program arguments. It is located in the XXX01 library in the `tab` directory.

```
! 1                 2                              3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
YBP   match_doc_uid                  I-ISBN
YBG   match_doc_gen                  I-ISBN
RLIN  match_doc_uid                  T-020
!RLIN3 match_doc_uid                  T-022
!OCLC  match_doc_uid                  T-020
OCLC  match_doc_script               tab_match_script_oclc
MRCV  match_doc_uid                  I-909
!MRCV  match_doc_script               tab_match_script_oclc
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 OCLC records the match routine code is OCLC.
- Column 2 lists the match program used by the match routine. The program used by the OCLC match routine is `match_doc_script`.
- Column 3 indicates the program arguments. In this case the arguments are `tab_match_script_oclc` which is a table located in the tab directory.

Step 4: Check the setup of the tab_match_script_oclc table

Verify that the program arguments are set up correctly in the `tab_match_script_oclc` table.

In order to create tab_match_script_oclc, copy the header of tab_match_script, and save the new file as tab_match_script_oclc.

Note:
When displayed, this table's header name is `tab_match_script`

```
!1     2                  3      4                5
!!-!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!>
01 match_doc_gen        1    goto 03    TYPE=IND,TAG=035##,CODE=035
01                      0+   goto 02

02 match_doc_gen        20-  goto 03
TYPE=ACC,TAG=245##,SUBFIELD=abdefgknp,CO
DE=TIT,TRUNCATION=Y
02                      20+  stop
```

Key to tab_match_script_oclc

This table contains five columns:

- Column 1 - the match set identifier

- Column 2 - the name of the match program. Currently there are two match programs and `match_doc_gen` is used for OCLC.

- Column 3 - refers to the number of records in the database that match the incoming record. You can specify an exact number, an upper number limit (*nn*-) or a lower number limit (*nn*+). 0+ indicates at least one match; 0 indicates no match.

- Column 4 - indicates the action to be taken where the condition of number of matched records is true. Supported actions are: `skip` (to skip to the next match set); `stop` (to stop script execution); `goto <xx>` (to jump forwards/backwards to a different match set <xx>; `<any text>` acts in the same manner as `skip`. The table above uses the `goto` and the `stop` actions.

- Column 5 lists the match program arguments. For the 01 match set, the program arguments are: `TYPE`, `TAG` and `CODE`. For the 02 match set, the program uses the `TYPE`, `TAG + SUBFIELD`, `CODE` and `TRUNCATION` arguments.

Step 5: Check the tab_z30 configuration file setup in the XXX50 library

```
!    1          2       3     4    5   6   7                8
!         9     10 11
!!!!!!!!!!-!!!!!!!!!!-!!!!!-!!!!!-!!-!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!-!!!!!!!!!!-!-!
WID        WID        WID   GEN   92 BOOK  N
                 3
XZLA       MASTER50   WID   GEN   28 BOOK
       3
```

Key to tab_z30

This table is used for setting the default values of item records that are system-created through the OCLC server. The `tab_z30` table, which is located in the XXX5*n* `tab` directory, defines the sublibrary, collection, item status, material type, OWN field, OWN group, item creation, holdings creation and the location that will be assigned to the new item record, based on the 049 field in the incoming bibliographic record.

Note:
Item records are never created automatically when a bibliographic record is being overlaid.

The header information in `tab_z30` specifies the character length of each column as well as the purpose of each column. For example, column 6 can accommodate up to five characters. The table can hold up to 1000 lines.

Note:
The `oclc_server` program looks for `CSCR-OCLC-Z30-BARCODE` in `tab_checksum`. Make sure that such a routine is present before starting to import.

- ❑ Column 1 indicates the holding code that appears in the 049 OCLC tag of bibliographic records. In a multi-ADM environment, the content of the field can only appear once within all the `tab_z30` tables in the various ADM libraries.

- ❑ Column 2 contains data to be added to the OWN field of records received from the OCLC server. If Column 2 is empty, no OWN field will be created.

- ❑ Column 3 lists the sublibrary of the item.

- ❑ Column 4 shows the collection to which the item belongs.

- ❑ Column 5 is the item status.

- ❑ Column 6 is the material type code of the item.

- ❑ Column 7 Not in Use

- ❑ Column 8 specifies the item call number. Normally this column is left blank. If an institution imports numerous bibliographic records using a specific call number for a set of materials, that call number can be put in this column for the duration of the project. If this column is left empty, the call number mapped in `tab_mapping` is taken.

- ❑ Column 9 is used to group different OWN codes to a common code. This is used when a library uses different 049 codes, and it creates a different OWN code for each 049 code. If no OWN group is defined, a separate BIB record will be created for each separate OWN code. If several OWN codes are assigned to the same OWN group, each OWN code is added to a single shared BIB record. The OWN group code is not written in the BIB record, it is used only in this table for grouping

- ❑ Column 10 defines how to create an item for the loaded record

  0 - Don't create item(s)
  1 - Create the first item for new records only; the system will generate a barcode.
  2 - Create first and additional items using the definitions in 949/852 fields. If the field is missing (or barcode missing) the system will generate a barcode.
  3 - Create first items using the definitions in 949/852 fields. If the field is missing (or barcode missing) the system will generate a barcode.
  4 - Create first and additional items using the definitions in the 949/852 fields. If one of the fields or the barcode is missing, the item will not be created (the ADM record will be created in any case). Items will be created even though the BIB record already has linked items.
  5 - Create first items using the definitions in 949/852 fields. If one of the fields or the barcode is missing, do not create items (the ADM record will be created in any case). In this case, an ADM record will be created. Items will not be created if the BIB record already has linked items.

  Item creation for types 4 and 5 is based on two fields in the BIB record:

  Field 949: $$a (for barcode), $$c (for Shelf list)

  Field 852: $$p (for barcode), $$a (for Shelf list)

  An item is created for every occurrence of the 949/852 field+Shelf list.

  A uniqueness check is performed based on the barcode.

  System-generated barcodes are created based on the definition of CSCR-OCLC-Z30-BARCODE in tab_checksum

- ❑ Column 11 defines how to create the HOL record

  0 – Do not create HOL record.
  1 - Create the first HOL for new records only.
  2 - Create HOL records (check uniqueness with using the 852 field – if a matching HOL record exists, a new one will not be created). Use information that is in brackets in the 049 field to created 852 subfields k and m in brackets.
  3 - Create HOL records (check uniqueness with 852 field– if matching HOL record exists, a new one will not be created).Do not created 852 subfields k and m from information in brackets in the 049 field. .

Step 6: Check the tab_mapping configuration file setup in the administrative libraries

In `tab_z30` (above), when there is 1-3 in column 11, the system automatically creates a holdings record during the OCLC transfer process when a new bibliographic record is added. The `tab_mapping` file defines where information from the OCLC record is placed in the ALEPH holdings record. The table responsible, resides in the `$data_tab` directory of each administrative library.

Note:
The `tab_mapping` table also exists in the bibliographic library.

```
!    1            2     3     4     5                 6                7 8
!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!

BNGG            049## a     8520  b     s/BNGG/XBIMA/                  Y Y
BNGG            049## a     852   c     s/BNGG/MAIN/                   Y N
BNGG            050## a     852   h                                    Y N
BNGG            050## b     852   i                                    Y N
BNGG            090## a     852   h                                    Y N
BNGG            090## b     852   i                                    Y N
BNGG            086## a     8523  h                                    Y N
BNGG            099## a     8524  j                                    Y N

BNGD            049## a     8523  b     s/BNGD/XBIMA/                  Y Y
BNGD            049## a     852   c     s/BNGD/MGDOC/                  Y N
!BNGD            050## a    852   h                                    Y N
!BNGD            050## b    852   i                                    Y N
!BNGD            090## a    852   h                                    Y N
!BNGD            090## b    852   i                                    Y N
BNGD            086## a     8523  h                                    Y N
BNGD            099## a     8524  j                                    Y N
```

Key to tab_mapping
Each of the columns in the table above indicates how many characters the length of the column can contain. For example, column 6 can accommodate up to 30 characters.

- Column 1 specifies the OCLC 049 code of the library.
- Column 2 is the Source code from the OCLC record. These are the tags where the information is originally located. The call number tags in the example above are listed (from the top down) in the order of least preferred.
- Column 3 indicates the Source subfields (from within the source code fields).
- Column 4 lists the Target Code in the ALEPH Holdings record where incoming data are placed. It can also accommodate the first indicator of the 852 field (which specifies call number type) as shown in the example above.
- Column 5 specifies the Target subfields for the exact location of the incoming data in the ALEPH Holdings record.
- Column 6 is where textual information may be entered. For example on line 1 in the table above the formula `s/BNGG/BXIMA/` instructs the system to substitute the text `XBIMA` for the text `BNGG` in subfield b of the 852 field when the data is coming from the subfield a in the 049 field of the OCLC record. On line 2 in the example above, the formula `s/BNGG/MAIN/` tells the system to substitute the text `MAIN` for the text `BNGG` in subfield c of the 852 field when

the data is coming from the subfield a of the 049 field in OCLC. You can also append information using an a for append which adds a prefix/suffix to the input code. For example, this line in tab_mapping:

```
BNGD 050 a 852 b a/<foo_>/<_bar>
```

produces

foo_BNGD_bar as 852 subfield b.

- Column 7 is the overlay flag. If it is set to “Y” all incoming data will overlay any previous data. If it is set to “N” there will be no overlay of data

- Column 8 is the New Line Flag which can be set to “N” or “Y”. When it is set to “N” a new line is not created and, based on the value in column 7, data can be overlaid in the ALEPH Holdings record.

The tab_mapping file is essential for the automatic creation of holdings records during the import of OCLC records. The table above illustrates an institution with two OCLC holding code symbols (BNGG, the main library holding code and BNGD, the government documents holding code).

Step 7: Check the tab_merge_overlay configuration file setup

The tab_merge_overlay file defines which fields are retained when overlaying cataloging records in ALEPH. It is located in the tab directory in the XXX01 library.

```
1 2 3          4
!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
01 1 N #####
01 1 Y LDR
01 1 Y 001
01 1 Y 09###
01 1 Y 5####
01 1 Y 79###
…/…
```

Key to tab_merge_overlay
In the example above all the fields will be overlaid from the second to the first document.

Note that column 3 of the tab_merge table contains the merge set - from the tab_merge_overlay table - that is run when the merge routine in column 1 is selected. Column 8 of the tab_oclc table contains a merging routine that should match a routine from the tab_merge table (column 1).

- Column 1 has a default value of 01, but other values can be added for use with various batch services (see those that have been commented out with exclamation marks).

- Column 2 defines lines for the original record (value=1), the document into which fields are pasted, as well as defining the lines for the document from which the fields are copied (value=2).

- Column 3 determines what the final form of the bibliographic record includes. This is based on the fields from the original record, together with the fields from the copied record, depending on:

    - the following values: Y, N, or C as defined in the header for each field in Column 4;
    - which document the field is from in Column 2.

- Column 4 lists the OCLC MARC tag codes for the fields in the bibliographic record. The hash marks (#) in the above example indicate all tag codes.

Step 8: Check the OCLC Connexion for Windows Setup

OCLC Connexion
Connexion is a robust package of integrated cataloging tools and services. It has two interfaces; one in the browser and the other is a client system.

To set up the client system:

1. In Connexion, from General/Preferences/Export Options, select the TCP/IP Connection option.
2. Enter an external IP for the ALEPH server and a designated port (per `tab_oclc`).
3. Start the `oclc_server` from ALEPH. Do not forget to export records in MARC format.

More detailed information on setting up OCLC's Connexion is available from OCLC. (Try the URL: http://www.oclc.org/connexion/default.htm)

Step 9: Start the OCLC server

1. From the command prompt, select UTIL W (Server Management)
2. Choose option 3, Start Servers
3. Select option 7, Other Server and then option 3, OCLC Server
4. Accept the default port number unless you need to choose another

If you decide to monitor the servers under UTIL W/1, the OCLC server is listed as "Generic"

After steps 1 – 9 have been carried out, you can begin to import OCLC bibliographic records into ALEPH. Please note that any time you make changes to the tables described above, you must stop and restart the OCLC server.

2 General Work Flow of the OCLC Loader

After the OCLC server is running, the OCLC loader handles records as follows:

1. The loader looks for the `tab_z30` table which contains the holding code from field 049. The ADM library where `tab_z30` resides is the ADM environment for the loading.
2. The loader looks for matching records using the section defined in `tab_oclc` (col. 9).
3. The loader filters the matched records according to the owner group from `tab_z30`, col. 9. It leaves only those records which have an OWN field that belongs to the OWN group as defined in `tab_z30`.
   If there is more than one match, an error is returned.

   If there is one matched record, then the loader merges the new loaded record with the existing one, adds a new OWN field, and saves it.

4. If there is no matched record, then the loader creates a new bibliographic record, adds an OWN field, and then executes the special fix routine defined in col. 4 of `tab_oclc`.
5. The OCLC loader creates an ADM record in the ADM library.
6. The OCLC loader creates an HOL record, according to the definition in `tab_z30`. The HOL library is determined using tab_library_relation.
   The HOL fields are created using tab_mapping table.

   The HOL record will not be written if a record already exists with the same 852 field with the following subfields: $b, $c, $h, $i, $j, $k, $m, $l.

7. The OCLC loader creates item record(s) according to the definitions in `tab_z30`.
   The new item(s) will be linked to a HOL record if the later record's 852 $b and $c subfields match with the Z30-SUB-LIBRARY and Z30-COLLECTION fields of the first.

8. If an item with the same barcode already exists, no item will be created, but if the item does not have a HOL link then it will be linked to the HOL record if it exists. Again, the match is done by comparing subfields $b and $c of the 852 field with the Z30-SUB-LIBRARY and Z30-COLLECTION fields of the item.

Note:
If position 05 of the LDR field of the input record contains "d" or position 23 holds X'03', then the matching bibliographic record and all the associated administrative records will be deleted. A STA $$a DELETED will be added to the record.

However, if the record has any of the following records, it will not be deleted and appropriate error message will be displayed:

Loans (Z36)
Hold Requests (Z37)
Photocopy Request (Z38)
Acquisition Order (Z68)

3 OCLC fix routines

The OCLC server always looks for the presence of the OCLC fix routine in the `tab_fix` table. In the `tab` directory of both the XXX01 library and the XXX10 library, there are OCLC procedures that can be specified in the `tab_fix` table. Two of the programs are:

fix_doc_oclc- This program moves the OCLC 001 and 003 fields to the ALEPH (MARC 21) 035 field, in the following format: (003)001. For authority records, `fix_doc_oclc` also adds the UPD field (Y or N).

There are variations of this called `fix_doc_oclc_2` and `fix_doc_oclc_retain_001`.

fix_doc_oclc_2 - deletes pre-existing 035 fields.
fix_doc_oclc_retain_001 - retains the 001 field.

Only one of these two programs should be active.

Here is an example of OCLC `fix_doc` programs in a `tab_fix` table from a XXX01 bibliographic library (only contains lines relevant for OCLC):

```
! 1                 2                              3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
OCLC  fix_doc_tag_008_open_date
OCLC  fix_doc_non_filing_ind
OCLC  fix_doc_punctuation_usm
OCLC  fix_doc_005
OCLC  fix_doc_oclc_2
OCLC  fix_doc_usm_001
```

From a XXX10 authority library (only contains lines relevant for OCLC):

```
! 1                 2                              3
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
!OCLC  fix_doc_tag_008
OCLC  fix_doc_oclc
```

In order to export these fix routines to the Cataloging GUI, refer to the `fix_doc.eng` table in the `$data_root/pc_tab/catalog` directory. This table defines the `fix_doc` routines that are included in the Cataloging module under the Edit menu/Fix record and the Edit menu/Derive record options. Here is an example of a `fix_doc.eng` setup:

```
! 1   2 3           4
!!!!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
04-01 N L Convert UNIMARC Records to USMARC Records
008   N L Update 008 field from 260 field
MAR   N L Generic fix
ANA   Y L Create a new Analytic record
AUT   Y L Create an authority record based on 1XX,4XX,6XX,7XX field
REF   N L Update heading from authority x-ref
HOLD  N L Create local note in HOL rec.
OCLC  N L OCLC fix
OCLC  N L OCLC
```

Key to fix_doc.eng

- Column 1 lists the procedure code which is the unique code by which the system identifies the procedure. It must be a routine name defined in column 1 of the `tab_fix` table (UTIL M/11), such as `OCLC`.

- Column 2 defines whether a new record is going to be created when performing a fix routine, or if the current record is going to be fixed. The possible values are: Y = Open as a new record; C = Conditional Open as a new record - only active when the record has a system number (that is, not NEW); N = Fix current record.

- Column 3 is the alphabetic code and must always be set to L.

- Column 4 is the text that displays in the window when the user invokes the Fix record option or the Derive record option from the Cataloging module.

4 ALEPH Loading Services

4.1 Loader Logger: the Z73 Oracle table

The Z73 Oracle table stores information on the running of the MARCIVE (p-file-99) and OCLC loaders.. The log has a running number. This number also serves as the key of the Z73 table. The sequence has to be defined under Z52 of the Bibliographic library – “last-loader-log-no”.

Each message created by MARCIVE or OCLC loaders is saved in a different Z73 record.

Each message contains a type (I=Information/ S=success /F=Fail) and explanatory text. The text is defined under $aleph_error_eng/generic_loader.

Note that a new Z73 log number will be generated for each record loaded via the OCLC server.

4.2 Load OCLC Records (file-93)

This service loads OCLC records in batch mode into the system without making use of the OCLC Server. This service can load both bibliographic and authority records. The service can create an output file that can be found later in the library's PRINT directory. The file has the same name as the input file.

The output file contains the following information:

Administrative library, OWN Field, Open Date, Log No., Action (Success, Failed or Information), Message information.

This information is also stored in the Z73 Oracle table (Loader Logger).

Note that a new Z73 log number will be generated for each file-93 batch run. For each loaded record in a specific run a new log sequence will be generated.

This is different from a record loaded via the OCLC server, in which case a new Z73 log number + log sequence will be generated for each record.

The parameters for this batch are:

Input File which should reside in the library's SCRATCH directory.

Fix Routine from tab_fix of the relevant library (Bibliographic or Authorities). Please note that the OCLC section in tab_fix is executed as well.

Special Fix Routine enables you to include up to 5 special fixes. Currently the following options are available:

1 - Creates OWN field, populated by the value set in col.2 of the tab_z30 table.

2 - Modifies/creates 001 field using the value from col.2 of the tab_z30 table and the record system number in the following manner: <Col.2 value>-<system number>

3 - Modifies/creates the 001 field using the record system number In addition, it modifies/creates the OWN field based on col. 2 of the tab_z30 table. If the column is empty, no OWN field will be created.

Match Routine from tab_match of the relevant library (Bibliographic or Authorities).

Merge Routine from tab_merge of the relevant library (Bibliographic or Authorities).

Produce Loading Report determines whether or not to produce a report regarding the current load.

Report File should hold the report’s file name.

4.3 Produce Report for Loader Logger (file-94)

This service produces a report of records loaded from OCLC and MARCIVE loaders into ALEPH. The information generated in the report comes from the Z73 (Loader Logger) Oracle table.

The report contains the following information:

Administrative library, OWN Field, Open Date, Log No., Action (Success, Failed or Information), Message information.

The parameters for this batch are:

Output File will be found later in the library's PRINT directory.
Action Date From/To to retrieve all records that have been generated between the dates you enter here.
Log Number for which the report will be generated. If no number is entered, all logs will be included in the report.

ADM Library for which the report will be generated. If no ADM library is entered, all libraries will be included in the report.

OWN Field for which the report will be generated. If no OWN field is entered, all OWN field contents will be included in the report.

Report Format from a list of predetermined report formats.

Sort By determines the order by which the information to be sorted. Three sort options are available: ADM library, OWN field and Log Number.

5 Troubleshooting

You can use the OCLC server transaction log as a troubleshooting tool (similar to the PC server transaction log). The transaction log can be checked as follows:

```
oclc_server view <port> <number of last transactions>
```

This command can be used to view the transactions specified in the `<number of last transactions>` parameter.

```
oclc_server check <port> <transaction number>
```

This command can be used to replicate the transaction specified in the `<transaction number>` parameter.