



ALEPH VERSIONS 16, 17, 18, 19

How to Export BIB Records for Processing

Ex Libris

© Ex Libris Ltd., 2008

Last Update: October 16, 2008

Document Version 1.2

Code: A-ver16-ver17-ver18-ver19HTBIBProcess-1.2

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.

Copyright Ex Libris Limited, 2008. All rights reserved.
Documentation produced July 2005
Document version 1.2
Web address: <http://www.exlibrisgroup.com>

Table of Contents

1	BIB EXPORT	4
1.1	p_ret_01	4
1.2	p_print_03	5
2	BIB IMPORT	6
3	AUTHORITY IMPORT	8
4	INDEXING	9

1 BIB Export

Note

This document is intended for sending BIB records to a vendor for processing. If you are exporting BIB records to a union catalog, where holdings information is especially important, then you may want to use p_export_01 instead of the p_ret_01 / p_print_03 described below. There is a separate document, *Exporting Holdings Using p_export_01*, which describes this.}

To export BIB records for processing, follow steps 1.1 and 1.2 below:

1.1 p_ret_01

First, you must create a set of record numbers to use as input to the GUI Services p_print_03 function (see Step 1.2 below). This set of record numbers is created using the *Retrieve Catalog Records (ret-01)* GUI service or by doing an OPAC search and saving the set to the server

p_ret_01 filters the records it includes by:

- Document number range,
- Creation date range, and/or
- Field/subfield/"From"-text/"To"-Text.

(For further information, see the Help section at the bottom of the ret-01 submission screen in the GUI Services.)

You can use these filters to exclude, for example, the following:

- Suppressed, Deleted, or "Circ-Created" (Fast Cataloging) records based on the "STA" field value.
- On-order records, based on the Item Process Status (via the PST field, subfield e) and/or the HOL 852, subfield z
- Multiple filters

Example: Suppressed, Deleted, or "Circ-Created" (Fast Cataloging) records based on the "STA" field value

In the Help section for the From - Up To field of p_ret_01, the following appears:

"In order to retrieve documents in which the requested field is missing, enter "ZZ" in both the From and Up To fields. For example, if you wish to find documents that do not have an Author field in them, enter 100 for Field and then enter "ZZ" for both From and Up To."

To apply this:

1. Retrieve all records without an STA field (by putting "ZZ" in both the From and To fields).
2. Retrieve the set of records with STA values which you “do” want to include if there are any such records).
3. Concatenate these two sets (as described below)

Example: On-order records, based on the Item Process Status (via the PST field, subfield e) and/or the HOL 852, subfield z.

{You must ensure, for the latter, that you have any relevant expands (such as expand_doc_bib_loc_x_n, expand_doc_loc_usm, or expand_doc_hol_usm) defined in the RET section of the xxx01/tab/tab_expand file.}

Multiple filters

You can create multiple p_ret_01 outputs {specifying, for instance, From-text = A, To-text = N in one run; and From-text = P, To-text = “Z” (to omit anything beginning with “O”)}. Then concatenate them:

```
M505> cat file1 file2 > allrecs
```

You can then input the resulting file to p_ret_01 to achieve multiple field-level filtering.

You can also use the p_ret-10 procedure. p_ret_10 allows you to combine two sets using Boolean operators.

You can also sort the p_ret_01 (or p_ret_10) output, by running it through the p_ret_21 (“Sort records”) routine.

1.2 p_print_03

The p_ret_01 (or p_ret_21) list of record numbers is then input to the “Retrieve Catalog Records”/“Download Machine-Readable Records” (p_print_03) GUI service which outputs the records in MARC 21 format.

Run p_print_03 as follows:

- For “Field 1” specify ALL
- For "Format" specify "MARC with alphanumeric tags". This is to preserve the OWN, LKR, CAT, etc., fields in your bib records. (If your vendor does not support alphabetic tags, they are dropped in this process. See Step 5 of BIB Import.)
- For “Expand routine” you would normally specify “NONE” – but you can specify any of the routines you see in the dropdown –which is all of the routines in the xxx01 tab_expand
- For “Character Conversion” specify “NONE

- For “Fix routine”;

The exported records need to have an 001 field so they can be matched to the database records on import. (See section 2 below.)

There are two fix routines which will create an 001 field with the system number in it: `fix_doc_001` and `fix_doc_001_prefix_sysno`. The former puts only the 9-digit system number in the 001, while the latter includes a prefix (which you specify), usually an ALEPH library code.

Two complementary programs were developed:

- `fix_doc_001_sysno` to insert a special 001 on export and
- `fix_doc_001_sysno_inv` to match on this special 001 on import. The `fix_doc_001_sysno_inv` removes the special 001 upon import.

I say "special 001" because the 001 which the `fix_doc_001_sysno` generates looks like this: `EXU01000000000000nnnnnnnnnn` . (The 5-byte ALEPH library code, followed by 12 bytes of zeros, followed by the 9-byte ALEPH system number.) The idea is that this distinctive format can be used to distinguish these 001s from "real", permanent 001s which need to be preserved on import.

Also, if the record has an existing, permanent 001 field, the `fix_doc_001_sysno` will not generate the special 001.

What this means is that these programs work well if your records do not have any existing, permanent 001s, but not so well if they do. We, therefore, recommend that if you want to export/re-import your records, you set them up either so none of them have 001's (and you can use `fix_doc_001_sysno` in running `p_print_03` and `fix_doc_001_sysno_inv` in running `p_manage_18`) or so all of them have 001s (and you can then use `p_manage_36`, `tab_match`, etc.).

Note

Catalogers should be told *not* to update the records which have been exported for processing. These records are completely replaced upon return and any updates made to them are lost. (It's possible that something could be done with `tab_merge` to preserve certain fields.)

2 BIB Import

Note

The BIB import can take place after the authority import (step 3), if you prefer.

To import BIB records, follow these steps:

1. Put the file in `$data_scratch (xxx01/scratch)`.
2. Run it through Convert MARC records -- step 1 (`p_file_01`), creating a new output file.

3. Run the output from the previous step 2 through Convert MARC records -- step 2 (p_file_02), again creating a new output file.
4. If you want to remove fields or add fields, you can run the p_file_02 output through "Modify MARC record file" (p_file_08), using a locally created processing script.
5. If you specified "MARC with alphanumeric tags" in the [p_print_03](#) export and if your vendor preserved the alphabetic tags, then no additional steps are necessary. But if these vendor-processed records lack the alphabetic tags (OWN, LKR, CAT, etc.), then you need to *merge* the incoming records with the existing records. This can be done in one of two ways:
 - Run p_manage_38 as a separate step after p_file_02 and before p_manage_18
 - Run p_manage_18 and select "MERGE" as the "If Updating Current Records" value and then specify a tab_merge "Merge/Preferred Routine" which you create (with the existing alphabetic tags being preserved).
6. If you do not plan to use fix_doc_001_sysno_inv for the fix routine in p_manage_18 (see below), then you will need to run p_manage_36 to match the incoming records to the corresponding database records before running p_manage_18.
7. Load the BIB records using "Load catalog records" (p_manage_18), using the file created in step 3 (or step 4) as your input.
 - For "Procedure to run" specify "Update current records in the database."
 - For "If updating current records" specify "Replace entire record".
 - For "Headings" specify "Do not change heading based on cross-reference".
 - For "Indexing":
 - If you are reloading the entire database and if that database is greater than 200,000 records, then specify "NONE" for Indexing.
 - If you reloading just part of the database or if reloading the entire database and it's less than 200,000 records, then specify "FULL" for indexing.
 - For "Fix routine" specify "Input comparing sysno with 001". (For this to work, you will need to have a "SYSIN fix_doc_001_sysno_inv" entry in your xxx01/tab/tab_fix. If you want to run other fixes along with this fix_doc_001_sysno_inv, you can put them under FIX3, for instance, and specify that instead.) This is essential if you want the records to overlay the existing records in your database from which they came. Unless, as noted above, you are using p_manage_36 to match the records.
8. If you want a cataloger name and/or level, enter it.
 - For "No. of catalogers currently working" specify "Single user". (This presumes that you are running the job when other BIB records are not being created/updated.)

3 Authority Import

Note

Your processing may not have involved the creation of authority records, in which case you can skip this section. The authority import can take place before the BIB import (step 2), if you prefer

The authority files go into the authority database (usually xxx10), using the same process as above (BIB Import) with a few exceptions.

To import authority records:

1. Put the file in \$data_scratch (xxx10/scratch).
2. Run it through Convert MARC records -- step 1 (p_file_01), creating a new output file.
3. Run the output file from the previous step through Convert MARC records -- step 2 (p_file_02), again creating a new output file.
4. Add the UPD field to each MARC authority record so that UE_08 receives processing instructions (see UE_08 description below). To add the UPD or other field, use the "Modify MARC record file" (p_file_08) utility, using a locally created processing script.
5. Load the authority records using "Load catalog records" (p-manage-18), using the file created in step 4 as your input.
6. Select the option to Add new records to the database
7. Choose "Do not change heading based on cross reference" as your authority vendor will have taken care of that.
8. For indexing: If you are reloading the entire database and if that database is greater than 200,000 records, then specify "NONE" for Indexing.
 - If you are reloading just part of the database or if you are reloading the entire database and it's less than 200,000 records, then specify "FULL" for indexing.
 - No fix routine
 - If you desire a cataloger name and/or level, enter it.

[The following has been noted by Pat Riva at McGill University: tab07 has the same problem with \$w positions as edit_doc_999.eng -- it does not recognize the code "g" in the sequence "gnna", so generates too few links. As the "nna" is something we had put in by BNA to avoid "blind" see also, and isn't in the LC versions of the records, and we had never had time to keep it up to date, and it has no useful function on ALEPH. If other libraries are planning to do an authorities project, they should be warned not to ask for "deblinding".]

4 Indexing

If you have specified “FULL” for Indexing above in p_manage_18, then ue_01 will perform the necessary index updates. If you did not specify “FULL”, then you will need to run a complete suite of index jobs. In this latter case, please consult the *How To Run Index Jobs* document.