USER DOCUMENTATION

Exporting Holdings Using p_export_01

Releases 12.4 and Higher
Last Update: December 11, 2006

Table of Contents

Overview

Libraries which belong to a bibliographic utility (e.g., OCLC or RLIN) register their holdings with that utility (i.e., to indicate ownership of the item). If the library has obtained a bibliographic record from somewhere other than the utility, they usually notify the utility of the library's ownership of the record.

Also, libraries which belong to a consortium may need to report their holdings to the consortia for inclusion in a Union Catalog.

The batch process p_export_01 provides the ability to identify holdings records that need to be exported and then creates a file of bibliographic records with embedded holdings that can be sent to the utility or consortium for processing/uploading.

Important Note:
From version 15 onwards, p_export_02 provides the same functionality as p_export_01 in a simpler and more efficient matter.

Identifying Records to be Exported

Libraries commonly identify the records they want to export using a tag in the HOLdings record and a flag. The example used in EXU is XPT, with a value of Y indicating that the record should be exported. So, for example, the holdings record would look like this:

```
852 $b XLINC $c GEN $h PN567 $i .M55
XPT $a Y
```

A fix_doc can be used to automatically insert this flag into each new HOLdings record as it is added to the database, preparing the holdings statement for export using the p_export_01 process. Libraries may choose to use a tag other than XPT. Also, libraries may choose to export all holdings for a sublibrary or group of sublibraries, whether or not it is marked. See the section *"Table setup – Not using the XPT flag"*.

Table Setup – For a Single Sublibrary

This first section describes how to use the export process to export holdings from only a single sublibrary, where the holdings records are marked with an **XPT $a Y** tag. See the section below on multiple sublibraries for more information on exporting holdings from a group of sublibraries.

1. $alephe_tab/tab_export

```
   1     2    3     4                                                    5
!!!!!--!!!!!--!--!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!--!!!!!!!!!!>
XLINC  MARC   N  XPT,a,Y,Y,852,b,XLINC,XLINC                         /aleph/u01/outdir
```

The first column is the target. It has a maximum of 5 characters and may be either a single ALEPH sublibrary or an artificial name that represents a group of sublibraries. This will later be used as a fix section (in tab_fix), as the name of a fix script, and also to name the output file.

The second column is the export format of the records - the available options are MARC, MARC with alphanumeric fields or ALEPH (for ALEPH sequential). Most bibliographic utilities expect records in MARC format.

The third column specifies whether or not you want the export file zipped. If 'Y' (Yes) is specified, the resulting export file will be zipped (using gzip). If 'N' (No) is specified, the resulting export file will not be zipped.

The fourth column contains the retrieval parameters. There are 8 parameters, and they should be listed as above, delimited by commas. These 8 parameters are:

- tag in the HOL record (XPT)
- subfield of the tag (e.g., $a)
- beginning value of the tag and subfield to look for (Y)
- ending value of the tag and subfield to look for (Y)

- tag in the HOL record that must also be present (852)
- subfield of the tag being searched for (e.g., $b for sublibrary)
- beginning value of the tag and subfield to look for (XLINC)
- ending value of the tag and subfield to look for (XLINC)

These two groups of four represent the retrieval parameters for p_ret_01. The first group indicates that records with `XPT $a Y` should be retrieved. The second group indicates that only those records that have `852 $b XLINC` should be included.

The fifth column indicates the directory in which the output file should be created. For example, `/aleph/u_exl/exu01/scratch`. Note that this directory must exist. The `p_export_01` process will not create the directory.

2. /exu01/tab/tab_expand

Include the following section in the expand table:

```
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!
PRINT-REC  expand_doc_bib_852_1          XPT##,a,Y
```

This expands (embeds) the 852 field from the HOLdings record into the bib record when the record is exported. The parameters listed indicate that only 852s from holdings records that have `XPT $a Y` present should be embedded (expanded) into the bib record.

3. /exu01/tab/tab_fix

Note: There may be more 852s expanded into the bib than are wanted. For example, you may want to export the XLINC holdings (which has XPT=Y), and the XSCI Holdings may also have an XPT=Y, even though you don’t want the XSCI holdings. You can eliminate the unwanted holdings by using a fix in tab_fix.

Include the following section in the table /exu01/tab/tab_fix:

```
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!
XLINC fix_doc_do_file_08             xlinc_only.fix
```

The file `/exu01/tab/import/xlinc_only.fix` consists of the following:

```
!-!!!!!-!!-!-!!!-!!!-!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!>
1 852##                 REPLACE-STRING     $$bXLINC,$$0PICKME$$bXLINC
2 852##                 DELETE-FIELD-COND  N,$$0PICKME
3 852##                 DELETE-SUBFIELD    0
```

The above process removes all non-XLINC 852s.   Libraries are encouraged to copy the fix and replace the string XLINC with their own sublibrary and not to modify this script any more than is necessary. Users who want to customize these scripts should consult the document *“Generic Fix Doc Scripts”*.

4. /exu01/tab/import/xxx.fix

xxx is the target name from column 1 of tab_export. In this example, the file should be XLINC.fix (please note that case is important).

Libraries may choose to “rearrange” the data that is in the bibliographic record. This is done with another fix script. The script does not need to be specified in tab_fix; the p_export_01 process looks directly in the import directory for this script. A sample script XLINC.fix might look like this:

```
!-!!!!!-!!-!-!!!-!!!-!!-!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!
1 852##                 CHANGE-FIELD       949
2 949##                 DELETE-SUBFIELD    c
3 949##                 REPLACE-STRING     $$i,^
4 949##                 CHANGE-SUBFIELD    h c
```

The above fix script changes this:

```
852 $b XLINC $b GEN $h PN567 $i .M444
```

to this:

```
949 $b XLINC $c PN567 .M444
```

Libraries that want to customize this script should consult the document *"Generic Fix Doc Scripts"*.

Table Setup - Multiple Sublibraries

If you would like to retrieve records from more than one sublibrary (for example, XLINC, XLAW and XEDUC), you may choose to group them, using a group name. The above tables would be modified as follows:

1. $alephe_tab/tab_export

```
   1     2    3     4                                                    5
!!!!!--!!!!!--!--!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!--!!!!!!!!!>
MANY   MARC   Y  XPT,a,Y,Y,852,b,GROUP,GROUP                         /aleph/u01/outdir
```

Normally, 852 $b includes a sublibrary (for example, XLINC). In the case where you want to include more than one sublibrary, we need to group them together first with a fix script, this time in the HOLdings library, EXU60. In the example above, "GROUP" was used, but the library can use any string (with a maximum of five characters).

a. /exu60/tab/tab_expand

Include the following section:

```
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!
RET        fix_doc_do_file_08            mapsublibs.fix
```

b. /exu60/tab/import/mapsublibs.fix

The file mapsublibs.fix will look something like this:

```
!-!!!!!-!!-!-!!!-!!!-!!-!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!>
1 852##                 REPLACE-STRING     $$bXLAW,$$bGROUP
2 852##                 REPLACE-STRING     $$bXLINC,$$bGROUP
3 852##                 REPLACE-STRING     $$bXEDUC,$$bGROUP
```

The string "GROUP" is important in the script, as this is what is being used to retrieve the holdings records. 'GROUP' should correspond to parameters 7 and 8 from column 4 of tab_export.

2. /exu01/tab/tab_expand

Include the following section in the expand table (this is no different from the table setup for a single sublibrary)

```
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!
PRINT-REC  expand_doc_bib_852_1          XPT##,a,Y
```

3. /exu01/tab/tab_fix

Include the following section in the fix table:

```
!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!
MANY  fix_doc_do_file_08             linc_law_educ.fix
```

‘MANY’ should correspond to parameter 1 of tab_export.

The file `/exu01/tab/import/linc_law_educ.fix` consists of the following:

```
!-!!!!!-!!-!-!!!-!!!-!!-!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!>
1 852##                 REPLACE-STRING     $$bXLAW,$$0PICKME$$bXLAW
2 852##                 REPLACE-STRING     $$bXLINC,$$0PICKME$$bXLINC
3 852##                 REPLACE-STRING     $$bXEDUC,$$0PICKME$$bXEDUC
4 852##                 DELETE-FIELD-COND  N,$$0PICKME
5 852##                 DELETE-SUBFIELD    0
```

The above eliminates 852s that do not have XLINC, XLAW or XEDUC as the sublibrary.

4. /exu01/tab/import/xxx.fix

xxx is the target name from column 1 of tab_export. In this example, the file should be MANY.fix (please note that case is important).

The fix structure is no different from what was described above.

Table Setup -- Not Using the XPT Flag

If you do not want to use the XPT flag, in other words, you want to export all holdings for a sublibrary or a group of sublibraries, whether or not they are marked for export, you can modify the tables accordingly.

1. $alephe_tab/tab_export

```
   1     2    3     4                                                    5
!!!!!--!!!!!--!--!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!--!!!!!!!!!!>
XARCV  XACV   Y  852,b,XARCV,XARCV,,,,                               /aleph/u01/outdir
```

Only the first set of parameters is needed, because we are not checking to see if XPT=Y is present.

2. /exu01/tab/tab_expand

Include the following section in the expand table:

```
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!
PRINT-REC  expand_doc_bib_852_1           852##,b,XARCV
```

3. /exu01/tab/tab_fix

Only HOLdings records for the XARC sublibrary were selected, so this step is not necessary.

4. /exu01/tab/import/xxx.fix

xxx is the target name from column 1 of `tab_export`. In this example, the file should be `MANY.fix` (please note that case is important).

The fix structure is no different from what was described above.

Running the p_export_01 job

Now that all of the tables are set up, run the p_export_01 batch job.

There are four parameters for p_export_01:

parameter 1: Holdings library that includes the holdings data to be exported.
Example: EXU60
parameter 2: Name of the bib library associated with the holdings library
in parameter1.
Example: EXU01
parameter 3: The target of the export file being created.
Example: XLINC or MANY
parameter 4: A flag that indicates whether step 7 should be run. (This is the step that uses global change functionality to change XPT $$a ‘Y’ to ‘N’.) To specify that this step should be executed, set this flag to ‘XPT’.
Otherwise, it should be omitted.

This job can be submitted via Web services, or from the command line as follows:

Single sublibrary:

```
$ csh -f $aleph_proc/p_export_01 EXU60,EXU01,XLINC,XPT
```

Multiple sublibrary:

```
$ csh -f $aleph_proc/p_export_01 EXU60,EXU01,MANY,XPT
```

The p_export_01 output file is placed in the ./xxx01/files/export/ directory. If there is no ./export directory, it creates one.