

Harvesting Data for Primo in Aleph (publish-06 etc)

- **Article Type:** General
- **Product:** Aleph
- **Product Version:** 20

Description:

How is a pipe prepared for Primo in Aleph?

Resolution:

[1]

Aleph table prerequisites:

(the Aleph table lines shown below are examples, not recommendations, unless specifically stated)

XXX01 / tab / tab_publish

COL1 publish set (only PRIMO-FULL and PRIMO-AVAIL for Primo, they are hard-coded and related to specific software actions)

COL2 logical base or BIB library name (blank means all)

COL4 fix and expand codes (match tab_expand column 1)

COL5 format (e.g., MARC_XML)

(Directory for tar file is now defined when publish-06 is executed)

! 1 2 3 4 5

!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!-!!!!-!!!!!!!!!!!!!!!!!!!!

PRIMO-FULL N PRIMO MARC_XML

PRIMO-AVAIL N AVA MARC_XML

XXX01 / tab / tab_expand

* PUBLISH is not related to Primo. It is used Google Publishing.

! 1 2 3

!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>

PRIMO expand_doc_bib_hol -006,007,008

PRIMO expand_doc_bib_avail THRESHOLD=050;AVA=NB,FR

PRIMO expand_doc_type tab_type_config

PRIMO fix_doc_redo_880

PRIMO expand_doc_del_fields AVA##

* expand_doc_bib_avail ... THRESHOLD=050;AVA=NB,FR

Sets minimum number of items (50 defined here; default = 10) allowed on BIB record checked for availability -- if too many, system responds "check holdings". AVA=NB,FR sets process status values to be included as available in the expand. Only Item Process Statuses listed here are available. All items on loan are also unavailable.

* fix_doc_redo_880

Relevant only if you have parallel 245 fields (one Latinized, and the other, vernacular).

* expand_doc_del_fields

If PRIMO-AVAIL is not being used, there is no need for this line. This setting limits the fields stored in a PRIMO-AVAIL

Z00P to availability fields, but is not used to determine publish-06 output (that is, it does NOT remove BIB fields from publish-06 output). The BIB fields are "deleted", that is, removed from PRIMO-AVAIL in Z00P and from consideration, so only Availability fields are checked.

* To see expand procedures in action, view a record via UTIL-F-4 > doc_expand, and enter the desired expand procedure (e.g., PRIMO).

```
XXX01 / tab / tab_fix
recommended to prevent updating of Z00P due to 005 field change (updated when a BIB record is saved)
! 1 2 3
!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
PRIMO fix_doc_do_file_08 del_005
```

```
XXX01 / tab / import / del_005 (related parameter file)
! 2 3 4 5 6 7 8 9
!-!!!!-!!-!-!!!-!!!-!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!...
1 005 DELETE-FIELD
```

[2]
publish-04 Initial Publishing Process (creates Z00P)

* disk space required:
-- BIB/HOL/AUT/AVA = 20 KB / Z00 record = PRIMO-FULL
-- AVA alone = 8 KB / Z00 record = PRIMO-AVAIL

* Each set creates one Z00P record for each BIB record. In Z00P, PRIMO-FULL records contain all fields, including availability data. In Z00P, PRIMO-AVAIL records contain only the availability data.

* It is a good idea to first drop the Z00P table in the \$usr_library (UTIL-A-17-1) before running publish_04 the first time in case Z00P records already exist and to be sure that the table is properly created. To delete a particular set from Z00P, use publish-05. (Run SQL queries for Z00P is in usr_library)

* publish-04 does not need to be run a second time unless tab_publish is changed or a base that appears in tab_publish is changed in \$alephe_tab/tab_base.eng.

```
* csh -f $aleph_proc/p_publish_04 XXX01,PRIMO-FULL,000000000,999999999,04, > $alephe_scratch/
publish04-20100528.log &
```

* publish_04 will not create a z00p for records which have either of the following:
-- DEL field \$a = Y
-- LDR field position 5 = d

Additional Information

[3] ongoing updates daemon = ue_21 * runs in the BIB library (set by PUB_STARTUP_LIBS in aleph_start.private or aleph_start) * ue_21 should NOT be running during publish_04, since publish_04 creates a Z00P for every record, while ue_21 is processing new records (the logic is similar to ue_01 and index jobs). Be sure to check that ue_21 has restarted after completion of the publish_04 process. * Z07. A Z07 record is created for each new or modified record for Aleph indexing. * Z07P. If tab_publish exists, when a Z07 record is created for a BIB record, a Z07P record is also created. The

Z07P record triggers ue_21. Although Z07P are created from BIB records, their creation can be triggered by changes in: holdings, related authority records, items, loans, returns, hold request status change to S, record deletion, or base change.
* An existing Z00P that satisfies either of the conditions below is updated by adding "deleted" status to the header of the published record () - - DEL field \$a = Y - - LDR field

- **Article last edited:** 4/16/2014