

****Request's Send Method is illegal**** processing held items from other libraries

- **Article Type:** General
- **Product:** Aleph
- **Product Version:** 20, 21, 22, 23

Description:

Local holds work OK; the problem shows up when we process held books that are owned by other colleges (other sublibraries, in our set-up).

When we attempt to check in one of these items with a hold request on it, the system doesn't generate a notice, and displays the error "Request's Send Method is illegal" as a "Remote Service Error (c429 24)".

Before, on 8/11 and 12 (after the upgrade) when we checked it in, the box came up saying "item has holds" but nothing printed for the hold letter slip, and the email apparently wasn't sent; also the screen says "no holds for patron." But looking at the item record it says there are holds, and under Due Date it says "Requested Reshelving." The Holds Request screen shows the patron, and clicking on "Letter" here does print the "Hold letter slip."

Resolution:

"Request's Send Method is illegal" is error 24 in \$aleph_root/error_eng/pc_cir_c0429. (The program calls execute_hold_request, which gets an error code 04, to which pc_cir_c0429 adds 20, thus arriving at 24.)

It is the library's intention that the tab27 Process option always be "01", and the only value in the ABC50 tab27 column 5 is "01". Despite this, SQL shows that values of "0" and null have been assigned:

```
SQL> select z37_send_action, count(*) from z37 group by z37_send_action;
**** Hit return to continue ****
Z37_SEND_ACTION COUNT(*)
-----
                725
1                2528
0                 644
```

I backed up the z37 table and used SQL to change all of these to "1":

```
SQL> update z37 set z37_send_action = 1 where z37_send_action = 0 or z37_send_action is null;
```

but new ones with 0 or null started being written within several minutes.

Their tab27 had lines for each library with that library's sublibrary code in columns 2 and 4, but it did not have all the combinations of sublibraries in these columns.

Adding the following line to handle all the other combinations prevented new Z37s from being written with 0 or null:

```
H ##### ## ##### 01
```

Note: the SQL shown above needed to be run after making this change and restarting the pc_server, in order to correct existing ones.

-
- **Article last edited:** 10/8/2013