

---

## Display order of shelf list with numbers

- **Article Type:** General
  - **Product:** Aleph
  - **Product Version:** 18.01
- 

### Description:

Attached is a snapshot showing the odd display order of shelf list with numbers. As you can see, the display order is not sequential:

```
...  
bldr vert 3105  
bldr vert 3106  
bldr vert 3107  
bldr vert 1  
bldr vert 2  
...
```

### Resolution:

The SHL (Shelf List) index is using tab\_filing #22, which includes the following line:

```
22 F lc_call_no
```

The lc\_call\_no routine is described in KB 16384-10097. The problem is that not all the numbers in the Shelf list are LC call numbers. Certainly these are not.

Here is what the Z11 records look like (in util f/4):

```
01 z11_index \  
02 z11_rec_key \  
03 ind_code .....SHL  
03 filing_text ....bldr vert #3105  
03 sequence_1 .....001402033  
02 z11_doc_number ...001402033  
02 z11_alpha .....L  
02 z11_text .....$$bBLDR$$cVERT$$h3105  
new key = K, exit = Q, continue = RETURN
```

```
01 z11_index \  
02 z11_rec_key \  
03 ind_code .....SHL  
03 filing_text ....bldr vert #3106  
03 sequence_1 .....001402038  
02 z11_doc_number ...001402038  
02 z11_alpha .....L
```

```
02 z11_text .....$$bBLDR$$cVERT$$h3106
new key = K, exit = Q, continue = RETURN
```

```
01 z11_index \
02 z11_rec_key \
03 ind_code .....SHL
03 filing_text ....bldr vert #3107
03 sequence_1 .....001300235
02 z11_doc_number ...001300235
02 z11_alpha .....L
02 z11_text .....$$bBLDR$$cVERT$$h3107
new key = K, exit = Q, continue = RETURN
```

```
01 z11_index \
02 z11_rec_key \
03 ind_code .....SHL
03 filing_text ....bldr vert 1
03 sequence_1 .....001125683
02 z11_doc_number ...001125683
02 z11_alpha .....L
02 z11_text .....$$bBLDR$$cVERT$$h1
new key = K, exit = Q, continue = RETURN
```

```
01 z11_index \
02 z11_rec_key \
03 ind_code .....SHL
03 filing_text ....bldr vert 2
03 sequence_1 .....002573419
02 z11_doc_number ...002573419
02 z11_alpha .....L
02 z11_text .....$$bBLDR$$cVERT$$h2
```

<etc.>

"#" files before "1" or any number.

The abc01 tab\_filing #22 has "pack\_spaces" following "lc\_call\_no":

```
22 F lc_call_no
22 F pack_spaces
```

This is causing the extra space -- which is necessary in this case -- to be eliminated. The distributed usm01 tab\_filing does \*not\* have this. The filing\_lc\_call\_no routine does its own pack\_spaces (prior to inserting the special character) so there is no need for its inclusion after lc\_call\_no.

To test this change you need to restart both the pc\_server and the ue\_01 indexing daemon.

util f/1/3 confirms that the change works:

Enter Filing Procedure (Q to quit)  
Enter values as defined in Col. 1 of tab\_filing: 22  
Enter values D/N/F or Space:  
Enter text (Q to quit) : \$\$\$BLDR\$\$cVERT\$\$h1  
IN -\$\$\$BLDR\$\$cVERT\$\$h1 -  
Load: /exlibris1/utf18/utf\_files/exlibris/aleph/u18\_1/abc01/tab/tab\_filing  
Load: /exlibris/aleph/u18\_1/alephe/unicode/unicode\_case  
OUT -bldr vert 1 -

Enter text (Q to quit) : \$\$\$BLDR\$\$cVERT\$\$h2  
IN -\$\$\$BLDR\$\$cVERT\$\$h2 -  
OUT -bldr vert 2 -

To correct this for all records, you would need to run p\_manage\_05.

---

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