Label printing does not work, problem with tab_label_parse

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

Description:
Table tab_label_parse does not seem to be parsing call numbers according to the routine specified for parse_call_no_lc_1 or parse_call_no_lc_2: we cannot get it to break after the initial alpha element of the classmark (or indeed any other point at which it should break, unless a space is present), i.e. it does not break as described in the header help for parse_call_no_lc_1 or parse_call_no_lc_2. We would prefer parse_call_no_lc_2, but neither works. This is in abc50.

Example used in testing: abc50 item with barcode 71161022 (call number P 353 KOO).

Background. In the past we have used non-standard LC call numbers with a space in the data after the alpha element, as in the example above; we used parse_call_no_lc_3. Recently, we have started following strict LC practice, with no space after the alpha, though we continue to use non-standard filing suffixes at the end (so that the above example becomes K353 KOO). We want to print labels with a break after the alpha element; parse_call_no_lc_1 or parse_call_no_lc_2 should give us this.

Nothing we do will make the prints break as described: it always comes out as

P353
KOO

instead of what we want

P
353
KOO

Only if we restore the space in the data do we get the correct layout. I have checked the Raw XML for these prints and the failure to break properly is present there, so the problem must be within Aleph, not the LABEL_PRINT software. Here is an example of the raw XML:

<call-no-piece-01>P353</call-no-piece-01>
<call-no-piece-02>KOO</call-no-piece-02>

I have tested also on v16 and get the same results there.

I have also edited the item to have a 'longer' classmark with more expected break points (P353.1234.H4 KOO). When set to parse_call_no_lc_3, which is not supposed to break after the alpha element, I get all the expected break points, i.e. at both decimal points and at the space; when set to parse_call_no_lc_1, I do not get *any* of the expected break points, not the one after the alpha nor the second (on decimal preceding a letter), it only breaks when there is a space in the data. parse_call_no_lc_2 does not break after the first alpha, and though it does break on some decimal points, it duplicates data, as shown below (this is the same problem we reported on v16 for parse_call_no_lc_3, which was fixed by v16 fix

#1103).

parse_call_no_lc_1.

<call-no-piece-01>P353.1234.H4</call-no-piece-01>
<call-no-piece-02>KOO</call-no-piece-02>

parse_call_no_lc_2.

<call-no-piece-01>P353</call-no-piece-01>
<call-no-piece-02>.1234.H4</call-no-piece-02>
<call-no-piece-03>P353.1234</call-no-piece-03>
<call-no-piece-04>.H4</call-no-piece-04>
<call-no-piece-05>KOO</call-no-piece-05>

parse_call_no_lc_3.

<call-no-piece-01>P353</call-no-piece-01>
<call-no-piece-02>.1234</call-no-piece-02>
<call-no-piece-03>.H4</call-no-piece-03>
<call-no-piece-04>KOO</call-no-piece-04>

(The fact that I can get different results in these cases confirms that the system is reading changes I make to tab_label_parse.)

Resolution:
Our print_label documentation assumes the presence of $$h and $$i. In addition to the header, the document (ALEPH_Distributors/ALEPH_Customers/17/System Librarian Information/System Librarian Information - Specific Topics/ Printing/How to Set Up Label Printing - 16 and 17.pdf) includes examples that show use of $$h and $$i.

To overcome this issue, we would suggest you to use a new label program. See attached presentation for details on how to set up and use the BIAF label software with Aleph.

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