
p_manage_17: stopping/restarting; last-long-acc-number

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

Description:

In 15.2 p_manage_17 is multi-process and the last-long-acc-number isn't updated until all processes are complete. If you need to stop the job, what's going to happen? Will it just start over at the beginning?

Resolution:

The multi-process p_manage_17 reads the util g/2 last-long-acc-number to see where to start and the last-acc-number to see where to stop.

If you specify 6 processes, then it divides the headings to be processed into 6 cycles.

You can see in the \$data_scratch manage_17 cycles file what range it has assigned to each cycle.

If the job doesn't finish before it needs to be stopped, then basically it's going to start over. Its work will not be saved.

There are two possibilities:

1. Run it as a single process. (So you will know that all the headings it processes from 0 to wherever it gets have been done and then you can manually set the last-long-acc-heading, after you cancel the job.) This presumes that you can tell how far it has gotten. It seems you may not be able.... OR
2. Say that the last-acc-number is 30000000. And you think that your p_manage_17 will be able to do 15 million headings, in multi-process mode, before you have to shut ALEPH down. You could:
 - * set the last-long-acc-number to 15000000
 - * run p_manage_17

This will cause p_manage_17 to process the Z01's from 15000001 to 30000000. Then, the next time you run it (again with a 15-million-record-window before shutdown) you would:

- * stop ue_01 and ue_08, to prevent any new z01s from being created
- * note the current last-acc-number and last-long-acc-number
- * set last-acc-number to 15000000
- * set last-long-acc-number to 0
- * run p_manage_17
- * restore last-acc-number and last-long-acc-number to the values noted in the first step
- * restart ue_01 and ue_08

This will cause p_manage_17 to process Z01s 0 - 15000000.

The advantage to this second method is that it can run multi-process, with 9 processes -- in this case, would mean that it would run 9 times as fast.

KEYWORDS: p-manage-17 manage-17 manage_17 UTIL-G-2

Additional Information

15.2, p_manage_17, multi-process

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