p_ret_01 takes too long

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

Description:
Our p_ret_01 jobs take a very long time and use many system resources. We don't know if it's our search strategies or what.

Resolution:
p_ret_01 reads records sequentially, so it can take a very long time. (See further below for the use of SQL against the z00r as an alternative to p_ret_01.)

If the fields which you are searching are Word- (or Browse-) indexed, then, rather than using p_ret_01, you should use p_ret_03, or do a GUI/Web search and Save the result set to the server. This will be 50 to 100 times faster than p_ret_01.

The speed of p_ret_01 is affected by several variables:

1) If one of the fields that you are retrieving on is indexed, but another one is not, then do p_ret_03 or GUI/Web Search to create a set for the field which *is* indexed and use that set as an input file to p_ret_01;

2) Specifying a p_doc_number_from and p_doc_number_to can greatly reduce the time, since the system number has an Oracle index (z00_id).

3) Specifying a p_update_date_from and/or p_update_date_to will reduce the time, since the z13_update_date which it uses has an Oracle index (z13_id1).

4) Removing unnecessary lines from the $data_tab/tab_expand can reduce the time by as much as 50%.

Specifically, if the query is not using:

* the LOC field, then the "RET expand_doc_bib_loc_usm " can be eliminated;

* the PST field, then all "RET expand_doc_bib_loc_n_x " lines can be eliminated; or

* any HOL fields, then the "RET expand_doc_bib_hol_usm " can be eliminated.

Note: The RET expands are used only by p_ret_01.

The specific query (p_text_from / p_text_to) usually does *not* affect p_ret_01 performance very much.

[The following is an alternative to p_ret_01 posted to the Global Aleph Users Discussion list by Mike Rogers, UT - Knoxville:]

If you have the Z00R table buit in your bib library, you can use this query:

```
SET LINESIZE 14
SET PAGESIZE 0
SET HEADING OFF
SET FEEDBACK OFF
SET PAUSE OFF

SPOOL /exlibris/aleph/u20_1/alephe/scratch/carol

SELECT /*+ DYNAMIC_SAMPLING(2) ALL_ROWS */
Z00R_DOC_NUMBER||'xxx01'
FROM xxx01.Z00R
WHERE Z00R_FIELD_CODE LIKE '740%' and Z00R_TEXT like '%Othello%'
GROUP BY Z00R_DOC_NUMBER||'xxx01';

SPOOL OFF

SET PAUSE ON
SET HEADING ON
SET FEEDBACK ON
SET LINESIZE 24
SET PAGESIZE 80
```

This ran in 48 seconds against a 2.3-milliion record database.

This will output a file called 'carol.lst' to the alephe/scratch directory, which can then be used as input to p_manage_40 (reindexing), p_manage_21 (global-change), etc.

keywords: faq slow performance

Additional Information

faq

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