**In this Document**

|  |  |
| --- | --- |
|  | [Goal](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=528259266981390&parent=SrDetailText&sourceId=3-14571274811&id=822527.1&_afrWindowMode=0&_adf.ctrl-state=chhd4l55j_94" \l "GOAL) |

|  |  |
| --- | --- |
|  | [Solution](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=528259266981390&parent=SrDetailText&sourceId=3-14571274811&id=822527.1&_afrWindowMode=0&_adf.ctrl-state=chhd4l55j_94" \l "FIX) |

|  |  |
| --- | --- |
|  | [Recordings](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=528259266981390&parent=SrDetailText&sourceId=3-14571274811&id=822527.1&_afrWindowMode=0&_adf.ctrl-state=chhd4l55j_94" \l "aref_section21) |

|  |  |
| --- | --- |
|  | [References](https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=528259266981390&parent=SrDetailText&sourceId=3-14571274811&id=822527.1&_afrWindowMode=0&_adf.ctrl-state=chhd4l55j_94" \l "REF) |

## Applies to:

Oracle Database - Enterprise Edition - Version 10.2.0.4 to 12.1.0.2 [Release 10.2 to 12.1]  
 Information in this document applies to any platform.

## Goal

How to use V$PROCESS\_MEMORY and V$PROCESS\_MEMORY\_DETAIL to identify where the memory is growing.  
  
 In Oracle 10.2 and higher exist 2 new views that can be used to find where the memory continue to grow. This views can be used instead of heap dump to find where the memory is growing:  
  
  - V$PROCESS\_MEMORY:  
  
      This view displays dynamic PGA memory usage by named component categories for each Oracle process. This view will contain up to six rows for each Oracle process, one row for:  
       - Java  
       - PL/SQL  
       - OLAP  
       - SQL  
       - Freeable  
       - Other  
  
  - V$PROCESS\_MEMORY\_DETAIL  
     Contain break down of memory allocation for each component.  
     - To activate this view can one of following commands executed:  
        SQL> alter session set events'immediate trace name PGA\_DETAIL\_GET level <PID>';  
        From ORADEBUG:  
        SQL> ORADEBUG SETMYPID;  
        SQL> ORADEBUG DUMP PGA\_DETAIL\_GET <PID>;  
     - To remove all rows in the view run following command:  
        SQL> alter session set events'immediate trace name PGA\_DETAIL\_CANCEL level <PID>';  
        From ORADEBUG:  
        SQL> ORADEBUG DUMP PGA\_DETAIL\_CANCEL <PID>;

## Solution

1.         Find which process is continue to consume more and more memory. This can be found by using the following query:

COLUMN alme     HEADING "Allocated MB" FORMAT 99999D9  
 COLUMN usme     HEADING "Used MB"      FORMAT 99999D9  
 COLUMN frme     HEADING "Freeable MB"  FORMAT 99999D9  
 COLUMN mame     HEADING "Max MB"       FORMAT 99999D9  
 COLUMN username                        FORMAT a15  
 COLUMN program                         FORMAT a22  
 COLUMN sid                             FORMAT a5  
 COLUMN spid                            FORMAT a8  
 SET LINESIZE 300  
 SELECT s.username, SUBSTR(s.sid,1,5) sid, p.spid, logon\_time,  
        SUBSTR(s.program,1,22) program , s.process pid\_remote,  
        s.status,  
        ROUND(pga\_used\_mem/1024/1024) usme,  
        ROUND(pga\_alloc\_mem/1024/1024) alme,  
        ROUND(pga\_freeable\_mem/1024/1024) frme,  
        ROUND(pga\_max\_mem/1024/1024) mame  
 FROM  v$session s,v$process p  
 WHERE p.addr=s.paddr  
 ORDER BY pga\_max\_mem,logon\_time;

            You will get output:

        USERNAME  SID   SPID     LOGON\_TIM PROGRAM  PID\_REMOTE   STATUS    Used MB Allocated MB Freeable MB   Max MB  
         --------- ----- -------- --------- -------- ------------ -------- -------- ------------ ----------- --------  
         TEST       141   3095     08-MAY-09 test    3080         ACTIVE       14.0         15.0          .0     15.0

            Rerun the query:

        USERNAME  SID   SPID     LOGON\_TIM PROGRAM  PID\_REMOTE   STATUS    Used MB Allocated MB Freeable MB   Max MB  
         --------- ----- -------- --------- -------- ------------ -------- -------- ------------ ----------- --------  
         TEST       141   3095     08-MAY-09 test    3080         ACTIVE       29.0         30.0          .0     30.0

2.         User TEST with SID 141 continues to use more and more memory. To get more detailed information in which component is growing can view V$PROCESS\_MEMORY be used.

The following query can be used:

COLUMN category      HEADING "Category"  
 COLUMN allocated     HEADING "Allocated bytes"  
 COLUMN used          HEADING "Used bytes"  
 COLUMN max\_allocated HEADING "Max allocated bytes"  
 SELECT pid, category, allocated, used, max\_allocated  
 FROM   v$process\_memory  
 WHERE  pid = (SELECT pid  
               FROM   v$process  
               WHERE  addr= (select paddr  
                             FROM   v$session  
                             WHERE  sid = 141));

   PID        Category        Allocated bytes Used bytes Max allocated bytes  
    ---------- --------------- --------------- ---------- -------------------  
            22 SQL                      191888     160792             3500976  
            22 PL/SQL                    35448      27912               35448  
            22 Freeable                 262144          0        
            22 Other                  30846677                       30846677

            The query show in which component the process is memory, in this case has Other the highest memory allocation.  
       
             Wait a couple of minutes and run the query again to find in which component the memory is growing:

  PID        Category        Allocated bytes Used bytes Max allocated bytes  
   ---------- --------------- --------------- ---------- -------------------  
           22 SQL                      191856     160792             3500976  
           22 PL/SQL                    35448      27912               35448  
           22 Freeable                 196608          0         
           22 Other                  45657845                       45657845

3.         The process is continue to consuming more memory in Other, to break down this has view V$PROCESS\_MEMORY\_DETAIL to be used. First step is to activate the view by run one of following commands, where "level 22" is "level <PID>" from previous query:

alter session set events'immediate trace name PGA\_DETAIL\_GET level 22'

            or:

ORADEBUG SETMYPID;  
 ORADEBUG DUMP PGA\_DETAIL\_GET 22;

            Wait some minutes to create the first temporary table for the process:

CREATE TABLE tab1 AS  
 SELECT category, name, heap\_name, bytes, allocation\_count,  
        heap\_descriptor, parent\_heap\_descriptor  
 FROM   v$process\_memory\_detail  
 WHERE  pid      = 22  
 AND    category = 'Other';

            Wait some time and collect the information again. Run the command again to get new values in the  
              view:

alter session set events'immediate trace name PGA\_DETAIL\_GET level 22'

or:

ORADEBUG SETMYPID;  
 ORADEBUG DUMP PGA\_DETAIL\_GET 22;

            Wait some minutes than create the second temporary table for the process:

CREATE TABLE tab2 AS  
 SELECT category, name, heap\_name, bytes, allocation\_count,  
        heap\_descriptor, parent\_heap\_descriptor  
 FROM   v$process\_memory\_detail  
 WHERE  pid      = 22  
 AND    category = 'Other';

            The above steps can be execute several times to collect more information during longer period.:

4.         Now can the result from both tables compared to find in which area the memory is increasing, by running following query:

COLUMN category      HEADING "Category"  
 COLUMN name          HEADING "Name"  
 COLUMN heap\_name     HEADING "Heap name"  
 COLUMN q1            HEADING "Memory 1st"  Format 999,999,999,999  
 COLUMN q2            HEADING "Memory 2nd"  Format 999,999,999,999  
 COLUMN diff          HEADING "Difference"  Format S999,999,999,999  
 SET LINES 150  
 SELECT tab2.category, tab2.name, tab2.heap\_name, tab1.bytes q1, tab2.bytes q2, tab2.bytes-tab1.bytes diff  
 FROM   tab1, tab2  
 WHERE  tab1.category  =  tab2.category  
 AND    tab1.name      =  tab2.name  
 AND    tab1.heap\_name =  tab2.heap\_name  
 AND    tab1.bytes     <> tab2.bytes  
 ORDER BY 6 DESC;

Category  Name                     Heap name         Memory 1st   Memory 2nd   Difference  
 --------- ------------------------ --------------- ------------ ------------ ------------  
     Other     permanent memory         kolaGetRfcHeap    30,059,544   36,416,968   +6,357,424  
     Other     free memory              session heap       6,475,848    7,638,528   +1,162,680  
     Other     free memory              kolaGetRfcHeap       479,832      586,856     +107,024  
     Other     kolasl: kolaslCreateCtx  koh dur heap d       441,304      528,600      +87,296  
     Other     kolaGetRfcHeap:sheap     koh dur heap d       402,152      481,688      +79,536  
     Other     kolraloc-1               kolr heap ds i       206,800      247,728      +40,928  
     Other     free memory              kolr heap ds i         8,608       35,640      +27,032  
     Other     free memory              pga heap              96,192      112,648      +16,456  
     Other     free memory              top uga heap          54,392       63,752       +9,360  
     Other     free memory              lpxHeap subhea       198,088      200,912       +2,824  
     Other     free memory              Alloc server h         7,416        8,928       +1,512  
     Other     permanent memory         qmxtkAggCrtAgg         4,016        4,120         +104  
     Other     free memory              koh dur heap d       178,120      175,840       -2,280  
     Other     free memory              koh-kghu call         17,456        3,120      -14,336  
     Other     kollalo2                 koh-kghu sessi        41,032        8,216      -32,816

            The query show that the largest memory increases is in kolaGetRfcHeap.

The output from view V$PROCESS\_MEMORY\_DETAIL can be compared with heapdump.  
 Output from view V$PROCESS\_MEMORY\_DETAIL:

COLUMN heap\_name        HEADING "heap name"  
 COLUMN name             HEADING "Type"  
 COLUMN allocation\_count HEADING "Count"  
 COLUMN bytes            HEADING "Sum"  
 COLUMN avg              HEADING "Average" FORMAT 99999D99  
 SELECT heap\_name, name, allocation\_count, bytes, bytes/allocation\_count avg  
 FROM   tab2  
 WHERE  heap\_name = 'kolaGetRfcHeap';

heap name       Type                            Count        Sum   Average  
 --------------- -------------------------- ---------- ---------- ---------  
 kolaGetRfcHeap  free memory                      5886     586856     99.70  
 kolaGetRfcHeap  permanent memory                 8949   36416968   4069.38

Heapdump after been processed by heap.awk::

---> HEAP DUMP heap name="kolaGetRfcHeap"  desc=0x2ae989b31fd8  
               Type           Count             Sum         Average  
               ~~~~           ~~~~~             ~~~         ~~~~~~~  
               perm               4           16240         4060.00  
               free               3             312          104.00

|  |  |
| --- | --- |
| |  | | --- | | Recordings | |