**How to Determine the Blocking Session for Event: 'cursor: pin S wait on X' (Doc ID 786507.1)**

## Goal

This note helps find the blocking session for mutex related wait event "cursor: pin S wait on X"  
  
 To Troubleshoot this event see:

[Document 1377998.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&amp;sourceId=786507.1&amp;id=1377998.1) Troubleshooting: Waits for Mutex Type Events  
[Document 1349387.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&amp;sourceId=786507.1&amp;id=1349387.1) Troubleshooting 'cursor: pin S wait on X' waits  
  
[Document 1356828.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&amp;sourceId=786507.1&amp;id=1356828.1) FAQ: 'cursor: mutex ..' / 'cursor: pin ..' / 'library cache: mutex ..' Type Wait Events  
  
[Document 1377446.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&amp;sourceId=786507.1&amp;id=1377446.1) Troubleshooting Performance Issues

## Solution

**Cursor: pin S wait on X.**

A session waits on this event when requesting a mutex for shareable operations  related to pins (such as executing a cursor), but the mutex cannot be granted because it is being held exclusively by another session (which is most likely parsing the cursor).

The column P2RAW in v$session or v$session\_wait gives the blocking session for wait event  cursor: pin S wait on X.

The top bytes of p2raw is the blocker.  It is in hex so needs to be converted in decimal.

SQL> select p2raw from v$session where event = 'cursor: pin S wait on X';  
  
 P2RAW  
 ----------------  
 0000001F00000000  
  <SID>  <RefCnt>

The top bytes of p2raw is the blocker.  
 Taking 0000001F (the first 8 bytes) and converting to decimal gives session id 31.

Or simply:

SQL> select p2raw,to\_number(substr(to\_char(rawtohex(p2raw)),1,8),'XXXXXXXX') sid  
      from v$session  
      where event = 'cursor: pin S wait on X';  
  
 P2RAW               SID  
 ----------------    ---  
 0000001F00000000     31

**64 bit platforms  
8 bytes are used.  
 Top 4 bytes hold the session id (if the mutex is held X)  
 Bottom 4 bytes hold the ref count (if the mutex is held S).  
  
32 bit platforms  
 4 bytes are used.  
 Top 2 bytes hold the session id (if the mutex is held X)  
 Bottom 2 bytes hold the ref count (if the mutex is held S).**

SQL> select p1, p2raw, count(\*) from v$session  
      where event ='cursor: pin S wait on X'  
      and wait\_time = 0  
      group by p1, p2raw;

* p1 = the mutex Id  
   This has the same definition as v$mutex\_sleep\_history.mutex\_identifier
* p2raw = holding Session Id | Ref Count  
   The most significant bytes always store the Holding Session Id (Holding SId).  
   The least significant bytes always store the Ref Count.

The blocking session can be queried to see what it is doing and if anyone is blocking it.

SQL> select sid,serial#,SQL\_ID,BLOCKING\_SESSION,BLOCKING\_SESSION\_STATUS,EVENT  
      from v$session where SID=31;

As a result of [Bug 7568642](https://support.oracle.com/epmos/faces/BugDisplay?parent=DOCUMENT&amp;sourceId=786507.1&amp;id=7568642) BLOCKING\_SESSION EMPTY FOR "CURSOR: PIN S WAIT ON X"  the blocking\_session is not populated in 10.2.The bug is fixed in 11g R1.

In 11g, the blocking session can be found directly using the following sql:

SQL> select sid,serial#,SQL\_ID,BLOCKING\_SESSION,BLOCKING\_SESSION\_STATUS,EVENT  
      from v$session where event ='cursor: pin S wait on X'

 SID SERIAL# SQL\_ID        BLOCKING\_SESSION BLOCKING\_SESSION\_STATUS EVENT   
 ---- ------- ------------- ---------------- ----------------------- ----------   
  125    8190 3d3pd7g7dwuf6              135 VALID                   cursor: pin S wait on X

**Likely Causes**

One of the most likely causes of cursor: pin S wait on X is high parsing time. Therefore the reason for the high parse time should be investigated.