**Alert.log shows ORA-609 with TNS-12537: TNS:connection closed (Doc ID 1538717.1)**

**In this Document**

|  |  |
| --- | --- |
|  | [Symptoms](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=obsyz0juf_4&_afrLoop=83101864603963%20\l%20SYMPTOM) |
|  | [Changes](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=obsyz0juf_4&_afrLoop=83101864603963%20\l%20CHANGE) |

|  |  |
| --- | --- |
|  | [Cause](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=obsyz0juf_4&_afrLoop=83101864603963%20\l%20CAUSE) |
|  | [Solution](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=obsyz0juf_4&_afrLoop=83101864603963%20\l%20FIX) |

|  |  |
| --- | --- |
|  | [References](https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=obsyz0juf_4&_afrLoop=83101864603963%20\l%20REF) |

graphic

**APPLIES TO:**

Oracle Net Services - Version 11.2.0.1 to 11.2.0.4 [Release 11.2]  
 Oracle Net Services - Version 12.1.0.2 to 12.1.0.2 [Release 12.1]  
 Information in this document applies to any platform.

**SYMPTOMS**

Alert log shows failed incoming connection:

    Fatal NI connect error 12537, connecting to:  
      (LOCAL=NO)  
       
       VERSION INFORMATION:  
         TNS for Linux: Version 11.2.0.3.0 - Production  
         Oracle Bequeath NT Protocol Adapter for Linux: Version 11.2.0.3.0 - Production  
         TCP/IP NT Protocol Adapter for Linux: Version 11.2.0.3.0 - Production  
       Time: 26-FEB-2013 02:23:51  
       Tracing not turned on.  
       Tns error struct:  
         ns main err code: 12537  
       
     TNS-12537: TNS:connection closed  
         ns secondary err code: 12560  
         nt main err code: 0  
         nt secondary err code: 0  
         nt OS err code: 0  
     opiodr aborting process unknown ospid (28725) as a result of ORA-609

**CHANGES**

Changes in database server load, client connect descriptor, changes in network infrastructure (firewall configuration).

**CAUSE**

First an explanation of this kind of errors.

The message  
    opiodr aborting process unknown ospid (.....) as a result of ORA-609  
 is just a notification that oracle database closed (aborted) a dedicated process because of ORA-609.  
  
 ORA-609 means  "could not attach to incoming connection" so the database process was 'aborted' (closed) because it couldn't attach to the incoming connection passed to it by the listener.  
 The reason for this is found in the sqlnet error stack, in our case is:  
    *TNS-12537: TNS:connection closed.  
 Basically the dedicated process didn't have a client connection anymore to work with.  
  
 As a big picture, these are the steps for a client connection:*

1. Client initiates a connection to the database so it connects to the listener
2. Listener starts (fork) a dedicated database process that will receive this connection (session)
3. After this dedicated process is started, the listener passes the connection from the client to this process
4. The server process takes the connection from the listener to continue the handshake with the client
5. Server process and client exchange information required for establishing a session (ASO, Two Task Common, User logon)
6. Session is opened

In the case of the above error the connection from the client was closed somewhere between 3. and 4. So when the dedicated process tries to communicate with the client it finds that connection closed.  
  
  
 To determine the client which hit this problem we can try to match the timestamp of the error from alert log with an entry in listener.log, but this might be difficult in case of a loaded listener with many incoming connections per second.  
 Server sqlnet trace will not provide any information about the client.  
  
 We can enable sqlnet server trace to catch the error (the match is done based on the ospid found in sqlnet server trace file name and the line with ORA-609 error):

    nscon: doing connect handshake...  
     nscon: recving a packet  
     nsprecv: entry  
     nsprecv: reading from transport...  
     nttrd: entry  
     nttrd: exit  
     ntt2err: entry  
     ntt2err: Read unexpected EOF ERROR on 15    <<<<<<< error  
     ntt2err: exit  
     nsprecv: error exit  
     nserror: entry  
     nserror: nsres: id=0, op=68, ns=12537, ns2=12560; nt[0]=507, nt[1]=0, nt[2]=0; ora[0]=0, ora[1]=0, ora[2]=0  
     nscon: error exit  
     nsdo: nsctxrnk=0  
     nsdo: error exit  
     nsinh\_hoff: error recving request

Server process got an unexpected End Of File on the connection socket passed from the listener.

**SOLUTION**

It is often possible to eliminate this error by increasing the following sqlnet.ora file value for SQLNET.INBOUND\_CONNECT\_TIMEOUT.  The default is 60 seconds.  
 This parameter is set in the database environment in RDBMS\_HOME/network/admin.  
  
 Example:   
  
 SQLNET.INBOUND\_CONNECT\_TIMEOUT=300

See the following [Note  1116960.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1538717.1&id=1116960.1)  11g: ORA-609 TNS-12537 and TNS-12547 or TNS-12170 in 11g Alert.log

Several possible situations can cause this to happen:

* client changed its mind and closed the connection immediately after initiating it
* client crashed
* firewall kills the connection
* some oracle timeout set on client

From all the above causes only the last one can be checked by Oracle Support, because all the others are outside oracle product, they are not under our control/possibility for investigation.  
  
 Because the entry from listener.log contains only CONNECT\_DATA and CID related information we need to check the client configuration for any sqlnet  timeouts:

* possible timeouts in sqlnet.ora in client oracle home:

    sqlnet.outbound\_connect\_time  
     sqlnet.recv\_timeout  
     sqlnet.send\_timeout  
     tcp\_connect\_timeout

* possible timeout in client connect descriptor (hardcoded in client application or in client tnsnames.ora):

    connect\_timeout

Important note:  If the 12537 or ORA-609 errors are accompanied by any other failure posted to the alert.log or they are persistent and very frequent,  
 changing the connect timeout setting will not likely help.   These errors can be thrown in the alert when other issues are present.   It is important to  
 check for other underlying issues when this error is posted along with other messages or other performance related issues.  
  
  
 Some examples:   
  
 See [Note 13691225.8](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1538717.1&id=13691225.8)  Bug 13691225 - File handle leak for processes accessing /proc/stat  
 ORA-00600: internal error code, arguments: [ksb\_shut\_detached\_process3], [MMNL], [10], [300], [], [], [], [], [], [], [], []  
  
 See also:   [Note 1532481.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1538717.1&id=1532481.1)  11g: Persistent TNS-12537/ORA-609 Errors & LSNRCTL Hangs

**REFERENCES**

[NOTE:13691225.8](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1538717.1&id=13691225.8) - Bug 13691225 - File handle leak for processes accessing /proc/stat  
[NOTE:1532481.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1538717.1&id=1532481.1) - 11g: Persistent TNS-12537/ORA-609 Errors & LSNRCTL Hangs  
[NOTE:1116960.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1538717.1&id=1116960.1) - 11g: ORA-609 TNS-12537 and TNS-12547 or TNS-12170 in 11g Alert.log