

## Oracle reporting "ORA-00020: maximum number of processes (%s) exceeded"

- **Article Type:** General
  - **Product:** Aleph
  - **Product Version:** 20, 21, 22, 23
- 

### Description:

Oracle connections are being refused on our Aleph system.

```
*** util_c_04 - check ABC50 lock status ***  
Oracle error: handle_connection  
ORA-00020: maximum number of processes (%s) exceeded
```

### Resolution:

The maximum number of oracle servers (an oracle server provides a connection from a client to the database... e.g. sqlplus client -> oracle server -> database) is limited by a parameter which is called "processes". You can "see" this parameter either in sqlplus: "show parameter processes" or in the spfile which is located in \$ORACLE\_HOME/dbs/spfile<instance>.ora

Any connection to the oracle database is taking one allowed server from this maximum number. To see where the problem is do the following:

1.) Proof: Check for oracle server processes: `ps -eaf | grep -c <instance name>` ["echo \$ORACLE\_SID" gives you the instance name]

```
ps -eaf | grep -c oraclealeph18
```

```
123
```

In this case we see 123 processes.

2.) Check processes in ALEPH: `ps -eaf | grep -v http | grep -c a18_2`  
(grep -v http ---- take out apache because this has no connection to oracle)  
(grep -c a18\_2 ---- count everything from ALEPH)

In this case we see 94(!) processes and it seems that the most of them are sip2 procs

3.) Check for sip2: `ps -eaf | grep -v http | grep -c sip2`  
-> 43 ---> we got it.

4.) Solution: shutdown and startup SIP2 services to solve the current problem,

---

If you want to modify "processes" in Oracle do this:  
(Let's say in your case the parameter is set to 150)

Connect to oracle as sysdba:

From your ALEPH account:

```
sqlplus /nolog
connect aleph_dba/aleph_dba as sysdba
alter system set processes=200 scope = spfile;
exit
```

Stop and start the database.

150 and 200 is just an example, it depends on what you found by executing my example in the step before.

What does it mean: The alter system command does write a new value for processes to the parameter file of the oracle instance. This will be activated when the database is re-started, because "processes" can't be modified when oracle is running.

Note: In Oracle 9 (Aleph 17 and lower) the number of processes is defined in \$ORACLE\_HOME/dbs/initaleph0.ora. See also Article 000036727 ("[Number of Oracle processes exceeded](#)")

---

If also modifying the "processes"-parameter in Oracle does not solve the problem and the problem occur again, please check if there is a general problem with your SC machines.

You can check the sip2\_server-logfiles (in \$LOGDIR).

Each "Initialized"-message indicates that a telnet-connection to the SIP2-server by a client has been started. If everything works fine, the client has to close this connection afterwards (this is the "Client Closed Connection"-message). The difference between the "'Initialized from Address' -messages and the "Client Closed"-messages is the number of current open client sessions (that are occupying ORACLE sessions).

In you example on a customer machine we saw that the client has opened 35433 (!! ) connections without closing any connection:

```
alephprod-a16(1) >>grep -c 'Initialized from Address' sip2_server_5333.log.0909.1145
35433
alephprod-a16(1) >>grep -c 'Client Closed Connection' sip2_server_5333.log.0909.1145
0
```

That means that 35433 SIP2-server-processes are running and occupying ORACLE-connections.

Please check with the vendor of your SC machine if there is a problem. Also there might be network problems.

---

## Additional Information

Oracle, number of processes

---

- **Article last edited:** 21-Jul-2014