How to use this document

Each wait scenario will be given , along with key points in the systemstate which can be used to match to the corresponding entry
in your own systemstates. It will then give you examples of matching holders (ie what you need to find in the rest of the systemstate
to be able to identify who is blocking)

What is a Systemstate ?

A systemstate is made up of the processstate of each process in the instance found at the time the systemstate was called for. Each
processtate is made up of SO (State Objects) which hold details of the state of current objects owned by each PROCESS.

How to Navigate through a systemstate

What you need to do is start by determining what most session are waiting for (or in the case of a session you know
is blocked, the PROCESS number of the process). So - you will now have either a PROCESS XX or a , for example, 'latch free'
which you need to begin with. What you then need to do is navigate (through vi or a windows editor) and find either
PROCESS XX or the first example of 'latch free'. If you are using PROCESS XX then you now need to find what the process
is waiting for. You will now have :-

PROCESS XX waits for YYYYYYY

What you then need to do is find, by using this guide, the SO for the resource the session is waiting for and then find (by
searching back from that point) the PROCESS XX of that session. You now have:-

PROCESS XX waits for YYYYYYY
PROCESS YY holds YYYYYYY

You then begin again, finding the resource it is waiting
for (if any) and that resources holder. Eventually you will come to a process which is on the CPU (last waited) or you will
have navigated back to a PROCESS you already know about. In the case of

the process which is on the CPU you will then need
to get an errorstack to determine why it is blocking. In the case of a 'deadlock' you will now have a dependency tree of the form:-

PROCESS XX waits for YYYYYYY
PROCESS YY holds YYYYYYY and waits for ZZZZZZZZ
***PROCESS ZZ holds ZZZZZZZ** ... etc etc*

Common wait scenarios and corresponding Entries

1 - Enqueues

```
PROCESS 41
...
waiting for ’enq: TX - row lock contention’ blocking sess=0x39b3a5c90
seq=152 wait_time=0 seconds since wait
started=796
name|mode=54580006, usn<<16 | slot=20009, sequence=1fa04
54580006 is split into ASCII 54 + ASCII 58 (TX) + Mode 0006 (X) ...
```

To find more details on the enqueue, simply do a search for the string 'req:' (searching DOWN)　向下搜索 **req:**

```
SO: 39ad80d60, type: 5, owner: 393cb85e0, flag: INIT/-/-/0x00
(enqueue) TX-00020009-0001FA04 DID: 0001-0029-00000090
lv: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 flag: 0x6
res: 39aef20c8, req: X, prv: 39aef20e8, own: 39b383aa8, sess: 39b383aa8,
proc: 39b7384f0
```

Now you have the enqueue name as a string(**TX-00020009-0001FA04**) which you can use to search for the HOLDER:-　直接搜索 **TX-00020009-0001FA04**

```
(enqueue) TX-00020009-0001FA04 DID: 0001-002E-00000014
lv: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 flag: 0x6
res: 39aef20c8, mode: X, prv: 39aef20d8, own: 39b3a5c90, sess: 39b3a5c90,
proc: 39b73ac78
```

We can see we hold the enqueue (mode:X) in a incompatible mode to the req:X request...

2 - Rowcache locks

```
PROCESS 19:
```

```
...
waiting for 'row cache lock' blocking sess=0x0 seq=2174 wait_time=0
cache id=7, mode=0, request=3 *
We do not hold it currently (mode=0), but want it in Shared (mode=3) ...
----------------------------------------------------------------------
-----------
SO: 7000000c6de7678, type: 48, owner: 7000000a6c97cf8, flag:
INIT/-/-/0x00
row cache enqueue: count=1 session=7000000a660b8b0
object=7000000eedc13a0, request=S*Here we see the request is Shared(S)
savepoint=2148
row cache parent object: address=7000000eedc13a0
cid=7(dc_users)*dc_users is the cache type indicated by 7
hash=2a057ebe typ=9 transaction=7000000c42297a0 flags=00000002
own=7000000eedc1480[7000000c6de8518,7000000c6de8518]
wat=7000000eedc1490[7000000c6de7568,7000000c6deed98] mode=X *The holder
has it in this mode
status=VALID/-/-/-/-/-/-/-/-
request=N release=TRUE flags=0
```

To find the HOLDER, search for object,MODE of holder
(ie **object=7000000eedc13a0, mode=X**):-

```
SO: 7000000c6de84e8, type: 48, owner: 7000000c42297a0, flag:
INIT/-/-/0x00
row cache enqueue: count=1
session=7000000a6702710  object=7000000eedc13a0, mode=X*This confirms the
Mode we thought the holder had (X)
savepoint=109
row cache parent object: address=7000000eedc13a0 cid=7(dc_users)
hash=2a057ebe typ=9 transaction=7000000c42297a0 flags=00000002
own=7000000eedc1480[7000000c6de8518,7000000c6de8518]
wat=7000000eedc1490[7000000c6de7568,7000000c6df1b08] mode=X
status=VALID/-/-/-/-/-/-/-/-
request=N release=TRUE flags=0
instance lock id=QH 00000440 00000000
set=0, complete=FALSE
set=1, complete=FALSE
set=2, complete=FALSE
data=
```

3 - Library Cache Pins (10G - Mutexes)

```
PROCESS 16:
```

```
waiting for 'cursor: pin S wait on X' blocking sess=0x0 seq=58849
wait_time=0 seconds since wait started=0
idn=535d1a6c, value=c1600000000, where|sleeps=5003f2428
```

To find more details use the idn=XXXXXX to search down in the systemstate (ie **idn=535d1a6c**)

```
KGX Atomic Operation Log 7000002e5b9d160
Mutex 7000002b8e92268(3094, 0) idn 535d1a6c oper GET_SHRD  *We can see (a)
That SID 3094 holds it (3094,0) and (b) we want it in Shared (GET_SHRD)
Cursor Pin uid 2489 efd 0 whr 5 slp 58733
opr=2 pso=70000028c47def0 flg=0
pcs=7000002b8e92268 nxt=0 flg=34 cld=3 hd=70000030d6c6eb0
par=7000002eefe64d0
ct=31 hsh=0 unp=0 unn=0 hvl=b825a4d0 nhv=1 ses=700000309b42600
hep=7000002b8e922e8 flg=80 ld=1 ob=7000002de49f8a0 ptr=70000022cf39db8
fex=70000022cf390c8
```

To find the HOLDER, search for idn XXXXXXX oper until you find one which is held (ie not GET_XXX)(ie **idn 535d1a6c oper**):-

```
KGX Atomic Operation Log 7000002cd934270
Mutex 7000002b8e92268(3094, 0) idn 535d1a6c oper EXCL  *We can see SID 3094
holds in Exclusive (EXCL)
Cursor Pin uid 3094 efd 0 whr 7 slp 0
opr=3 pso=7000002a71c4180 flg=0
pcs=7000002b8e92268 nxt=0 flg=34 cld=3 hd=70000030d6c6eb0
par=7000002eefe64d0
ct=31 hsh=0 unp=0 unn=0 hvl=b825a4d0 nhv=1 ses=700000309b42600
hep=7000002b8e922e8 flg=80 ld=1 ob=7000002de49f8a0 ptr=70000022cf39db8
fex=70000022cf390c8
```

4 - Library Cache Pins (Pre 10G - non mutex)

```
PROCESS 20:

waiting for 'library cache pin' blocking sess=0x0 seq=575 wait_time=0
handle address=c00000006c0f8490, pin address=c0000000689b19a8,
10*mode+namespace=14
```

To find more details use the handle=XXXXXX to search down in the systemstate (ie **handle=c00000006c0f8490**) and you will see a 'request' line

```
SO: c0000000689b19a8, type: 34, owner: c00000006cf85e80, flag:
INIT/-/-/0x00
LIBRARY OBJECT PIN: pin=c0000000689b19a8 handle=c00000006c0f8490
request=S lock=c00000006d00e218  *We can see we want it in Shared (S)
user=c00000005eeafeb0 session=c00000005eeafeb0 count=0 mask=0000
savepoint=17 flags=[00]
```

To find the HOLDER, search for 'handle=XXXXXXXXX mode' oper until you find one which is held in an incompatible mode(ie **handle=c00000006c0f8490 mode**):-

```
SO: c00000006b1f4780, type: 34, owner: c0000000699758e8, flag:
INIT/-/-/0x00
LIBRARY OBJECT PIN: pin=c00000006b1f4780 handle=c00000006c0f8490 mode=X
lock=c00000006b6c40a0  *We hold it in Exclusive (X)
user=c00000005edf0f48 session=c00000005edf0f48 count=1 mask=0001
savepoint=49 flags=[00]
```

5 - Library Cache Lock

```
PROCESS 35:

waiting for 'library cache lock' blocking sess=0x0 seq=35844 wait_time=0
seconds since wait started=14615
handle address=70000030de975a8, lock address=70000026947e190,
100*mode+namespace=12d
```

To find more details use the handle address in the form handle=address to search down in the systemstate (ie **handle=70000030de975a8**)

```
SO: 70000026947e190, type: 53, owner: 700000308d726f0, flag:
INIT/-/-/0x00
LIBRARY OBJECT LOCK: lock=70000026947e190 handle=70000030de975a8
request=X  *We want it in Exclusive (X)
call pin=0 session pin=0 hpc=0000 hlc=0000
htl=70000026947e210[7000002b333ffe8,7000002b333ffe8]
htb=7000002b333ffe8 ssga=7000002b333f2a0
user=700000307a7ca68 session=700000307a7ca68 count=0 flags=[0000]
savepoint=0x23e411
LIBRARY OBJECT HANDLE: handle=70000030de975a8 mtx=70000030de976d8(0)
cdp=0
name=ACSELP.POLIZA  *This is the object we are trying to lock
```

To find the HOLDER, search for 'handle=XXXXXXXXXX mode=' until you find one which is held (but not in NULL)(ie **handle=70000030de975a8 mode=**):-

```
SO: 700000288b03ae0, type: 53, owner: 7000002cc697468, flag:
INIT/-/-/0x00
LIBRARY OBJECT LOCK: lock=700000288b03ae0 handle=70000030de975a8
mode=S  *We hold in in Shared (S)
call pin=0 session pin=0 hpc=0000 hlc=0000
htl=700000288b03b60[7000002a179a1a8,7000002b3800878]
htb=7000002b3800878 ssga=7000002b37ffb30
user=70000030fafab00 session=70000030fafab00 count=1 flags=[0000]
savepoint=0x417
LIBRARY OBJECT HANDLE: handle=70000030de975a8 mtx=70000030de976d8(0)
cdp=0
name=ACSELP.POLIZA  *This confirms the object
```

6 - Latch free

```
PROCESS 8:

waiting for 'latch free' blocking sess=0x0 seq=4577 wait_time=0
address=99ff60018, number=9d, tries=0  *9d is the latch# from v$latchname in
HEX
```

If you look towards the top of the PROCESS dump you will see the exact latch we are waiting for and even who holds it:

```
waiting for 99ff60018 Child library cache level=5 child#=3
Location from where latch is held: kglic: child
Context saved from call: 26
state=busy
possible holder pid = 127 ospid=23086  *This tell us PROCESS 127 (ospid:23086)
holds it
wtr=99ff60018, next waiter 9993858b8
```

So - PROCESS 127 holds it. If we now go to PROCESS 127 we will see :-

```
holding 99ff60018 Child library cache level=5 child#=3
Location from where latch is held: kglic: child
Context saved from call: 26
state=busy
```

Other useful information

If you wish to find what object a handle refers to then use the handle=XXXXXXXXXX until you come across the LIBRARY OBJECT HANDLE. ie**handle=c00000006c0f8490**:-

```
LIBRARY OBJECT HANDLE: handle=c00000006c0f8490
name=SELECT USER FROM DUAL  *This is the name of the handle
hash=cd1ceca0 timestamp=03-23-2007 09:00:00
namespace=CRSR flags=RON/TIM/PN0/SML/[12010000]*It is a CURSOR (CRSR).. but
we can tell that by the name!
```