#SHAREorg

SHARE
Technology • Connections • Results

CICS Extreme Debugging – DB2 Attachment

Ed Addison
IBM

August 9, 2012
Session Number 11445

SHARE in Anaheim 2012

Agenda

- **Resource Definitions**
- **Documentation Needed and Time Considerations**
- **CICS and DB2 VERBEXIT Displays for Debugging**
- **Problem One – Hang**
- **Problem Two – AD2U Abend**

Resource Definitions

CICS DB2 Resource Definitions

- DB2CONN - defines the attributes of the connection between CICS and DB2, and of the pool threads and command threads used with the connection.
  - Only one DB2CONN can be installed in a CICS system at any one time.
  - A DB2CONN must be installed before any DB2ENTRY or DB2TRAN definitions
- DB2ENTRY - defines the attributes of entry threads used by the CICS DB2 attachment facility
  - A transaction, or a group of transactions, can be associated with the DB2ENTRY
    - A group of transactions may be represented by the use of one or more wildcard characters
  - Further transactions can be associated with a DB2 entry by defining a DB2TRAN
- DB2TRAN - Defines a transaction, or a group of transactions, associated with a DB2ENTRY
  - These are additional to the transactions specified in the DB2ENTRY itself

DB2CONN – Connection Parameters

- **CONNECTERROR({SQLCODE|ABEND})** Specifies the way that the information, that CICS is not connected to DB2 is returned to the application
  - **ABEND** - The application abends with abend code AEY9.
  - **SQLCODE -** The application receives a -923 sqlcode. SQLCODE cannot be specified if STANDBYMODE is set to NOCONNECT.

- **DB2GROUPID(*name*)** Specifies the group ID of a data sharing group of DB2 subsystems. You cannot specify both DB2GROUPID and DB2ID-- the priorities are as follows:
  - Specifying a DB2GROUPID blanks out any DB2ID that is already set in the DB2CONN definition.
  - If you attempt to specify both a DB2GROUPID and a DB2ID on the same CEDA panel, the DB2ID is used.
  - If an individual subsystem's DB2ID is specified in a CEMT or EXEC CICS SET DB2CONN command, or in a DSNC STRT command, this overrides any DB2GROUPID attribute that is set in the installed DB2CONN definition.

- **DB2ID(*name*)** Specifies the name of the DB2 subsystem to which the CICS DB2 attachment facility is to connect.

- **MSGQUEUE1({*CDB2*|*tdqueue*}** Specifies the first transient data destination to which unsolicited messages from the CICS DB2 attachment facility are sent. This first destination cannot be blank.

- **NONTERMREL({YES|NO})** Specifies whether or not a non-terminal transaction releases threads for reuse at intermediate syncpoints.
  - **NO** Non-terminal transactions do not release threads for reuse at intermediate syncpoints.
  - **YES** Non-terminal transactions release threads for reuse at intermediate syncpoints.

DB2CONN – Connection Parameters

- **PURGECYCLE({***0***|***mm***},{** *30***|***ss***})** Specifies the duration, in minutes and seconds, of the purge cycle for protected threads.

- **RESYNCMEMBER({YES|NO})** If you are using group attach, use the RESYNCMEMBER attribute to select the strategy that CICS adopts if outstanding units of work are being held for the last DB2 data sharing group member to which CICS was connected.
  - **YES** indicates that if outstanding units of work are held, you require resynchronization with the last DB2 data sharing group member to which CICS was connected.
  - **NO** indicates that you do not require resynchronization. CICS makes one attempt to reconnect to the last connected DB2 data sharing group member.

- **SIGNID(***name***)** Specifies the authorization ID to be used by the CICS DB2 attachment facility when signing on to DB2 for pool and DB2ENTRY threads that specify AUTHTYPE(SIGN).
  - **Note:** If you specify a user ID on the SIGNID attribute, CICS performs a surrogate user check against the user ID performing the installation. Similarly, the CICS region user ID is subject to a surrogate user check during group list installation on a CICS cold or initial start.

- **STANDBYMODE({ RECONNECT|CONNECT|NOCONNECT})** Specifies the action to be taken by the CICS DB2 attachment facility if DB2 is not active when an attempt is made to connect CICS to DB2.
  - **CONNECT** Specifies that the CICS DB2 attachment facility is to wait in 'standbymode' for DB2 to become active. If the connection is made, and DB2 subsequently fails, the CICS DB2 attachment facility terminates.
  - **NOCONNECT** Specifies that the CICS DB2 attachment facility is to terminate.
  - **RECONNECT** Specifies that the CICS DB2 attachment facility is to go into 'standby mode' and wait for DB2. If DB2 subsequently fails after the connection is made, the CICS DB2 attachment facility reverts to 'standby mode', and CICS subsequently reconnects to DB2 when DB2 recovers.

DB2CONN – Connection Parameters

- **STATSQUEUE({*CDB2*|*tdqueue*})** Specifies the transient data destination for CICS DB2 attachment facility statistics produced when the CICS DB2 attachment facility is shut down.
- **TCBLIMIT({12|*value*})** Specifies the maximum number of TCBs that can be used to process DB2 requests. The default is 12. The minimum number is 4 and the maximum is 2000..
  - The TCBLIMIT value controls the total number of threads for the CICS region. For this reason, the recommended value for TCBLIMIT is the sum of all the thread limit values (that is, the sum of all THREADLIMIT attributes on the DB2 connection and DB2 entry resource definitions, plus the COMTHREADLIMIT value on the DB2 connection definition) up to the limit of 2000.
- **THREADERROR({N906D|N906|ABEND})** Specifies the processing that is to occur following a create thread error.
  - **ABEND** When the first SQL error is detected, CICS takes a transaction dump for abend code AD2S, AD2T, or AD2U, depending on the type of error. For the first error, the transaction does not abend. For a second or subsequent SQL error, the transaction abends with abend code AD2S, AD2T, or AD2U. The transaction must be terminated and reinitialized before it is allowed to issue another SQL request.
  - **N906D** A transaction dump is to be taken and the DSNCSQL RMI associated with the transaction is *not* to be disabled. The transaction receives a -906 SQLCODE if another SQL is issued, unless the transaction issues SYNCPOINT ROLLBACK. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend. The transaction dump records an abend of AD2S, AD2T or AD2U.
  - **N906** The DSNCSQL RMI associated with the transaction is *not* to be disabled. The transaction receives a -906 SQLCODE if another SQL request is issued, unless the transaction issues a SYNCPOINT ROLLBACK. SYNCPOINT without the ROLLBACK option results in an ASP3 or ASP7 abend.

DB2CONN – Pool Thread Parameters

- **ACCOUNTREC({NONE|TASK|TXID|UOW})** Specifies the minimum amount of DB2 accounting required for transactions using pool threads. The specified minimum may be exceeded as described in the following options.
  - **NONE** No accounting records are required for transactions using pool threads. DB2 produces at least one accounting record for each thread when the thread is terminated. Authorization changes additionally cause accounting records to be produced.
  - **TASK** The CICS DB2 attachment facility causes a minimum of one accounting record for each CICS task to be produced. A transaction containing multiple UOWs (assuming the thread is released at syncpoint) might use a different thread for each of its UOWs. The result might be an accounting record produced for each UOW.
  - **TXID** The CICS DB2 attachment facility causes an accounting record to be produced when the transid using the thread changes. A transaction containing multiple units of work (UOWs) might use a different thread for each UOW (assuming the thread is released at syncpoint). An accounting record might be produced per UOW
  - **UOW** The CICS DB2 attachment facility causes an accounting record to be produced for each UOW, assuming that the thread is released at the end of the UOW.

- **DROLLBACK({YES|NO})** Specifies whether or not the CICS DB2 attachment facility should initiate a SYNCPOINT ROLLBACK if a transaction is selected as the victim of a deadlock resolution.
  - **YES** The attachment facility issues a syncpoint rollback before returning control to the application. An SQL return code of -911 is returned to the program.
  - **NO** The attachment facility does not initiate a rollback for a transaction. An SQL return code of -913 is returned to the application.

- **PLAN(*plan*)** Specifies the name of the plan to be used for all pool threads. If PLAN is specified, PLANEXITNAME may not be specified.

- **PLANEXITNAME({DSNCUEXT|*exit*})** Specifies the name of the dynamic plan exit to be used for pool threads. If you change the PLAN and PLANEXITNAME while there are active transactions for the pool, the next time the transaction releases the thread the plan/exit will be determined using the new rules.
  - Note: DSNCUEXT is defined as Quasirent. For Threadsafe programs use DFHD2PXT

DB2CONN – Pool Thread Parameters

- **PRIORITY({HIGH|EQUAL|LOW})** Specifies the priority of the pool thread TCBs relative to the CICS main TCB (QR TCB). The thread TCBs are CICS open L8 TCBs.
  - **HIGH** Thread TCBs have a higher priority than the CICS QR TCB.
  - **EQUAL** Thread TCBs have equal priority with the CICS QR TCB.
  - **LOW** Thread TCBs have a lower priority than the CICS QR TCB.

- **THREADLIMIT({3|*value*})** Specifies the current maximum number of pool threads that the CICS DB2 attachment facility allows to be active before requests are made to wait or are rejected (subject to the THREADWAIT attribute). The default threadlimit (3) is also the minimum you can specify. The maximum value must not be greater than the value specified for TCBLIMIT.

- **THREADWAIT({YES|NO})** Specifies whether or not transactions should wait for a pool thread, or be abended if the number of active pool threads reaches the thread limit. The CICS DB2 attachment issues a unique abend code AD3T, message DFHDB2011, when THREADWAIT=NO is coded and the number of pool threads is exceeded.
  - **YES** If all threads are busy, a transaction must wait until one becomes available. A transaction can wait as long as CICS allows it to wait, generally until a thread becomes available.
  - **NO** If all threads are busy, the transaction is terminated with abend code AD2T or AD3T.

DB2CONN – Command Thread Parameters

- **COMTHREADLIMIT({*1*|value})** The number specifies the current maximum number of command threads the CICS DB2 attachment facility allows active before requests overflow to the pool

DB2ENTRY Parameters

- **TRANSID(*transaction*)** Specifies the transaction id associated with the entry. Only one transaction can be specified here. However, the use of one or more wildcard characters in the TRANSID allows a group of transactions to be represented. Additional transactions can be defined for this entry by defining a DB2 transaction that refers to this DB2 entry. Transid is optional on a DB2 entry. All transactions can be associated with a DB2 entry means of DB2 transactions instead. However, if only one transaction is associated with a DB2 entry it is easier to specify it on the DB2 entry.

- **PROTECTNUM({0|*value*})** Specifies the maximum number of protected threads allowed for this DB2 entry definition. A thread, when it is released by a transaction and there is no other work queued, can be protected, meaning that it is not terminated immediately. A protected thread is terminated after only two complete purge cycles if it has not been reused in the meantime.

- **THREADLIMIT({0|*value*})** Specifies the maximum number of threads for this DB2 entry definition that the CICS DB2 attachment allows active before requests are made to wait, are abended, or diverted to the pool.

- **THREADWAIT({POOL|YES|NO})** Specifies whether or not transactions should wait for a DB2ENTRY thread, be abended, or overflow to the pool should the number of active DB2ENTRY threads reach the THREADLimit number.
  - **POOL** If all threads are busy, the transaction is diverted to use the pool of threads. If the pool is also busy, and NO has been specified for the THREADWAIT attribute on the DB2 connection definition, the transaction is terminated with abend code AD3T.
  - **NO** If all threads are busy, a transaction is terminated with an abend code AD2P.
  - **YES** If all threads are busy, a transaction waits until one becomes available.

DB2TRAN Parameters

- **ENTRY(*db2entry*)** Specifies the name of the DB2 entry definition to which this DB2 transaction definition refers. It is the DB2 entry definition with which this additional transaction should be associated.

- **TRANSID(*transaction*)** Specifies the transaction id to be associated with the entry. If the TRANSID is not specified it defaults to the first four characters of the DB2 transaction definition name. The transaction id can include wildcard characters

Documentation Needed and Time Considerations

Documentation Needed

- z/OS Console dump of both CICS and DB2 taken when the DB2 wait is occurring. Enter the following z/OS command followed by the reply to capture the dump:

```
DUMP COMM=(dumpname)
```

```
R yy, JOBNAME=(*MASTER*,cicsjob,XCFAS,ssnmIRLM,ssnmMSTR,ssnmDBM1),
SDATA=(RGN,CSA,SQA,LPA,LSQA,SWA,PSA,ALLNUC,XESDATA,TRT,GRSQ,SUM),END
```

- *Where yy* is the reply number of the message*, ssnm* is the subsystem name of the DB2 members, and cicsjob is the CICS jobname

Documentation Needed

- CICS Internal Trace
  - The trace should be set to at least 10240K
  - Level 1-2 tracing for FC for CICS releases prior to 3.2
  - Level 1-2 tracing for RI for CICS TS V3.2 and later releases
  - Level 1 tracing for all other CICS components.

You’ve got a dump of a hang. What first?

- **Make sure CICS is healthy as a job**
  - Don’t focus on task hangs if CICS itself is not healthy
  - Find out what time the dump was taken
  - Compare to CICS internal time stamps
  - If CICS is healthy those times will be close together

What time is it?

- Most timestamps are in local time
  - CICS internal trace
  - Console log
  - CICS messages
  - Kernel formatter
- Some timestamps are in GMT
  - CICS dispatcher summary
  - Units of Work (UOW)
  - Dump Incident Token time
- You need to be able to convert from GMT to local
  - Can be tricky if Leap Seconds are being used

Convert GMT time to Local

- Figure out the difference between GMT and Local
  - Issue LTOD 0 from Option 6 of IPCS
  - This will provide the following output:

```
09/17/2042 23:53:47.370496 GMT
09/17/2042 17:53:26.370496 LOCAL
```

- In this case Local time is 6 hours and 21 seconds behind GMT
  - This 21 second difference is due to Leap Seconds
  - Leap Seconds are shop dependent

What time was the dump taken?

- **IP ST SYS**

```
SYSTEM STATUS:
    Sysplex name: PLEX1
  TIME OF DAY CLOCK: BFC60DEA 5FA34DA4  11/28/2011 15:48:46.296628 local
  TIME OF DAY CLOCK: BFC65E61 BD234DA4  11/28/2011 21:48:46.296628 GMT
  Program Producing Dump: SVCDUMP
  Program Requesting Dump: DFHKETCB


  Incident token:  11/28/2011 21:49:05.941810 GMT
```

- **Use Incident token time**
  - **21:49:05.941810 – 6:00:21 = 15:48:44.941810 Local Time**

What time does CICS think it is?

- **VERBX DFHPD630 ‘CSA=2’**

```
CSA 0004EF98 Common System Area
0000   00000248 0004B020 3296FD00 800B9A34  968A4EA4 0098AD30 00000000 7F787000 0008DCB8
0020   968A9288 968A943C 178A1C30 17A8CEA0  0004E770 000BDC60 00000000 03EC0900 000cBBDC
0040   00051020 00057680 0010400C 164E4080  1548449F 00000000 80CBEC30 00000000 7FF733E0
```

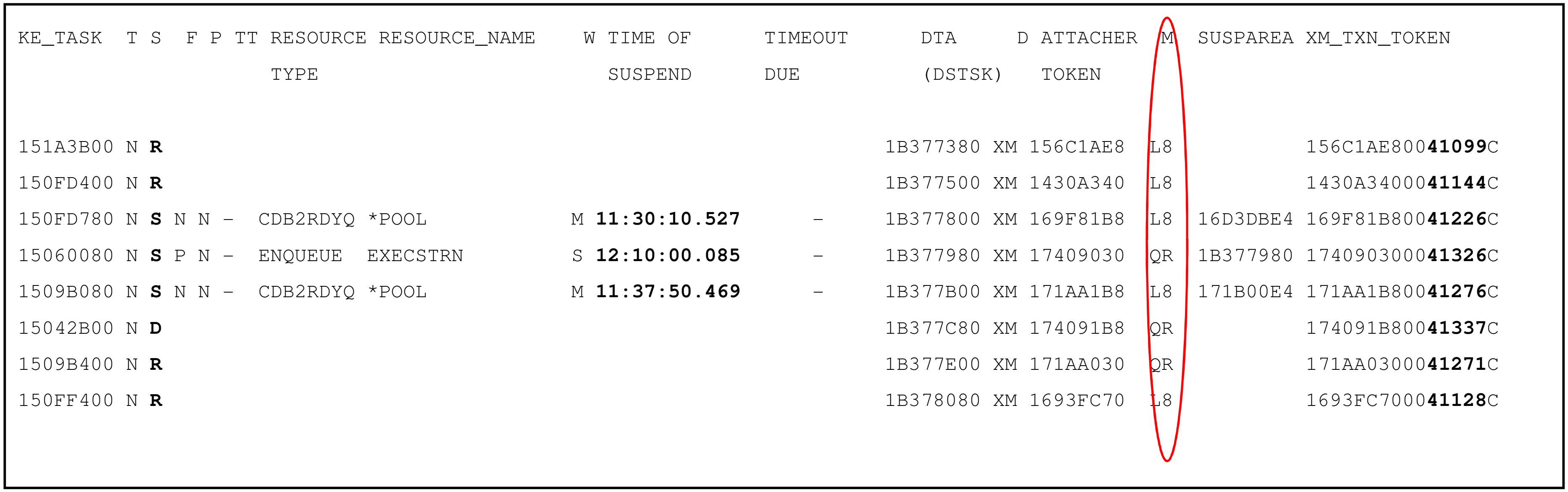
- **CSA +X’50’ is Local Time in packed decimal field of format HHMMSST.   This one is 15:48:44.9**
  - **Matches local time calculated on previous page**
  - **Might have thought CICS was stalled for 21 seconds if Leap Seconds were not accounted for**
  - **Updated every time a task is dispatched on the QR TCB**

If CSA time and Dump time are minutes apart

- CICS is probably not healthy as a job
  - Determine why CICS is not receiving resources needed to run
- Otherwise, proceed to the dispatcher summary to check out why the tasks are hung

CICS and DB2 VERBEXIT Displays

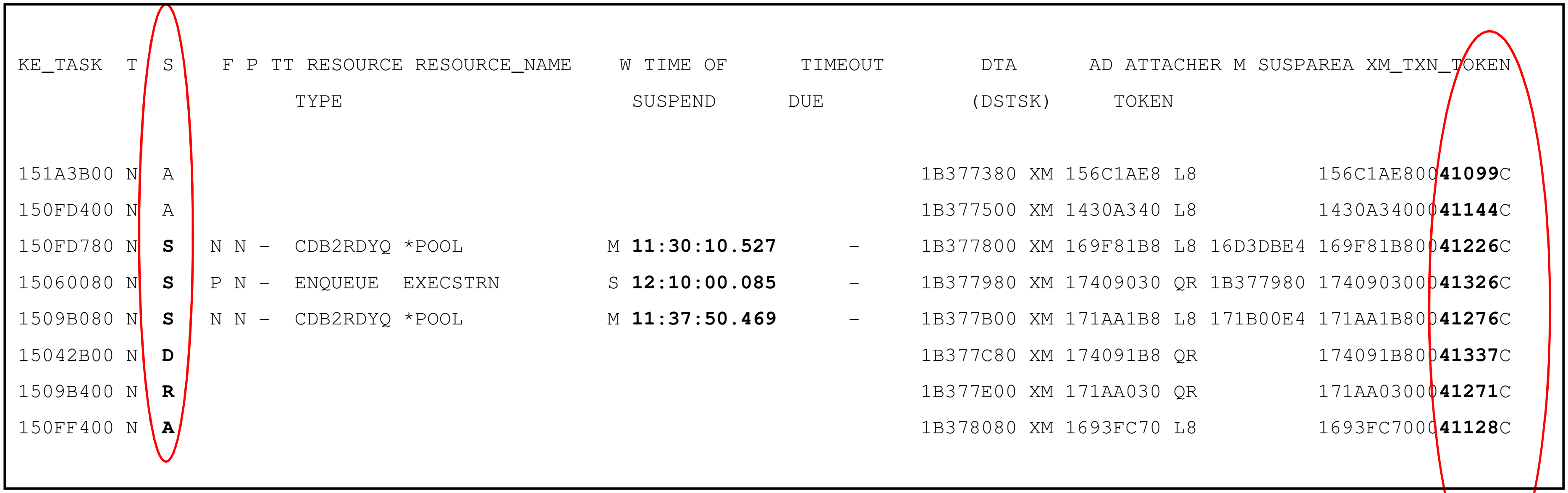
DS=1   The Dispatcher Summary



```
KE_TASK  T S  F P TT RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT      DTA     D ATTACHER  M  SUSPAREA XM_TXN_TOKEN
                     TYPE                        SUSPEND      DUE          (DSTSK)    TOKEN

151A3B00 N R                                                            1B377380 XM 156C1AE8  L8           156C1AE80041099C
150FD400 N R                                                            1B377500 XM 1430A340  L8           1430A3400041144C
150FD780 N S N N -  CDB2RDYQ *POOL            M 11:30:10.527      -     1B377800 XM 169F81B8  L8  16D3DBE4 169F81B80041226C
15060080 N S P N -  ENQUEUE  EXECSTRN         S 12:10:00.085      -     1B377980 XM 17409030  QR  1B377980 174090300041326C
1509B080 N S N N -  CDB2RDYQ *POOL            M 11:37:50.469      -     1B377B00 XM 171AA1B8  L8  171B00E4 171AA1B80041276C
15042B00 N D                                                            1B377C80 XM 174091B8  QR           174091B80041337C
1509B400 N R                                                            1B377E00 XM 171AA030  QR           171AA0300041271C
150FF400 N R                                                            1B378080 XM 1693FC70  L8           1693FC700041128C
```

Task Mode

VERBX DFHPD630 ‘DS=1’   The Dispatcher Summary



```
KE_TASK   T  S    F  P  TT RESOURCE RESOURCE_NAME    W  TIME OF      TIMEOUT        DTA       AD ATTACHER M SUSPAREA XM_TXN_TOKEN
                           TYPE                        SUSPEND       DUE           (DSTSK)        TOKEN

151A3B00 N  A                                                                  1B377380 XM 156C1AE8 L8          156C1AE80041099C
150FD400 N  A                                                                  1B377500 XM 1430A340 L8          1430A340004114 4C
150FD780 N  S   N N -  CDB2RDYQ *POOL          M 11:30:10.527      -           1B377800 XM 169F81B8 L8 16D3DBE4 169F81B80041226C
15060080 N  S   P N -  ENQUEUE  EXECSTRN       S 12:10:00.085      -           1B377980 XM 17409030 QR 1B377980 1740903000041326C
1509B080 N  S   N N -  CDB2RDYQ *POOL          M 11:37:50.469      -           1B377B00 XM 171AA1B8 L8 171B00E4 171AA1B80041276C
15042B00 N  D                                                                  1B377C80 XM 174091B8 QR          174091B80041337C
1509B400 N  R                                                                  1B377E00 XM 171AA030 QR          171AA03000041271C
150FF400 N  A                                                                  1B378080 XM 1693FC70 L8          1693FC70004 1128C
```

Task State

Task Number

R=Running A=Running Abterm Yes J=Running in JVM

S=Suspended

D=Dispatchable

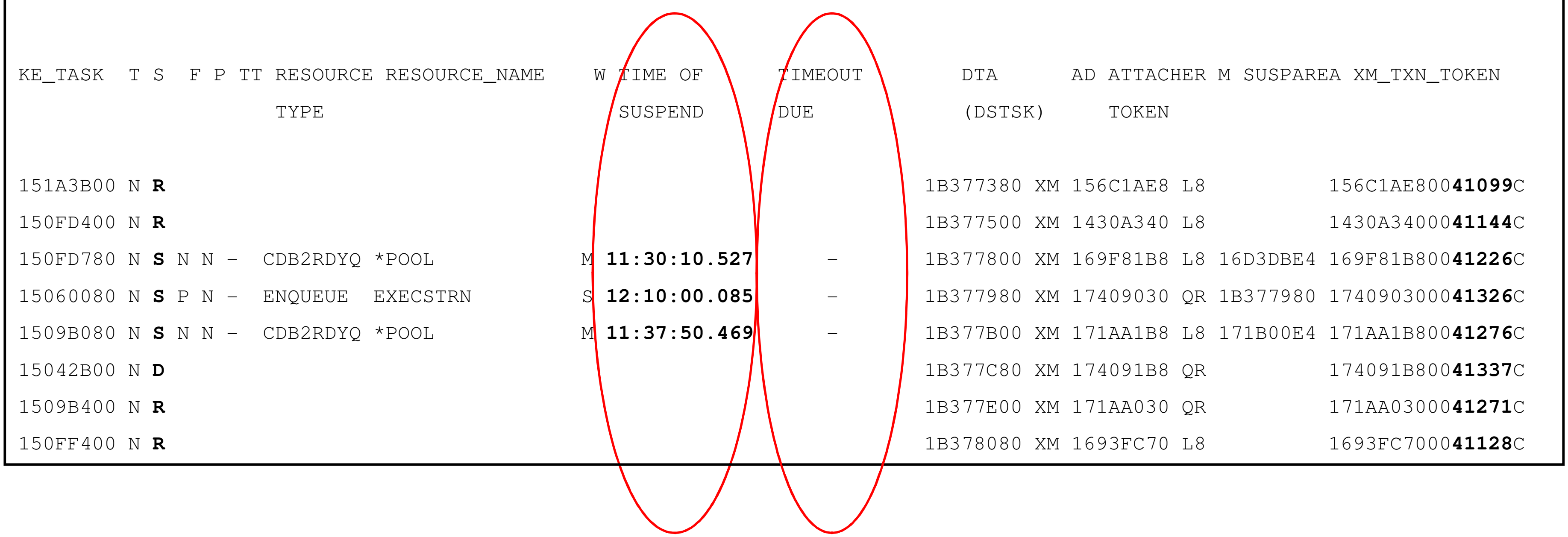
What does it mean to be Running

- The CICS Dispatcher has given control to a task and that task has not yet done anything to give control back to the CICS Dispatcher
  - A Running task might be doing non-yielding EXEC CICS commands
  - It might be in a tight loop in application code
  - It might be churning away in DB2 code processing a complicated SQL call
  - The TCB the task is running on might be in a wait for an IRLM lock or something similar

What does it mean to be Dispatchable

- A task is waiting for the CICS Dispatcher to give it control on the required TCB mode
  - Whatever the task was waiting on has completed but the task hasn’t yet been given control again
  - This is where Wait for Redispatch time gets accumulated
- On single-TCB modes (like QR and RO and CO) other tasks might be dispatched (running) on the TCB
  - Only one task runs at a time on these TCBs

DS=1 The Dispatcher Summary



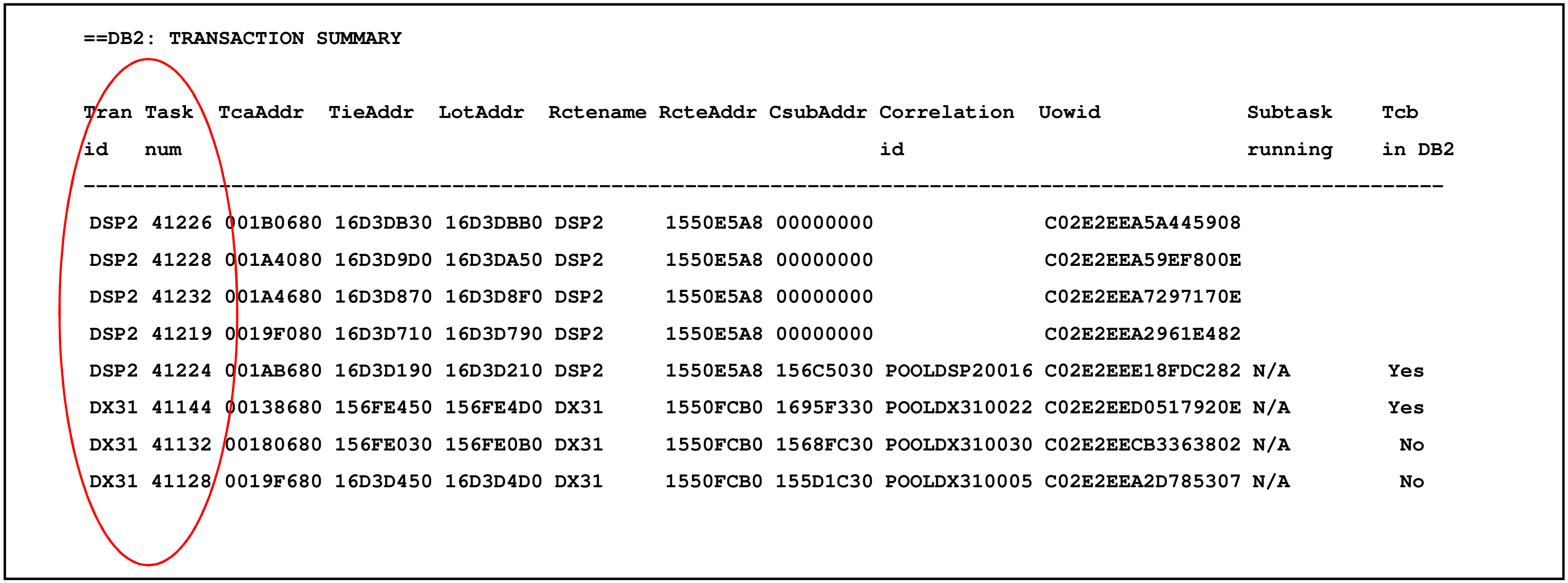
```
KE_TASK  T S  F P TT RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT        DTA      AD ATTACHER M SUSPAREA XM_TXN_TOKEN
                     TYPE                        SUSPEND      DUE            (DSTSK)     TOKEN

151A3B00 N R                                                              1B377380 XM 156C1AE8 L8          156C1AE80041099C
150FD400 N R                                                              1B377500 XM 1430A340 L8          1430A3400041144C
150FD780 N S N N - CDB2RDYQ *POOL            M 11:30:10.527      -       1B377800 XM 169F81B8 L8 16D3DBE4 169F81B80041226C
15060080 N S P N - ENQUEUE   EXECSTRN         S 12:10:00.085      -       1B377980 XM 17409030 QR 1B377980 174090300041326C
1509B080 N S N N - CDB2RDYQ *POOL            M 11:37:50.469      -       1B377B00 XM 171AA1B8 L8 171B00E4 171AA1B80041276C
15042B00 N D                                                              1B377C80 XM 174091B8 QR          174091B80041337C
1509B400 N R                                                              1B377E00 XM 171AA030 QR          171AA0300041271C
150FF400 N R                                                              1B378080 XM 1693FC70 L8          1693FC700041128C
```

Time of Suspend

Timeout Due

Convert from GMT to local

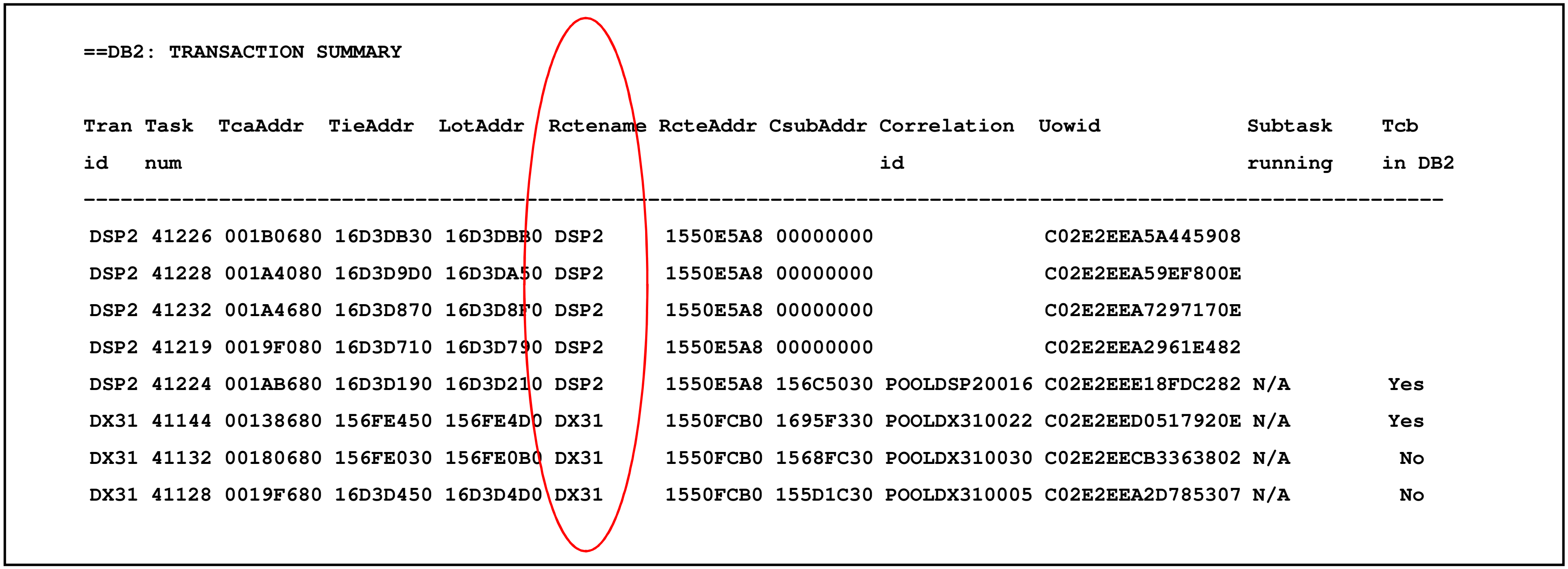
DFHPD630 ‘DB2=1’ The DB2 Transaction summary



```
==DB2: TRANSACTION SUMMARY

Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid            Subtask    Tcb
id   num                                                         id                            running    in DB2
---------------------------------------------------------------------------------------------------------------
DSP2 41226 001B0680 16D3DB30 16D3DBB0 DSP2     1550E5A8 00000000              C02E2EEA5A445908
DSP2 41228 001A4080 16D3D9D0 16D3DA50 DSP2     1550E5A8 00000000              C02E2EEA59EF800E
DSP2 41232 001A4680 16D3D870 16D3D8F0 DSP2     1550E5A8 00000000              C02E2EEA7297170E
DSP2 41219 0019F080 16D3D710 16D3D790 DSP2     1550E5A8 00000000              C02E2EEA2961E482
DSP2 41224 001AB680 16D3D190 16D3D210 DSP2     1550E5A8 156C5030 POOLDSP20016 C02E2EEE18FDC282 N/A        Yes
DX31 41144 00138680 156FE450 156FE4D0 DX31     1550FCB0 1695F330 POOLDX310022 C02E2EED0517920E N/A        Yes
DX31 41132 00180680 156FE030 156FE0B0 DX31     1550FCB0 1568FC30 POOLDX310030 C02E2EECB3363802 N/A         No
DX31 41128 0019F680 16D3D450 16D3D4D0 DX31     1550FCB0 155D1C30 POOLDX310005 C02E2EEA2D785307 N/A         No
```

- Each task in this summary owns an Open TCB.
- The OPEN TCB is owned by the task until the task ends.

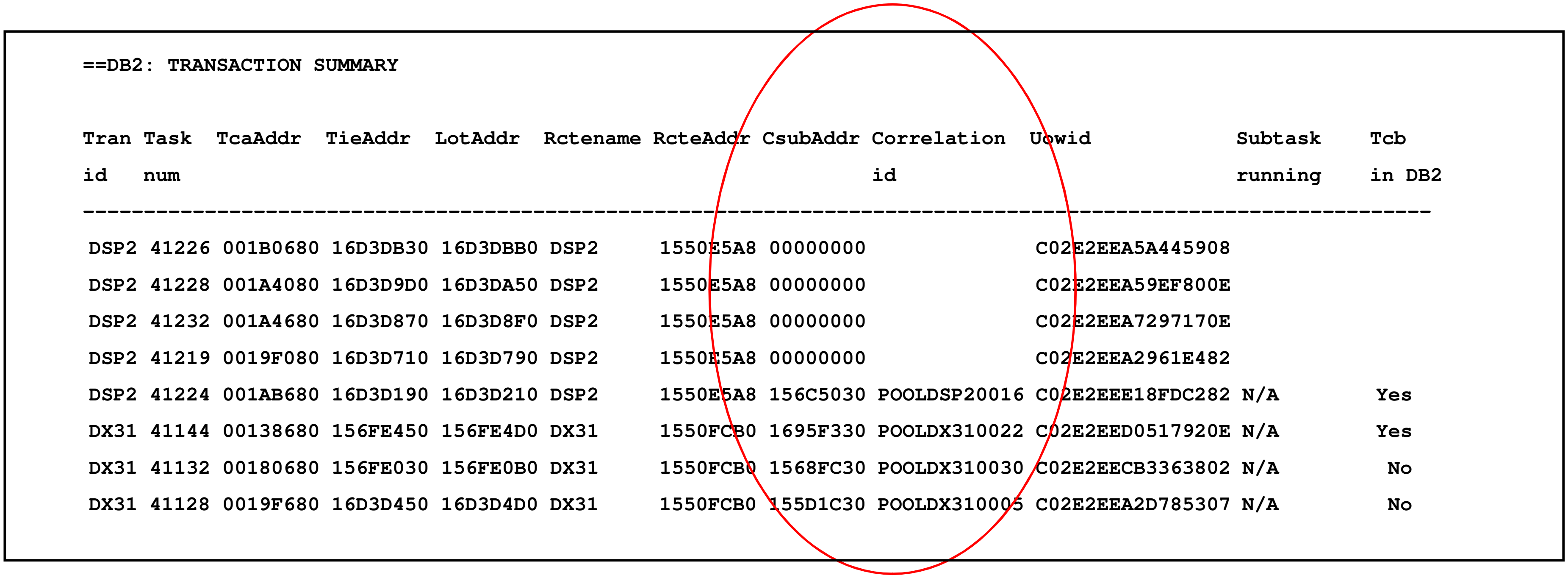
DB2=1 The DB2 Transaction summary



```
==DB2: TRANSACTION SUMMARY

Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid             Subtask    Tcb
id   num                                                         id                             running    in DB2
----------------------------------------------------------------------------------------------------------------------
 DSP2 41226 001B0680 16D3DB30 16D3DBB0 DSP2     1550E5A8 00000000              C02E2EEA5A445908
 DSP2 41228 001A4080 16D3D9D0 16D3DA50 DSP2     1550E5A8 00000000              C02E2EEA59EF800E
 DSP2 41232 001A4680 16D3D870 16D3D8F0 DSP2     1550E5A8 00000000              C02E2EEA7297170E
 DSP2 41219 0019F080 16D3D710 16D3D790 DSP2     1550E5A8 00000000              C02E2EEA2961E482
 DSP2 41224 001AB680 16D3D190 16D3D210 DSP2     1550E5A8 156C5030 POOLDSP20016 C02E2EEE18FDC282 N/A        Yes
 DX31 41144 00138680 156FE450 156FE4D0 DX31     1550FCB0 1695F330 POOLDX310022 C02E2EED0517920E N/A        Yes
 DX31 41132 00180680 156FE030 156FE0B0 DX31     1550FCB0 1568FC30 POOLDX310030 C02E2EECB3363802 N/A         No
 DX31 41128 0019F680 16D3D450 16D3D4D0 DX31     1550FCB0 155D1C30 POOLDX310005 C02E2EEA2D785307 N/A         No
```

- **RCTENAME is the DB2ENTRY that the transaction is using**
- **This is the DB2ENTRY whose THREADWAIT and THREADLIMIT parameters apply**

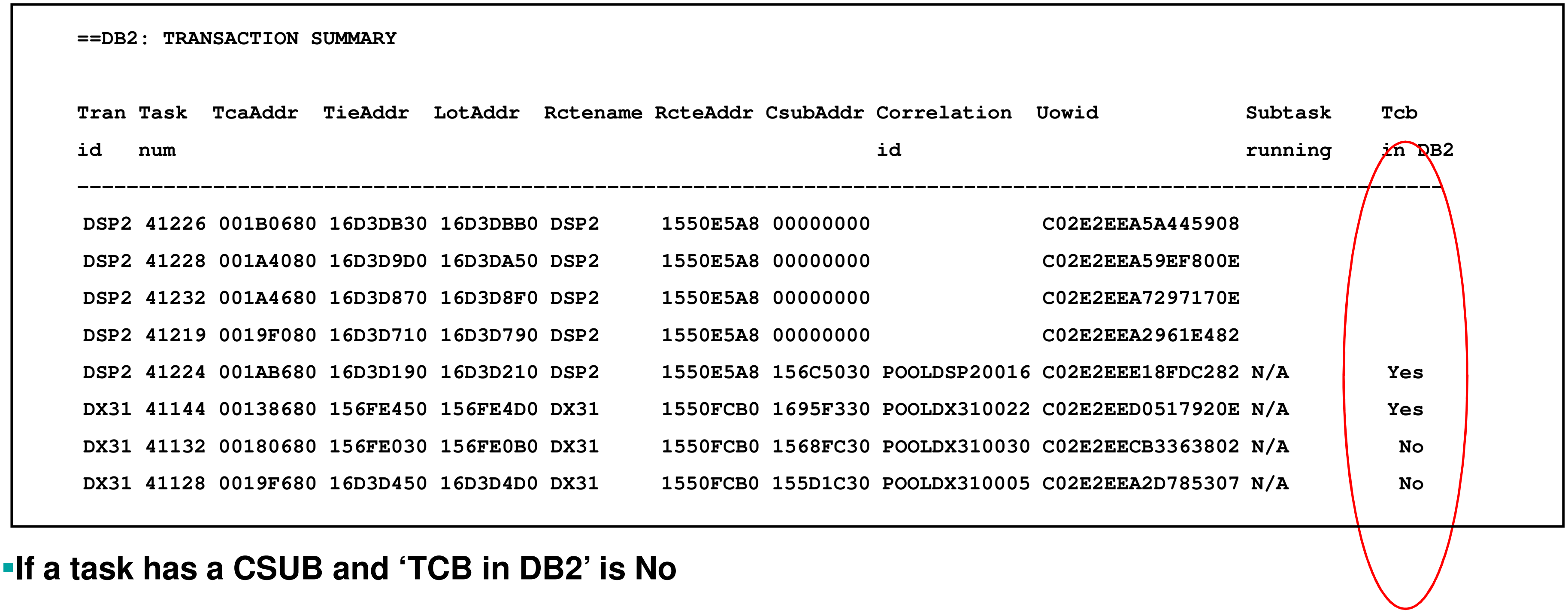
DB2=1 The DB2 Transaction summary



```
==DB2: TRANSACTION SUMMARY

Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid            Subtask    Tcb
id   num                                                         id                             running    in DB2
---------------------------------------------------------------------------------------------------------------
 DSP2 41226 001B0680 16D3DB30 16D3DBB0 DSP2     1550E5A8 00000000              C02E2EEA5A445908
 DSP2 41228 001A4080 16D3D9D0 16D3DA50 DSP2     1550E5A8 00000000              C02E2EEA59EF800E
 DSP2 41232 001A4680 16D3D870 16D3D8F0 DSP2     1550E5A8 00000000              C02E2EEA7297170E
 DSP2 41219 0019F080 16D3D710 16D3D790 DSP2     1550E5A8 00000000              C02E2EEA2961E482
 DSP2 41224 001AB680 16D3D190 16D3D210 DSP2     1550E5A8 156C5030 POOLDSP20016 C02E2EEE18FDC282 N/A        Yes
 DX31 41144 00138680 156FE450 156FE4D0 DX31     1550FCB0 1695F330 POOLDX310022 C02E2EED0517920E N/A        Yes
 DX31 41132 00180680 156FE030 156FE0B0 DX31     1550FCB0 1568FC30 POOLDX310030 C02E2EECB3363802 N/A         No
 DX31 41128 0019F680 16D3D450 16D3D4D0 DX31     1550FCB0 155D1C30 POOLDX310005 C02E2EEA2D785307 N/A         No
```

- The CICS DB2 Connection Block (CSUB) is the CICS control block that represents a thread. If a task has a CSUB and a Correlation id, it has a thread
- The tasks that do not have a CSUB are waiting for a thread

DB2=1 The DB2 Transaction summary



```
==DB2: TRANSACTION SUMMARY

Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid            Subtask    Tcb
id   num                                                         id                            running    in DB2
---------------------------------------------------------------------------------------------------------------
DSP2 41226 001B0680 16D3DB30 16D3DBB0 DSP2     1550E5A8 00000000              C02E2EEA5A445908
DSP2 41228 001A4080 16D3D9D0 16D3DA50 DSP2     1550E5A8 00000000              C02E2EEA59EF800E
DSP2 41232 001A4680 16D3D870 16D3D8F0 DSP2     1550E5A8 00000000              C02E2EEA7297170E
DSP2 41219 0019F080 16D3D710 16D3D790 DSP2     1550E5A8 00000000              C02E2EEA2961E482
DSP2 41224 001AB680 16D3D190 16D3D210 DSP2     1550E5A8 156C5030 POOLDSP20016 C02E2EEE18FDC282 N/A         Yes
DX31 41144 00138680 156FE450 156FE4D0 DX31     1550FCB0 1695F330 POOLDX310022 C02E2EED0517920E N/A         Yes
DX31 41132 00180680 156FE030 156FE0B0 DX31     1550FCB0 1568FC30 POOLDX310030 C02E2EECB3363802 N/A          No
DX31 41128 0019F680 16D3D450 16D3D4D0 DX31     1550FCB0 155D1C30 POOLDX310005 C02E2EEA2D785307 N/A          No
```

- **If a task has a CSUB and ‘TCB in DB2’ is No**
  - **The task has made an SQL call in the past but is currently doing something else.**
  - **It owns a thread while it is doing something else.**
  - **You have to go back to the Dispatcher Summary to see what this task is currently doing.**
- **If a task has a CSUB and ‘TCB in DB2’ is Yes**
  - **The task is in DB2. The Dispatcher state of this task is ‘Running’**

What it looks like when control is not in DB2

```
==DB2: TRANSACTION SUMMARY
Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid              Subtask   Tcb
 id   num                                                          id                             running   in DB2
 ---------------------------------------------------------------------------------------------------------------
  PW69 16526 1EC91680 1DDB2450 1DDB24D0 PWXX     17DEF5A8 17877330 ENTRPW690017 C0366382170A6D81 N/A        No
```

```
  ===DS:  Dispatcher Summary
  KE_TASK  T S F P TT RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT      DTA      AD ATTACHER M SUSPAREA XM_TXN_TOKEN
  |                   TYPE                        SUSPEND      DUE          (DSTSK)     TOKEN
  14155080 N S P N -  FCXCWAIT FILEA            C 00:19:10.634  -           5E2C7C80 XM 1B174AE8 QR 17CFEDBD 1B174AE80016526C
```

```
    ==PG: PTA SUMMARY FOR TRAN NUM : 16526   PTA ADDRESS : 1B29EBB8
    LOG-LVL : 4         SYS-LVL : 0        TASK-LLE : 00000000  PLCB : 1D3772F0
    =PG: TASK PLCB SUMMARY
    PLCB-ADD PROGRAM   LOG-LVL LOAD     ENTRY    LENGTH CA-CURR  CLEN INVK-PRG STG EXIT-NME ENV  PPTE-ADD
    1D3772F0 PWOU69          4 1962BF50 9962BF70 00BC68 1AE256C0 08EF PWOU6G                EXEC 17DC83A0
    1415AB90 PWOU6G          3 1961CCA0 9961CCC0 00F2B0 27EB1000 7FFF DFHWBBLI              EXEC 17DC8190
    14158B60 DFHWBBLI        2 184F17C0 984F17E8 005898 1AE048C8 0090 DFHWBA                EXEC 173E9920
    14156C58 DFHWBA          1 184EDE80 984EDEA8 003938 00000000 0000 CICS                  EXEC 173E9710
```

What it looks like when control is in DB2

```
==DB2: TRANSACTION SUMMARY
Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid            Subtask    Tcb
id   num                                                         id                            running    in DB2
----------------------------------------------------------------------------------------------------------------
DSP2 41224 001AB680 16D3D190 16D3D210 DSP2     1550E5A8 156C5030 POOLDSP20016 C02E2EEE18FDC282 N/A        Yes
```

```
===DS: Dispatcher Summary
KE_TASK  T S F P TT RESOURCE RESOURCE_NAME    W TIME OF      TIMEOUT      DTA      AD ATTACHER M SUSPAREA XM_TXN_TOKEN
                    TYPE                        SUSPEND      DUE          (DSTSK)     TOKEN

151C1080 N A                                                              1B376980 XM 156C1960 L8          156C19600041224C
```

```
==PG: PTA SUMMARY FOR TRAN NUM : 41124,  PTA ADDRESS : 1B29ABB8
LOG-LVL : 4         SYS-LVL : 0         TASK-LLE : 00000000  PLCB : 1D3772F0 =PG: TASK PLCB SUMMARY
PLCB-ADD  PROGRAM    LOG-LVL  LOAD      ENTRY     LENGTH  CA-CURR   CLEN INVK-PRG  STG EXIT-NME  ENV   PPTE-ADD
151C7DD8  DFHD2EX1          4 00000000  00000000  000000  00000000  0000  PRG3CICS                TRUE  1528D1E8
151C6B28  PRG3CICS          3 1673E650  9673E6E0  017AD0  16AD6760  0433  PRG2CICS                EXEC  15536D98
151C4B28  PRG2CICS          2 16736360  967363F0  0082F0  164A9608  012C  PRG1CICS                EXEC  15530F50
151C2C58  PRG1CICS          1 16200000  96253B20  08E018  00000000  0000  CICS                    EXEC  1553EB88
```

DB2 perspective when they are in control

- Issue VERBX DFHPDxxx ‘DB2=1’
- Near the top, make a note of the CICS ASID.

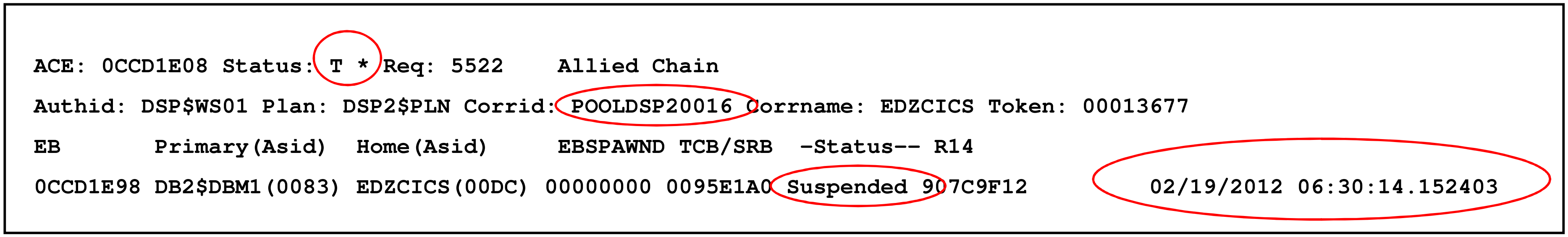
```
-- DFHPD0121I FORMATTING CONTROL BLOCKS FOR JOB EDZCICS
ADDRESS SPACE ASID NUMBER (HEX) = 00DC
```

- A little further down, note the DB2 Subsystem ID.

```
==DB2: GLOBAL STATE SUMMARY


   Db2conn name:                     EDZZ
   Connection status:                Connected
   In standby mode:                  No
   DB2 id:                           DB2E
```

DB2 perspective when they are in control

- Issue VERBX DSNWDMP ‘subsys=xxxx,ds=3’ where xxxx is the DB2 Subsystem ID
- Issue a Find for ‘ASID(xxxx)’ where xxxx is the CICS ASID. There will be a list of Threads like this one:

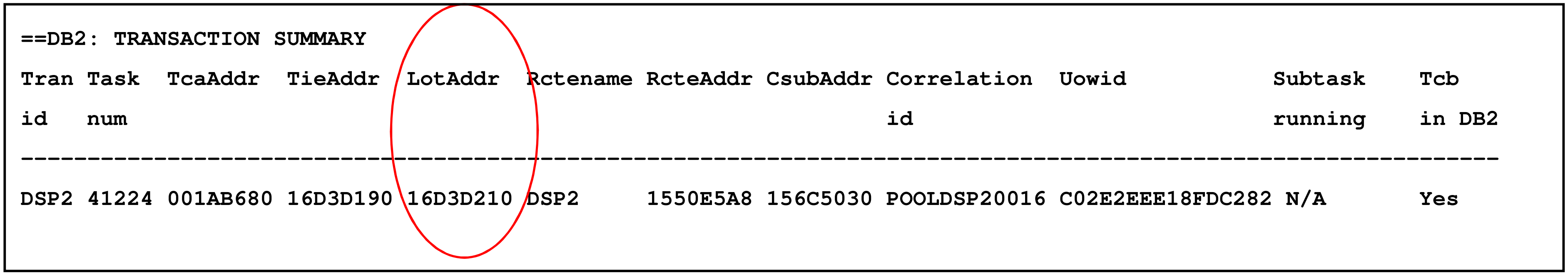


```
ACE: 0CCD1E08 Status: T * Req: 5522    Allied Chain
Authid: DSP$WS01 Plan: DSP2$PLN Corrid: POOLDSP20016 Corrname: EDZCICS Token: 00013677
EB       Primary(Asid)  Home(Asid)     EBSPAWND TCB/SRB  -Status-- R14
0CCD1E98 DB2$DBM1(0083) EDZCICS(00DC)  00000000 0095E1A0 Suspended 907C9F12        02/19/2012 06:30:14.152403
```

- **T * means control is in DB2**
- **Corrid matches Correlation ID from the CICS DB2 summary**
- **Thread was suspended at 6:30:14 local time**

SQL statement information when control is in DB2

- Obtain the LOT Address from the DB2 Transaction Summary



```
==DB2: TRANSACTION SUMMARY
Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid           Subtask    Tcb
id   num                                                         id                           running    in DB2
---------------------------------------------------------------------------------------------------------------
DSP2 41224 001AB680 16D3D190 16D3D210 DSP2     1550E5A8 156C5030 POOLDSP20016 C02E2EEE18FDC282 N/A       Yes
```

- Issue IP L xxxxxxxx+30?? L(X’28’)
  - Where xxxxxxxx is the LOT address
  - Length is X’40’ in DB2 V8 and later
- This will display the Relational Data system Input parameter list (RDI) for the SQL call

Format of RDI – Pre DB2 V8

```
LIST 164C8A40. ASID(X'00DC') LENGTH(X`28') AREA
164C8A40. 00284000 001EC5C4 E9E9D7D9 D6C717F7  AD120202 D542000C 164C3870 164C8A68 |.. ...EDZZPROG.7....N....<...|
164C8A60. 00000000 1ADD 00E8                                                         |.......Y                    |
```

- +X’6’ Program Name (8 bytes)
  - EDZZPROG
- +X’24’ DB2 Precompiler Statement Number (2 bytes)
  - X’1ADD’ = 6877 Decimal
- +X’26’ SQL Statement Type (2 bytes)
  - X’00E8’ = Insert

Format of RDI – DB2 V8 and later

```
LIST 2DF30A00. ASID(X'0316') LENGTH(X'40') AREA
2DF30A00. 00400400 001E4544 5A5A5052 4F4717D7 C62C0BD9 53D60010 2DF30CA0 00000000 |                          . . .|
2DF30A20. 2DF30BDC 04B80004 00000C1D 00000000 00000000 00000000 00000000 00000000 |...&.&....F..R.O...3...  ...3..|
```

- +X’6’ Program Name (8 bytes in ASCII !)
  - 45445A5A50524F47 = EDZZPROG
- +X’26’ SQL Statement Type (2 bytes)
  - 0004 = FETCH
- +X’28’ DB2 Precompiler Statement number (4 bytes)
  - Converted to decimal = 3101

DSNXRDI Partial Listing

```
0003 DC     H'0003',CL24'OPEN
0004 DC     H'0004',CL24'FETCH
0005 DC     H'0005',CL24'CLOSE
00E7 DC     H'0231',CL24'SELECT
00E8 DC     H'0232',CL24'INSERT
00E9 DC     H'0233',CL24'DELETE
00EA DC     H'0234',CL24'UPDATE
00EB DC     H'0235',CL24'INSERT WITHIN SELECT
02EB DC     H'0747',CL24'CONNECT TO
02EC DC     H'0748',CL24'CONNECT RESET
02ED DC     H'0749',CL24'CONNECT
02EE DC     H'0750',CL24'IMPLICIT CONNECT
030D DC     H'0781',CL24'SET CURRENT RULES
030E DC     H'0782',CL24'CALL STATEMENT
030F DC     H'0783',CL24'FETCH ABSOLUTE MRS
```

Problem One - Hang

Problem One - Hang

- Customer called the Support Center for DB2 transactions hung

- LTOD 0 shows no Leap Seconds usage

```
09/17/2042 23:53:47.370496 GMT
09/17/2042 18:53:47.370496 LOCAL
```

- ST SYS shows

```
SYSTEM STATUS:
  Sysplex name: EDZPLEX
  TIME OF DAY CLOCK: C1F903EB 2D3E2304  02/19/2012 10:01:48.186594 local
  TIME OF DAY CLOCK: C1F946F9 507E2304  02/19/2012 15:01:48.186594 GMT
  Incident token: LOCAL    EDZPROD  02/19/2008 15:01:47.569848 GMT
```

- VERBX DFHPD640 ‘CSA=2’ shows

```
 0000  00000198 0004A020 15E8C418 8870FC16  80C3A1E0 80800000 12EE4C10 12EE5170
 0020  1326A030 12EAD400 8870F2AC 8870FAEA  087102AC 12EBEC2C 12EE4C10 1326A030
 0040  00054690 12BA0080 0012038C 12BA0680  1001475F 12ED9800 00000100 00000000
```

VERBX DFHPD640 ‘DS=3’

```
===DS: DISPATCHER DOMAIN - SUMMARY

     DATA FOR TCB POOL CONTROLLED BY MAXOPENTCBS

          MODES IN POOL ARE:   L8

           MAX POOL SIZE  =    15        AT POOL LIMIT  =   YES

           MVS STORAGE CONSTRAINED =     NO

           NUMBER OF TASKS SUSPENDED AWAITING POOL TCBS =     9

           NUMBER OF TASKS SUSPENDED AWAITING OPEN TCBS =     0

           NUMBER OF TASKS SUSPENDED BECAUSE MVS STORAGE

                                         IS CONSTRAINED =     0

           NUMBER OF TCBS IN POOL        CURRENT HIGH WATER

           IN EXISTENCE                     15         15

           ALLOCATED TO TASKS               15         15
```

VERBX DFHPD640 ‘DS=3’

```
02800125 12BFBB00 N S P N -  FCPSWAIT EDZFILE     C 14:39:03.168      -     15D36080 XM 1406E4C8 QR 1333A2C5 1406E4C80005239C
02840221 12BFE080 N S P N -  FCPSWAIT EDZFILE     C 14:38:48.789      -     15D36380 XM 134F4960 QR 1333A2C5 134F49600004982C
02860025 12BFB080 N S P N -  DISPATCH OPENPOOL    S 14:39:08.660      -     15D36500 XM 1406E960 QR 12ACCA10 1406E9600005299C
02880075 12E6B780 N S P N -  FCPSWAIT EDZFILE     C 14:38:47.687      -     15D36680 XM 12B0A960 QR 1333A2C5 12B0A9600005046C
028A0011 12BFD080 N S P N -  DISPATCH OPENPOOL    S 14:38:59.528      -     15D36800 XM 14076C70 QR 12ACC9B0 14076C700005191C
028C0135 12E4EB00 N S N N -  CDB2RDYQ *POOL       M 14:38:53.532      -     15D36980 XM 134F44C8 L8 13EE9504 134F44C80005127C
02900193 12E6B080 N S P N -  FCPSWAIT EDZFILE     C 14:38:48.122      -     15D36C80 XM 12B097D8 QR 1333A2C5 12B097D80005047C
02920007 12E6B400 N S P N -  DISPATCH OPENPOOL    S 14:38:56.168      -     15D36E00 XM 12B0A340 QR 12ACC860 12B0A3400005174C
030000F1 12BFC400 N S P N -  DISPATCH OPENPOOL    S 14:39:01.777      -     15D37080 XM 1406E030 QR 12ACCA40 1406E0300005220C
030201F9 12BFE400 N S P N -  FCPSWAIT EDZFILE     C 14:38:47.748      -     15D37200 XM 12B0A7D8 QR 1333A2C5 12B0A7D80005026C
030A006B 12BFEB00 N S P N -  FCPSWAIT EDZFILE     C 14:38:49.107      -     15D37800 XM 12B0A1B8 QR 1333A2C5 12B0A1B80005061C
030C014B 12E31780 N S P N -  FCPSWAIT EDZFILE     C 14:38:47.193      -     15D37980 XM 14076030 QR 1333A2C5 140760300005030C
030E0053 12BFF080 N S P N -  FCPSWAIT EDZFILE     C 14:38:47.278      -     15D37B00 XM 134F4650 QR 1333A2C5 134F46500004961C
03100141 12BFCB00 N S P N -  DISPATCH OPENPOOL    S 14:38:59.210      -     15D37C80 XM 14076AE8 QR 12ACC890 14076AE80005189C
0380008D 12BFF780 N S P N -  FCPSWAIT EDZFILE     C 14:38:53.214      -     15D6E080 XM 134F4340 QR 1333A2C5 134F43400005123C
038600DD 12B3A780 N S P N -  DISPATCH OPENPOOL    S 14:39:09.293      -     15D6E500 XM 1406EC70 QR 12ACCCB0 1406EC700005306C
0388010D 12BFE780 N S P N -  DISPATCH OPENPOOL    S 14:38:57.746      -     15D6E680 XM 14076650 QR 12ACCC50 140766500005182C
038A0083 12BFD400 N S P N -  DISPATCH OPENPOOL    S 14:38:56.327      -     15D6E800 XM 140761B8 QR 12ACC770 140761B80005177C
038C014D 12BFDB00 N S P N -  FCPSWAIT EDZFILE     C 14:38:52.288      -     15D6E980 XM 14076340 QR 1333A2C5 140763400005100C
0392017D 12E4E400 N S P N -  FCPSWAIT EDZFILE     C 14:38:52.101      -     15D6EE00 XM 12B0A650 QR 1333A2C5 12B0A6500005101C
040001BF 12E4E780 N S N N -  CDB2RDYQ *POOL       M 14:38:53.235      -     15D6F080 XM 134F4DF8 L8 13455244 134F4DF80005125C
04020085 12E6BB00 N S P N -  FCPSWAIT EDZFILE     C 14:38:49.131      -     15D6F200 XM 12B0A030 QR 1333A2C5 12B0A0300005063C
040400EF 12E4E080 N S P N -  FCPSWAIT EDZFILE     C 14:38:52.967      -     15D6F380 XM 134F4AE8 QR 1333A2C5 134F4AE80005121C
040E01A3 12E31400 N S P N -  FCPSWAIT EDZFILE     C 14:38:52.403      -     15D6FB00 XM 140767D8 QR 1333A2C5 140767D80005107C
04120097 12BFB400 N S P N -  DISPATCH OPENPOOL    S 14:39:04.017      -     15D6FE00 XM 1406E7D8 QR 12ACCCE0 1406E7D80005246C
```

Problem One – What we know so far

- Customer’s DB2 tasks were hung
- Dump was taken at 15:01:47.569848 GMT
- CICS was running up until dump time
- DS=3 showed all 15 L8 TCBs used with 9 transactions waiting for L8 TCB
- DS=3 showed the transactions waiting since around 14:38 GMT
- Prevalent DB2 waits were DISPATCH OPENPOOL and CDB2RDYQ

DB2 Waits

- **DISPATCH OPENPOOL**
  - A task waiting on DISPATCH OPENPOOL is waiting for a free OPEN TCB. The limit is set by the MAXOPENTCBS SIT parameter
  - Every tasking making SQL calls needs an OPEN TCB
  - Need to figure out which tasks own all the Open TCBs
- **CDB2RDYQ**
  - A task waiting on CDB2RDYQ is waiting for a DB2 thread. The number of threads is limited by the THREADLIMIT parameter on the DB2ENTRY resource or on the DB2CONN Pool resource
  - Need to figure out which tasks own the threads
- Review DB2 Transaction Summary to find the owners of the OPEN TCBs and of the threads

VERBX DFHPD640 ‘DB2=3’

```
==DB2: GLOBAL STATE SUMMARY
     Db2conn name:                          EDZZ
     Connection status:                     Connected
     In standby mode:                       No
     DB2 id:                                DB21
     DB2 Group id:
     DB2 release:                           0810
     Operating in OpenAPI mode:             Yes
     Service task started:                  Yes
     Master subtask started:                No – not required
     Tcb limit:                             15
     Currently active tcbs:                 13
     Message Queue1:                        CSMT
     Message Queue2:
     Message Queue3:
     Statistics Queue:                      CSSL
     Standbymode:                           Reconnect
     Connecterror:                          Sqlcode
```

VERBX DFHPD640 ‘DB2=3’

```
==DB2: TRANSACTION SUMMARY

 Tran Task  TcaAddr  TieAddr  LotAddr  Rctename RcteAddr CsubAddr Correlation  Uowid            Subtask   Tcb
 id   num                                                         id                            running   in DB2
 ----------------------------------------------------------------------------------------------------------------
 EDZ1 05127 001F5680 13EE9450 13EE94D0 EDZ1     13419CB0 00000000              C1F941DA4EC7FA44
 EDZ1 05125 001F1080 13455190 13455210 EDZ1     1341AA58 00000000              C1F941D9F87F1904
 EDZ1 05123 001F8080 13455710 13455790 EDZ1     1341AA58 13224330 POOLEDZ10002 C1F941D9DE4A8304 N/A        No
 EDZ2 05121 001F2680 13455B30 13455BB0 *POOL    12F313D8 13EBDC30 POOLEDZ20010 C1F941D9C78FB004 N/A        No
 EDZ2 05107 001F4680 13455DF0 13455E70 *POOL    12F313D8 13EFB630 POOLEDZ20013 C1F941D90B69BC04 N/A        No
 EDZ1 05100 001F1680 13EE9030 13EE90B0 EDZ1     1341AA58 13224030 POOLEDZ10001 C1F941D8C5757F44 N/A        No
 EDZ1 05101 001F4080 134559D0 13455A50 EDZ1     1341AA58 13EBD030 POOLEDZ10006 C1F941D8DFD80984 N/A        No
 EDZ3 05063 001F2080 134555B0 13455630 EDZ3     1341B418 13EFB030 POOLEDZ30011 C1F941D5FEDB3A84 N/A        No
 EDZ4 05061 001F8680 13455870 134558F0 *POOL    12F313D8 13224630 POOLEDZ40003 C1F941D5E96043C4 N/A        No
 EDZ5 05047 001F7080 13455450 134554D0 EDZ5     134175A8 13224930 POOLEDZ50004 C1F941D4DB5DD644 N/A        No
 EDZ6 05046 001F5080 134552F0 13455370 *POOL    12F313D8 13EBD630 POOLEDZ60008 C1F941D4B4A7EEC4 N/A        No
 EDZ7 05030 0005E680 13EE95B0 13EE9630 *POOL    12F313D8 13EBD330 POOLEDZ70007 C1F941D4489B9504 N/A        No
 EDZ8 05026 001F7680 13455C90 13455D10 EDZ8     13426A58 13EBD930 POOLEDZ80009 C1F941D3E62C1304 N/A        No
 EDZ1 04982 001F0080 13EE92F0 13EE9370 EDZ1     1341AA58 13224C30 POOLEDZ10005 C1F941D0784673C4 N/A        No
 EDZ9 04961 001F0680 13EE9190 13EE9210 EDZ9     134211C0 13EFB330 POOLEDZ90012 C1F941CEC44B89C4 N/A        No
```

What we know from VERBX DFHPD640 ‘DS=3’

- There were 13 active threads in the summary and a TCBLIMIT of 15
- All of the threads were using POOL threads
  - Even though some transactions have RCTEs (DB2ENTRY) specified they still overflowed to the POOL
    - Customers will do this to get the PLAN name, ACCOUNTREC, and security attributes of a DB2ENTRY but have THREADLIMIT(0) and THREADWAIT(POOL)
  - EDZ2, EDZ4, EDZ6 and EDZ7 went directly to a POOL thread because they do not have a DB2TRAN coded pointing to a DB2ENTRY
- There were 2 tasks owning a TCB but did not have a CSUB
  - This indicates they were not connected to a DB2 Thread
  - If there were tasks waiting why were the two TCBs not connected to a thread?

VERBX DFHPD640 ‘DB2=3’ DB2 Global Block

```
  DFHD2GLB 12F31000 CICS/DB2 GLOBAL BLOCK
     0000   05686EC4 C6C8C4F2 C7D3C240 40404040  D9C3E3D7 D9404040 C3C9C3E2 D7D9D6C4  *..>DFHD2GLB     RCTPR   CICSPROD*
     0020   40404040 C4C2F2D7 F0F8F1F0 8000BB38  00045004 87F29028 00000000 00000000  *    DB2P0810......&.g2..........*
     0040   00000000 12B9E260 C3E2D4E3 00000000  00000000 C3C9C3E2 D7D9C4C2 00000000  *......S-CSMT........CICSPRDB....*
     0060   00000000 C1F940B0 26442544 00000000  00000000 C3E2E2D3 00000000 0000001E  *....A9 .............CSSL........*
     0080   0000000D 0000000D 0000000F 00000000  00000000 0000000D 00000000 00000000  *................................*
     00A0   00000000 00000000 00000000 0000000   00000000 00000000 00000000 00000000  *................................*
     00C0 -     03BF LINES SAME AS ABOVE
     03C0   00000000 00000000 00000000 00000000  00000000 00000000 0000C7D3 C240D7D6  *..........................GLB PO*
     03E0   D6D340E2 C5C3E3D5 5CD7D6D6 D3404040  C1F940AD 52AC3E04 E7C1F0F0 F0F1D7D3  *OL SECTN*POOL   A9 .....XA0001PL*
     0400   00000000 00000000 C3C5D7D3 00000000  00000000 00000000 80208080 80000000  *........CEPL....................*
     0420   C1F940AD 52AC3E04 41F940AD 52AC3E04  0000000D 00000000 0000000D 0000000D  *A9 ......9 .....................*
     0440   00000000 00000000 00000005 00000005  00000004 00000004 00000066 000001E8  *...............................Y*
     0460   0000006A 00000021 00000000 00000000  0000006A 00000002 000003A1 0000000C  *................................*
     0480   00000000 00000000 13224330 00000000  00000000 13455BB0 13EE9790 0000000C  *......................$...p.....*
```

```
(80) UNSIGNED 4 GLB_CURRENT_TCBS Current TCB number                    X’D’  = 13  NOTE: Really Current number of connections
(88) UNSIGNED 4 GLB_TCB_LIMIT Max number of TCBs                       X’F’  = 15
(430) UNSIGNED 4 RCT_THREAD_ Maximum active threads LIMIT              X’D’  = 13
(438) UNSIGNED 4 RCT_CURRENT_ ACTIVE_THREADS No of threads active      X’D’  = 13
(43C) UNSIGNED 4 RCT_THREAD_HWM hwm of active threads                  X’D’  = 13
```

- DB2CONN TCBLIMIT is set to 15
- DC2CONN Pool Thread Limit is set to 13
    - Since all DB2ENTRYs have THREADLIMIT(0) and THREADWAIT(POOL) there can only be 13 Threads maximum
    - 14th and 15th tasks will be put into CDB2RDYQ since they can get an Open TCB but not a DB2 Thread
    - Rest of the tasks will be put into DISPATCH OPENPOOL since they cannot get an Open TCB

Problem One – What we know so far

- Customer’s DB2 tasks were hung
- Dump was taken at 15:01:47.569848 GMT
- CICS was running up until dump time
- DS=3 showed all 15 L8 TCBs used with 9 transactions waiting for L8 TCB
- DS=3 showed the transactions waiting since around 14:38 GMT
- Prevalent DB2 waits were DISPATCH OPENPOOL and CDB2RDYQ
- DB2=3 showed 15 tasks owning an L8 TCB and 13 of the 15 owning a Thread
- DB2 Global Block showed TCBLIMIT of 15 coded and Pool Thread limit of 13
- Every DB2ENTRY had THREADLIMIT(0) and THREADWAIT(POOL)
  - This causes all DB2 calls to use Pool Threads
- All 13 Thread owners were in CICS

VERBX DFHPD640 ‘DS=3’

```
02800125 12BFBB00 N S P N - FCPSWAIT EDZFILE     C 14:39:03.168    -     15D36080 XM 1406E4C8 QR 1333A2C5 1406E4C80005239C
02840221 12BFE080 N S P N - FCPSWAIT EDZFILE     C 14:38:48.789    -     15D36380 XM 134F4960 QR 1333A2C5 134F49600004982C
02860025 12BFB080 N S P N - DISPATCH OPENPOOL    S 14:39:08.660    -     15D36500 XM 1406E960 QR 12ACCA10 1406E9600005299C
02880075 12E6B780 N S P N - FCPSWAIT EDZFILE     C 14:38:47.687    -     15D36680 XM 12B0A960 QR 1333A2C5 12B0A9600005046C
028A0011 12BFD080 N S P N - DISPATCH OPENPOOL    S 14:38:59.528    -     15D36800 XM 14076C70 QR 12ACC9B0 14076C700005191C
028C0135 12E4EB00 N S N N - CDB2RDYQ *POOL       M 14:38:53.532    -     15D36980 XM 134F44C8 L8 13EE9504 134F44C80005127C
02900193 12E6B080 N S P N - FCPSWAIT EDZFILE     C 14:38:48.122    -     15D36C80 XM 12B097D8 QR 1333A2C5 12B097D80005047C
02920007 12E6B400 N S P N - DISPATCH OPENPOOL    S 14:38:56.168    -     15D36E00 XM 12B0A340 QR 12ACC860 12B0A3400005174C
030000F1 12BFC400 N S P N - DISPATCH OPENPOOL    S 14:39:01.777    -     15D37080 XM 1406E030 QR 12ACCA40 1406E0300005220C
030201F9 12BFE400 N S P N - FCPSWAIT EDZFILE     C 14:38:47.748    -     15D37200 XM 12B0A7D8 QR 1333A2C5 12B0A7D80005026C
030A006B 12BFEB00 N S P N - FCPSWAIT EDZFILE     C 14:38:49.107    -     15D37800 XM 12B0A1B8 QR 1333A2C5 12B0A1B80005061C
030C014B 12E31780 N S P N - FCPSWAIT EDZFILE     C 14:38:47.193    -     15D37980 XM 14076030 QR 1333A2C5 140760300005030C
030E0053 12BFF080 N S P N - FCPSWAIT EDZFILE     C 14:38:47.278    -     15D37B00 XM 134F4650 QR 1333A2C5 134F46500004961C
03100141 12BFCB00 N S P N - DISPATCH OPENPOOL    S 14:38:59.210    -     15D37C80 XM 14076AE8 QR 12ACC890 14076AE80005189C
0380008D 12BFF780 N S P N - FCPSWAIT EDZFILE     C 14:38:53.214    -     15D6E080 XM 134F4340 QR 1333A2C5 134F43400005123C
038600DD 12B3A780 N S P N - DISPATCH OPENPOOL    S 14:39:09.293    -     15D6E500 XM 1406EC70 QR 12ACCCB0 1406EC700005306C
0388010D 12BFE780 N S P N - DISPATCH OPENPOOL    S 14:38:57.746    -     15D6E680 XM 14076650 QR 12ACCC50 140766500005182C
038A0083 12BFD400 N S P N - DISPATCH OPENPOOL    S 14:38:56.327    -     15D6E800 XM 140761B8 QR 12ACC770 140761B80005177C
038C014D 12BFDB00 N S P N - FCPSWAIT EDZFILE     C 14:38:52.288    -     15D6E980 XM 14076340 QR 1333A2C5 140763400005100C
0392017D 12E4E400 N S P N - FCPSWAIT EDZFILE     C 14:38:52.101    -     15D6EE00 XM 12B0A650 QR 1333A2C5 12B0A6500005101C
040001BF 12E4E780 N S N N - CDB2RDYQ *POOL       M 14:38:53.235    -     15D6F080 XM 134F4DF8 L8 13455244 134F4DF80005125C
04020085 12E6BB00 N S P N - FCPSWAIT EDZFILE     C 14:38:49.131    -     15D6F200 XM 12B0A030 QR 1333A2C5 12B0A0300005063C
040400EF 12E4E080 N S P N - FCPSWAIT EDZFILE     C 14:38:52.967    -     15D6F380 XM 134F4AE8 QR 1333A2C5 134F4AE80005121C
040E01A3 12E31400 N S P N - FCPSWAIT EDZFILE     C 14:38:52.403    -     15D6FB00 XM 140767D8 QR 1333A2C5 140767D80005107C
04120097 12BFB400 N S P N - DISPATCH OPENPOOL    S 14:39:04.017    -     15D6FE00 XM 1406E7D8 QR 12ACCCE0 1406E7D80005246C
```

Problem One – What we know

- Customer’s DB2 tasks were hung
- Dump was taken at 15:01:47.569848 GMT
- CICS was running up until dump time
- DS=3 showed all 15 L8 TCBs used with 9 transactions waiting for L8 TCB
- DS=3 showed the transactions waiting since around 14:38 GMT
- Prevalent DB2 waits were DISPATCH OPENPOOL and CDB2RDYQ
- DB2=3 showed 15 tasks owning an L8 TCB and 13 of the 15 owning a Thread
- DB2 Global Block showed TCBLIMIT of 15 coded and Pool Thread limit of 13
- Every DB2ENTRY had THREADLIMIT(0) and THREADWAIT(POOL)
  - This causes all DB2 calls to use Pool Threads
- All 13 Thread owners were in CICS
- All 13 thread owners were in a String Wait for EDZFILE

ABEND AD2U

AD2U Abend

- Explanation: An attempt to create a DB2 thread by the TCB servicing the DB2 request failed. The installed DB2CONN specifies THREADERROR(N906D) or THREADERROR(ABEND)

- System Action:
  - If THREADERROR(N906D) is specified in the DB2CONN, processing continues. A -906 sqlcode is returned to the application and a transaction dump is taken with abend code AD2U
  - If THREADERROR(ABEND) is specified in the DB2CONN, the task is abnormally terminated with a CICS transaction dump

- User Response: Examine the dump to determine why the create thread failed

Problem Two – AD2U Abend

- Customer indicated all of their high priority CICS/DB2 transactions fail with AD2U Abend

- Customer was told by consultant they could disable the DB2ENTRY and the transaction for that entry would be redirected to the pool without problems

- Customer came into support with AD2U Transaction Dump

MSGUSR Output

- DFHDB2072
  - Transaction SQLZ , task 00136 has been directed to the pool as DB2ENTRY SQLZ is disabled.
- DFHDU0203I
  - A transaction dump was taken for dumpcode: AD2U
- DFHAC2250
  - The coordinator system has indicated that the current unit of work is to be backed out. Transaction SQLZ running program SQLASMZ term TC07 has been abnormally terminated with abend ASP3.

CICS Trace from Transaction Dump

```
00136 QR    PG 0901 PGPG  ENTRY INITIAL_LINK          SQLASMZ                                                        =000016=
00136 QR    LD 0001 LDLD  ENTRY ACQUIRE_PROGRAM       138EA100                                                       =000017=
00136 QR    LD 0002 LDLD  EXIT  ACQUIRE_PROGRAM/OK    000C2000,000C2000,2A8,REUSABLE,SDSA,OLD_COPY,                  =000018=
00136 QR    AP 1940 APLI  ENTRY START_PROGRAM         SQLASMZ,CEDF,FULLAPI,EXEC,NO,13845F98,00000000 , 00000000      =000019=
00136 QR    AP 2520 ERM   ENTRY ASSEMBLER-APPLICATION-CALL-TO-TRUE(DFHD2PXT)                                         =000022=
00136 L8001 AP 3180 D2EX1 ENTRY APPLICATION           REQUEST EXEC SQL SELECT                                        =000027=
00136 L8001 ME 0301 MEME  ENTRY SEND_MESSAGE          818,13AFE380 , 00000004,137F0CC1 , 00000005,13AB8108 ,         =000028=
00136 L8001 DU 0500 DUDT  ENTRY INQUIRE_SYSTEM_DUMPCODE DB2072                                                       =000035=
00136 L8001 DU 0600 DUTM  ENTRY INQUIRE_SYSTEM_DUMPCODE DB2072                                                       =000036=
00136 L8001 DU 0601 DUTM  EXIT  INQUIRE_SYSTEM_DUMPCODE/EXCEPTION DUMPCODE_NOT_FOUND,0,0,,,,                         =000037=
00136 L8001 DU 0501 DUDT  EXIT  INQUIRE_SYSTEM_DUMPCODE/EXCEPTION DUMPCODE_NOT_FOUND,0,0,,,,                         =000038=
00136 L8001 AP F600 TDA   ENTRY WRITE_TRANSIENT_DATA  CDB2,137F1E5C , 00000001,NO                                    =000041=
00136 QR    DD 0301 DDLO  ENTRY LOCATE                12AF5F80,00049368,DCTE,CDB2                                    =000042=
00136 QR    DD 0302 DDLO  EXIT  LOCATE/OK             1381F1F0 , C4C3E3C5                                            =000043=
00136 QR    DD 0301 DDLO  ENTRY LOCATE                12AF5F80,00049368,DCTE,CSSL                                    =000044=
00136 QR    DD 0302 DDLO  EXIT  LOCATE/OK             138245D0 , C4C3E3C5                                            =000045=
00136 L8001 AP F601 TDA   EXIT  WRITE_TRANSIENT_DATA/OK                                                              =000046=
00136 L8001 ME 0314 MEME  EVENT ISSUE-MVS-FREEMAIN                                                                   =000047=
00136 L8001 ME 0315 MEME  EVENT MVS-FREEMAIN-COMPLETE                                                                =000048=
00136 L8001 ME 0302 MEME  EXIT  SEND_MESSAGE/OK                                                                      =000049=
00136 L8001 AP 3250 D2D2  ENTRY DB2_API_CALL          13B12330                                                       =000050=
00136 L8001 AP 3265 D2D2  *EXC* FAILED_TO_CREATE_THREAD_FOR_PLAN(DEFAULT )                                          *=000051=
00136 L8001 AP 3251 D2D2  EXIT  DB2_API_CALL/EXCEPTION CREATE_THREAD_FAILED                                          =000052=
00136 L8001 DU 0101 DUDU  ENTRY TRANSACTION_DUMP      AD2U,13AB73D8 , 000000C8,YES,YES,YES,YES,YES,YES               =000053=
AP 3265 D2D2 *EXC* - FAILED_TO_CREATE_THREAD_FOR_PLAN(DEFAULT )
     TASK-00136 KE_NUM-0046 TCB-L8001/007ABE88 RET-93DEC3BA TIME-20:45:22.9579484440 INTERVAL-00.0456906879*       =000051=
   1-0000  13B12410 13B1239C 13245BF2 13B123B4  13B123FC 93B124E0 00000000 00000000  *..........$2........l..\........*
   2-0000  C6D9C240 00010004 13B1248C 00050008  00F30034 00000000 00180805 00010400  *FRB .............3..............*
     0020  00000000 00000000 00000000 00000000                                       *................                *
   3-0000  C4C5C6C1 E4D3E340                                                         *DEFAULT                         *
   4-0000  03006EC4 C6C8C4F2 C3E2C240 40404040  C2864D97 654FA05E 13AB7000 13AB73D8  *..>DFHD2CSB     Bf(p.|.;.......Q*
     0020  13AFE370 007ABE88 00000000 00000000  C2864FCD 4CB9EBD3 00000000 00000000  *..T..:.h........Bf|.<..L........*
     0040  13AB7444 00000000 00000000 00000000  00000000 00000000 00000000 00000000  *................................*
     0060  00000000 00000000 00000000 C4C5C6C1  E4D3E340 E4E2C1E2 E2C3F140 40404040  *............DEFAULT USASSC1     *
     0080  40404040 D7D6D6D3 E2D8D3E9 F0F0F0F2  7F58B538 C2864D97 6A174790 00000000  *    POOLSQLZ0002"...Bf(p........*
     00A0  00000000 00000000 00000000 00000000  00000000 40800000 00000002 00000000  *.................... ...........*
     00C0  00000000 00000000 00000000 40404040  40404040 40404040 40404040 FF4D4400  *............                .(..*
     00E0  C6D9C240 00010004 13B1248C 00050008  00F30034 00000000 00180805 00010400  *FRB .............3..............*
     0240  6E6EE399 81838540 E2A38199 A3406E6E  1F00120C C4C9E2E2 00000000 00000000  *>>Trace Start >>....DISS........*
     0260  2000136C C1E2E2D6 00000000 00000000  2100136C C3E3C8C4 00000008 00F30034  *...%ASSO...........%CTHD.....3..*
     0280  1800118C E3C5D9D4 00000000 00000000  1900118C C4C9E2E2 00000000 00000000  *....TERM............DISS........*
     02A0  1A00120C C1E2E2D6 00000000 00000000  1B00120C C3E3C8C4 00000000 00000000  *....ASSO............CTHD........*
     02C0  1C00120C C1D7C940 00000000 00000000  1D00120C E2E8D5C3 00000000 00000000  *....API ............SYNC........*
     02E0  1E00120C E3C5D9D4 00000000 00000000  4C4CE399 81838540 C5958440 40404C4C  *....TERM........<<Trace End   <<*
   5-0000  00DE6EC4 C6C8C4F2 D3D6E340 40404040  E2D8D3E9 0005D680 13AB73D8 13B12330  *..>DFHD2LOT     SQLZ..O....Q....*
     0020  00000000 0004AFF0 00000000 13AB746C  001007D0 00000000 7F58B538 FF4D3A00  *.......0.......%...}...."....(..*
     0040  00000000 00000000 00000000 00000000  00000000 C4C5C6C1 E4D3E340 81000000  *....................DEFAULT a...*
     0060  A4008000 00000000 00240000 00000000  C9E8D5E7 F8404040 C2865752 84FADA19  *u..............IYNX8   Bf..d...*
     0080  E4E2C1E2 E2C3F140 40404040 40404040  00000000 00000000 C7C2C9C2 D4C9E8C1  *USASSC1         ........GBIBMIYA*
     00A0  C9E8C3D5 E3C3F2F3 86575284 FADA0000  00000000 00000000 00000000 00000000  *IYCNTC23f..d....................*
```

AP 3265 D2D2 *EXC* Trace Entry

Field 1 – DB2 Parameter List

Field 2 – Function Request Block (FRB)

Field 3 – Plan Name = DEFAULT

Field 4 – DFHD2CSB Control Block

- **+6C CSB_PLAN_NAME = DEFAULT**
- **+74 CSB_PRIMARY_AUTH_NAME = USASSC1**
- **+84 CSB_TYPE = POOL**
- **+88 CSB_TRANSID = SQLZ**
- **+E0 CSB_FRB = 00F30034**
- **+240 CSB_TRACE_HEAD**

Filed 5 – DFHD2LOT Control Block

CSUB TRACE Format

Bytes	Content
0-3	Trace request / transaction number
4-7	Trace request Possible values:
ABRT	Abort request
API	SQL or IFI request
ASSO	Associate request
COMM	Commit request
CTHD	Create thread request
DISS	Dissociate request
ERRH	Error handler request
IDEN	Identify request
PREP	Prepare request
PSGN	Partial signon request
SIGN	Full signon request
SYNC	Single phase request
TERM	Terminate thread request
TIDN	Terminate identify request
TSGN	Terminate signon request
*REC	Recovery routine entered
8-9	Reserved
10-11	FRB return code
12-15	FRB reason code

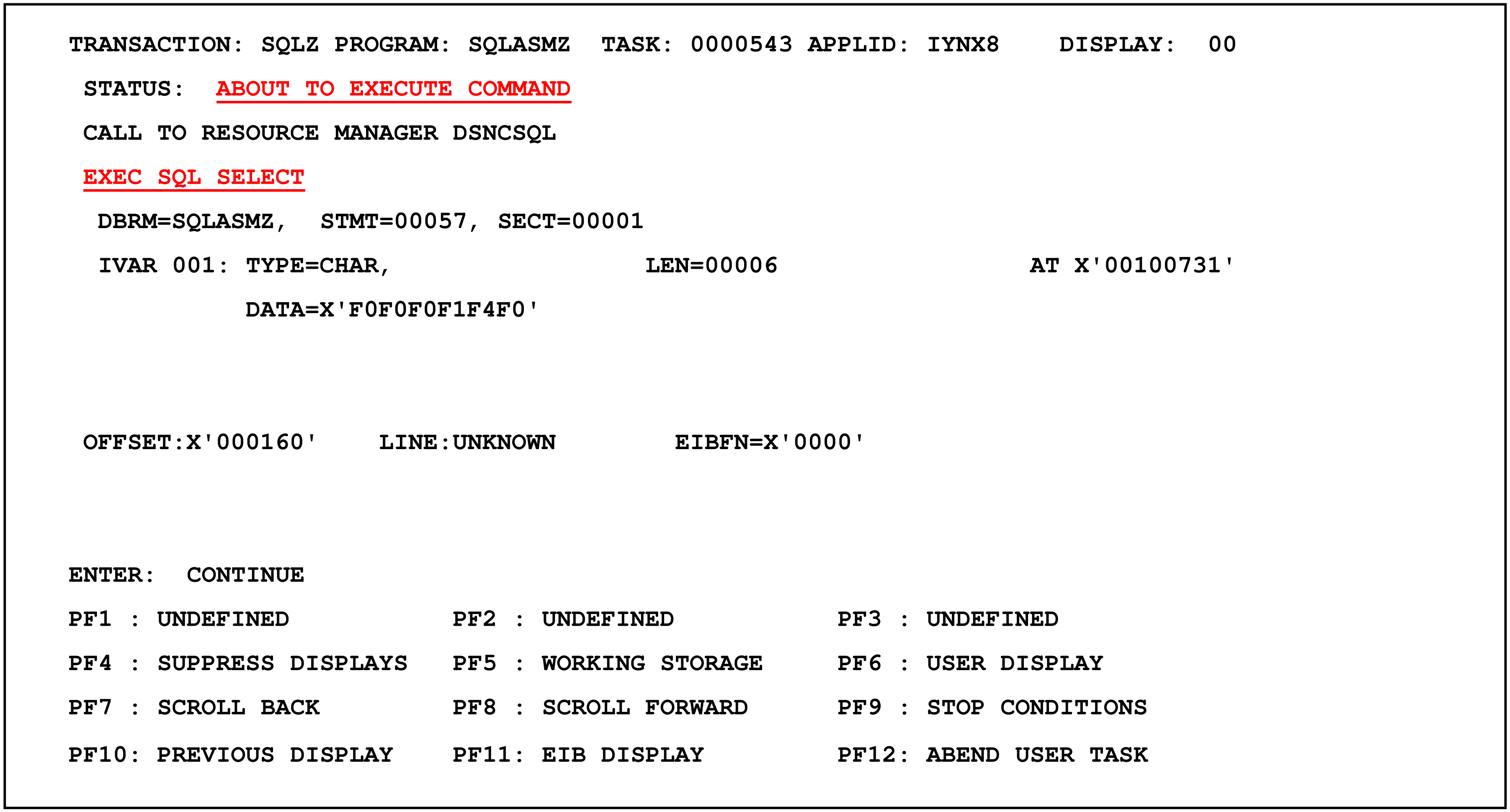
AD2U Abend – What we know so far

- Customer disabled DB2ENTRY for transaction so all requests would overflow to the Pool
  - This did work according to message DFHDB2072
- Transaction SQLZ task number 00136 received AD2U Abend
- CICS trace showed a Create Thread error for Plan DEFAULT and AUTHID USASSC1
- CSUB Traceback showed a Create Thread (CTHD) entry with FRB return and reason code of 00F30034

FRB code

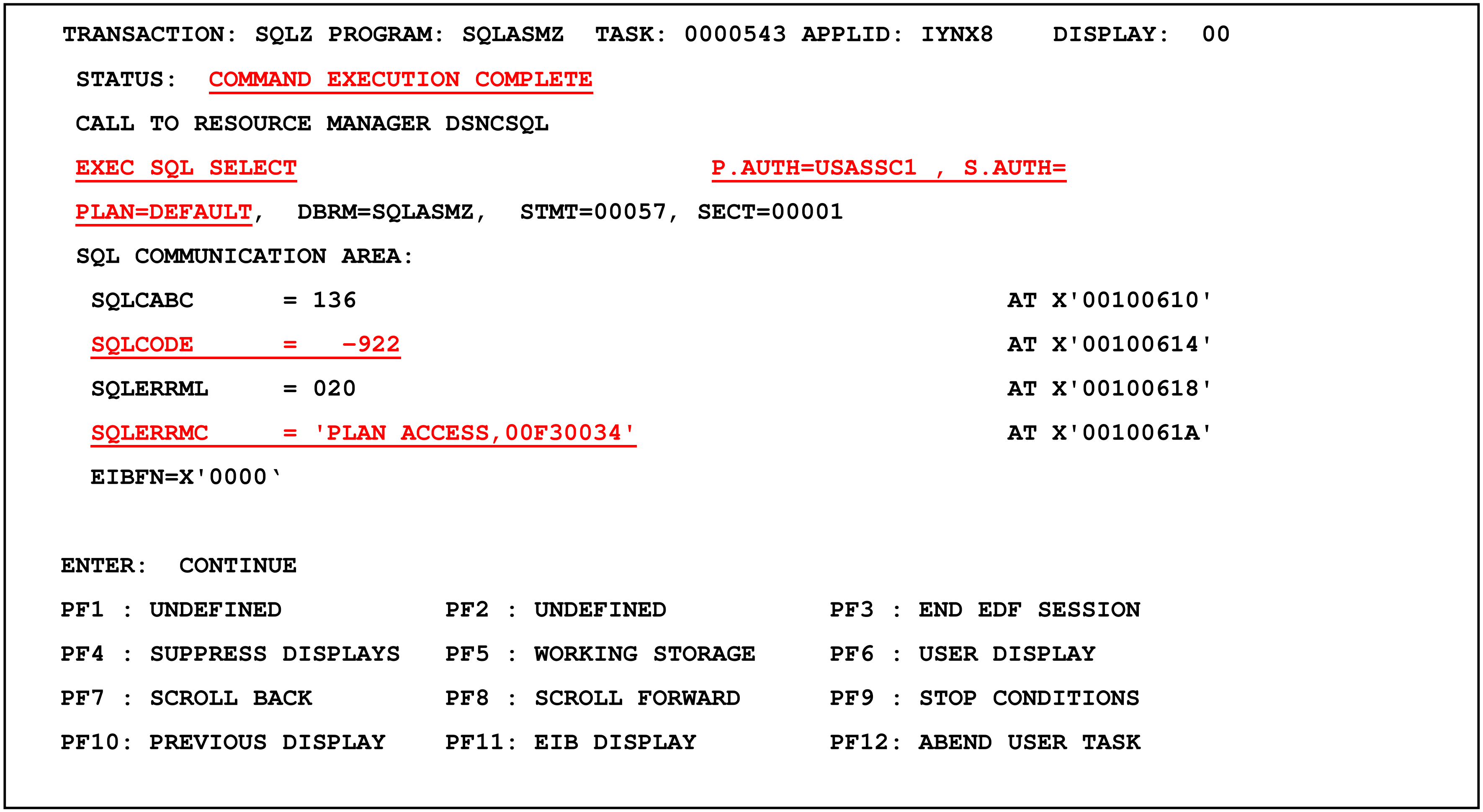
- 00F30034
  - Explanation: The authorization ID associated with this connection is not authorized to use the specified plan name or the specified plan name does not exist.
  - System Action: The request to allocate a plan to the authorization ID is denied.
  - User Response: Verify that the correct plan name was specified. If this plan exists, then request execution authority to the plan from either the owner of the plan or from another person given authority to grant execution authority to the plan.

CEDF for Transaction Receiving AD2U



```
TRANSACTION: SQLZ PROGRAM: SQLASMZ  TASK: 0000543 APPLID: IYNX8    DISPLAY:  00
 STATUS:  ABOUT TO EXECUTE COMMAND
 CALL TO RESOURCE MANAGER DSNCSQL
 EXEC SQL SELECT
  DBRM=SQLASMZ,  STMT=00057, SECT=00001
  IVAR 001: TYPE=CHAR,                 LEN=00006                 AT X'00100731'
            DATA=X'F0F0F0F1F4F0'



 OFFSET:X'000160'    LINE:UNKNOWN        EIBFN=X'0000'



ENTER:  CONTINUE
PF1 : UNDEFINED          PF2 : UNDEFINED          PF3 : UNDEFINED
PF4 : SUPPRESS DISPLAYS  PF5 : WORKING STORAGE    PF6 : USER DISPLAY
PF7 : SCROLL BACK        PF8 : SCROLL FORWARD     PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY   PF11: EIB DISPLAY        PF12: ABEND USER TASK
```

CEDF for Transaction Receiving AD2U



```
TRANSACTION: SQLZ PROGRAM: SQLASMZ  TASK: 0000543 APPLID: IYNX8    DISPLAY:  00
 STATUS:  COMMAND EXECUTION COMPLETE
 CALL TO RESOURCE MANAGER DSNCSQL
 EXEC SQL SELECT                            P.AUTH=USASSC1 , S.AUTH=
 PLAN=DEFAULT,  DBRM=SQLASMZ,  STMT=00057, SECT=00001
 SQL COMMUNICATION AREA:
  SQLCABC      = 136                                       AT X'00100610'
  SQLCODE      =   -922                                    AT X'00100614'
  SQLERRML     = 020                                       AT X'00100618'
  SQLERRMC     = 'PLAN ACCESS,00F30034'                    AT X'0010061A'
  EIBFN=X'0000`


ENTER:  CONTINUE
PF1 : UNDEFINED          PF2 : UNDEFINED          PF3 : END EDF SESSION
PF4 : SUPPRESS DISPLAYS  PF5 : WORKING STORAGE    PF6 : USER DISPLAY
PF7 : SCROLL BACK        PF8 : SCROLL FORWARD     PF9 : STOP CONDITIONS
PF10: PREVIOUS DISPLAY   PF11: EIB DISPLAY        PF12: ABEND USER TASK
```

AD2U Abend – What we know

- Customer disabled DB2ENTRY for transaction so all requests would overflow to the Pool
  - This did work according to message DFHDB2072
- Transaction SQLZ task number 00136 received AD2U Abend
- CICS trace showed a Create Thread error for Plan DEFAULT and AUTHID USASSC1
- CSUB Traceback showed a Create Thread (CTHD) entry with FRB return and reason code of 00F30034
  - FRB code indicates the plan does not exist or the Authorization ID was not correct
- Primary Authorization ID USASSC1 was not allowed access to Plan DEFAULT which was Plan Name for Pool Threads
- Instead of disabling the DB2ENTRY the customer could have set THREADLIMIT(0) and THREADWAIT(POOL) on the DB2ENTRY
  - This would keep the Plan Name defined on the DB2ENTRY and allow all requests to flow to the Pool Threads

Summary

- Resource Definitions
  - DB2CONN
  - DB2ENTRY
  - DB2TRAN
- Documentation Needed and Time Considerations
- CICS and DB2 VERBEXIT Displays for Debugging
  - DFHPDxxx
  - DSNWDMP
- Problem One – Hang
- Problem Two – AD2U Abend