Java Access to CA-IDMS Data at BT

Steve Terry

BT

Java Access to CA-IDMS Data at BT
Session Abstract

- BT, one of the world's leading providers of communications solutions and services, has recently introduced direct access to its major CA-IDMS CSS (Customer Service System) databases using the CA-IDMS Server JDBC interface.

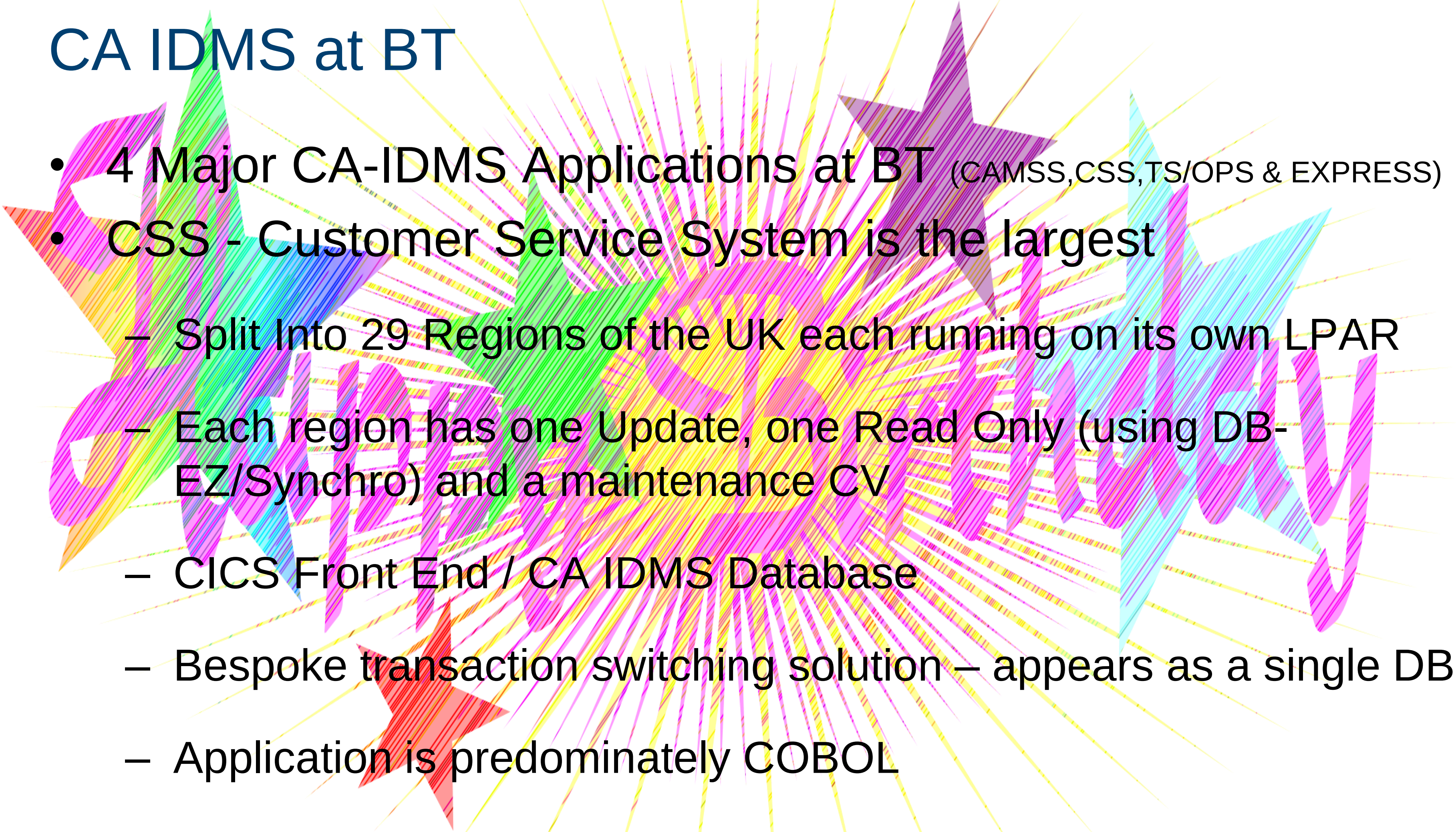
- This session explains why, how and a few of the lessons learnt so far.

- Steve Terry has worked for BT for 40 years, 26 of these on CA-IDMS as a Systems Programmer, DBA and manager

Java Access to CA-IDMS Data at BT
Agenda

- CA IDMS at BT
- Mainframe Modernisation Program
- JCSS (java access to CSS) Overview
- CA-IDMS JDBC Configuration
- Some Lessons Learnt / Things to Watch For
- Questions

Java Access to CA-IDMS Data at BT
CA IDMS at BT



- 4 Major CA-IDMS Applications at BT (CAMSS,CSS,TS/OPS & EXPRESS)
- CSS - Customer Service System is the largest
  – Split Into 29 Regions of the UK each running on its own LPAR
  – Each region has one Update, one Read Only (using DB-EZ/Synchro) and a maintenance CV
  – CICS Front End / CA IDMS Database
  – Bespoke transaction switching solution – appears as a single DB
  – Application is predominately COBOL
  – Bespoke Common Middleware solution provides application services

Java Access to CA-IDMS Data at BT
Mainframe Modernisation Program

- BT Recently undertook a review of the Mainframe Platform
  - Driven by Cost Reduction, Sustainability, System Rationalisation
  - Re-platforming of all mainframe applications was explored
  - Outcome was to retain the platform, seek to reduce costs, address sustainability issues and modernise where possible to become a 'better fit' with the rest of the BT estate
  - Several proof of concepts being undertaken including enabling java access to CA-IDMS to provide a CSS 'data as a service' interface running under Linux on a separate platform

Java Access to CA-IDMS Data at BT
JCSS IDMS Gateway Overview

- A Java based structured gateway that will enable authorised users to obtain CSS data as a web service
- Exploits CA-Server JDBC (Java Database Connectivity) to connect directly to the CSS CA-IDMS databases (via IDMSRD – read only CV at this stage)
- Business logic / application code being replicated using java in the JCSS application layer.
- Helping to move processing costs away from the Mainframe by substituting CSS COBOL transactions running under CICS on the mainframe with web services calls running on Linux under JCSS, reducing mainframes CPU usage, but that is not the main driver

Java Access to CA-IDMS Data at BT
JCSS IDMS Gateway Overview

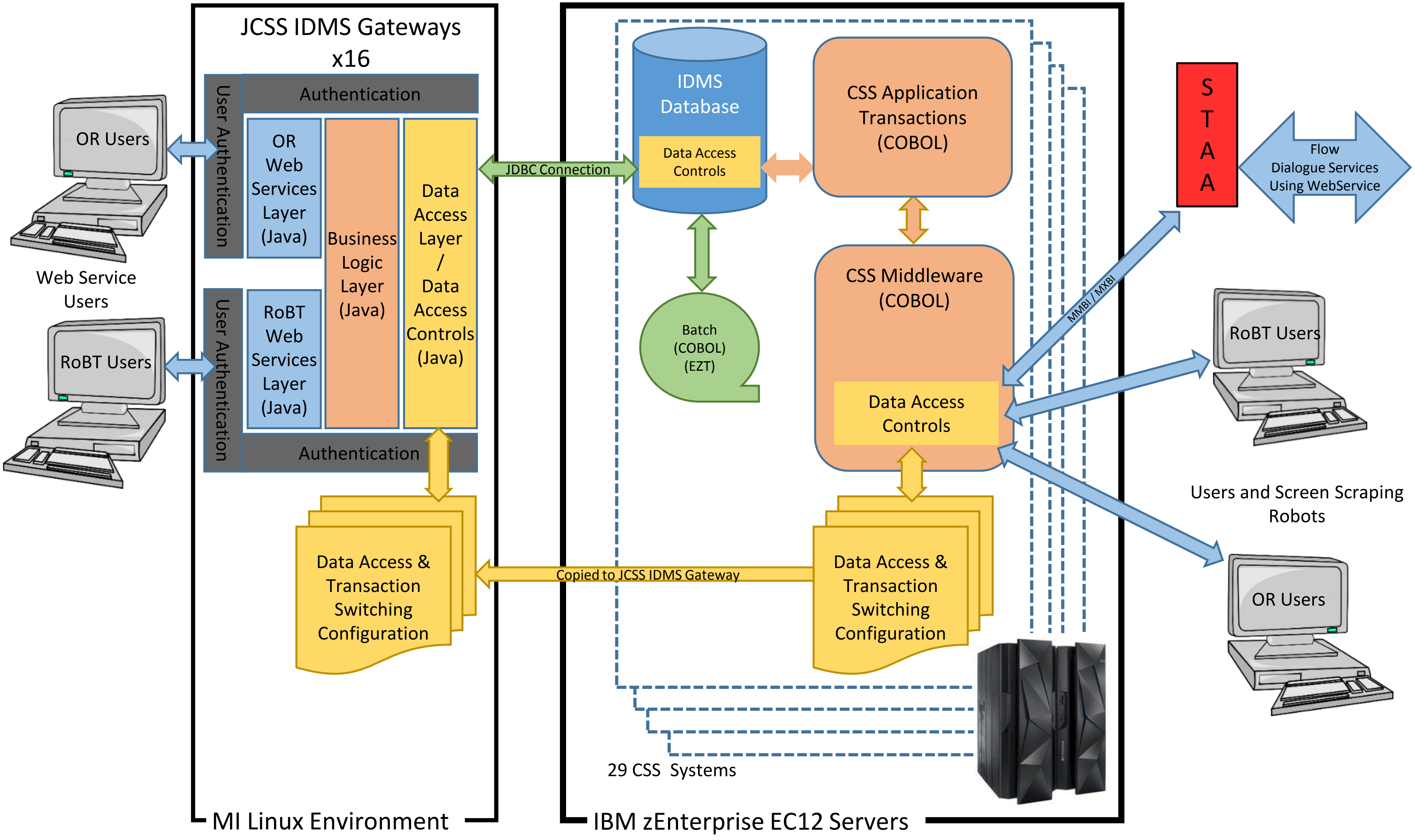
- Native SQL to be used whenever possible
  to avoid the overhead of maintaining table procedures

- We will be using the IDMS R19 Virtual Foreign Key feature

- Requires some schema changes – but not material changes
  - for example

  - FILLER redefined with alternative element name

  - Reordering redefined elements within a record
    - where redefined elements are being used as set keys. We need to reorder the definitions / redefinitions to make the key values the base elements



JCSS IDMS Gateways x16

Authentication

User Authentication

OR Web Services Layer (Java)

Business Logic Layer (Java)

Data Access Layer / Data Access Controls (Java)

User Authentication

RoBT Web Services Layer (Java)

Authentication

Data Access & Transaction Switching Configuration

MI Linux Environment

OR Users

Web Service Users

RoBT Users

JDBC Connection

IDMS Database

Data Access Controls

CSS Application Transactions (COBOL)

Batch (COBOL) (EZT)

CSS Middleware (COBOL)

Data Access Controls

Copied to JCSS IDMS Gateway

Data Access & Transaction Switching Configuration

29 CSS Systems

IBM zEnterprise EC12 Servers

MMBI / MXBI

S T A A

Flow Dialogue Services Using WebService

RoBT Users

Users and Screen Scraping Robots

OR Users

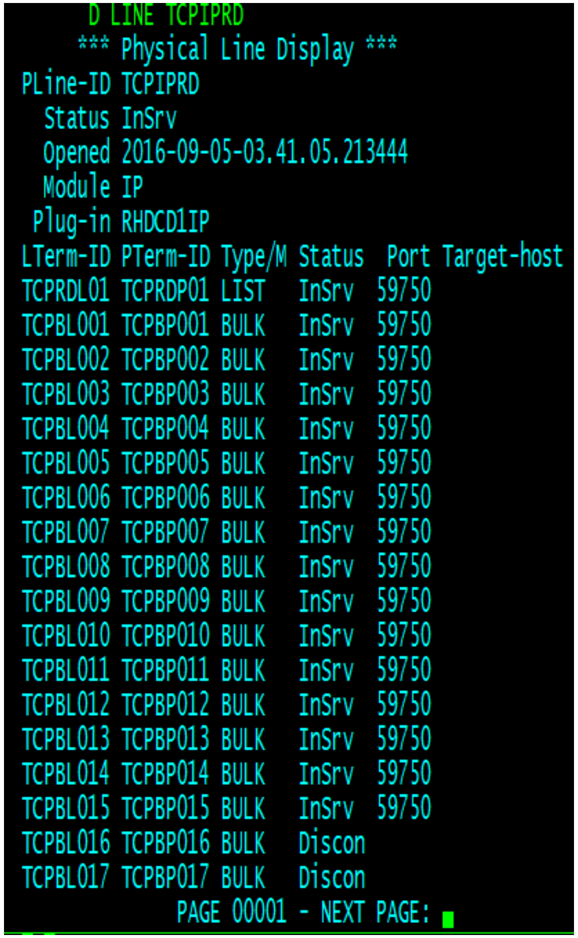
Java Access to CA-IDMS Data at BT
CA-IDMS JDBC Configuration

- Multiple JCSS (16) servers connect to each (29) CSS CA-IDMS Read Only CV

- Connection pooling is used in java, with min of 1 and max of 5 connections from each JCSS server to IDMS
  - *usually* 16 active connections to each CA-IDMS system
  - *could* get up to 80 active connections to each CA-IDMS system

- In CA-IDMS a TCPIP line is defined with 200 bulk LTERM/PTERM – so plenty of connections available

- The CV is mixed use – OLTP, Batch & now CA-Server The number of **ACTIVE TASKS** created by CA-Server could become an issue……..



```
      D LINE TCPIPRD
     *** Physical Line Display ***
PLine-ID TCPIPRD
  Status InSrv
  Opened 2016-09-05-03.41.05.213444
  Module IP
 Plug-in RHDCD1IP
LTerm-ID PTerm-ID Type/M Status  Port Target-host
TCPRDL01 TCPRDP01 LIST   InSrv   59750
TCPBL001 TCPBP001 BULK   InSrv   59750
TCPBL002 TCPBP002 BULK   InSrv   59750
TCPBL003 TCPBP003 BULK   InSrv   59750
TCPBL004 TCPBP004 BULK   InSrv   59750
TCPBL005 TCPBP005 BULK   InSrv   59750
TCPBL006 TCPBP006 BULK   InSrv   59750
TCPBL007 TCPBP007 BULK   InSrv   59750
TCPBL008 TCPBP008 BULK   InSrv   59750
TCPBL009 TCPBP009 BULK   InSrv   59750
TCPBL010 TCPBP010 BULK   InSrv   59750
TCPBL011 TCPBP011 BULK   InSrv   59750
TCPBL012 TCPBP012 BULK   InSrv   59750
TCPBL013 TCPBP013 BULK   InSrv   59750
TCPBL014 TCPBP014 BULK   InSrv   59750
TCPBL015 TCPBP015 BULK   InSrv   59750
TCPBL016 TCPBP016 BULK   Discon
TCPBL017 TCPBP017 BULK   Discon
              PAGE 00001 - NEXT PAGE:
```

Java Access to CA-IDMS Data at BT
Lessons Learnt – CA-IDMS Active Tasks

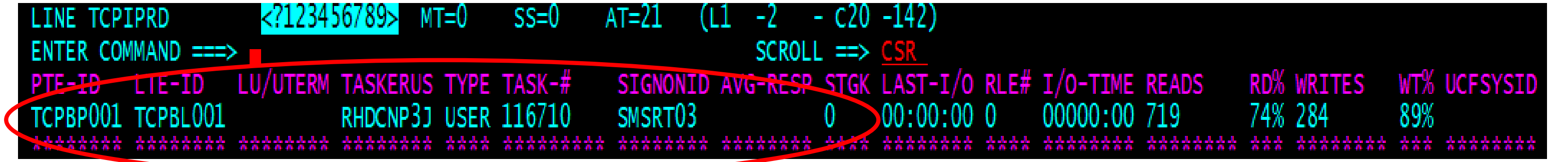
- Suspend Strategy for Pseudo-Conversational Processing has an impact
- Controls when CA-IDMS Server will issue a suspend which ends the CA-IDMS task and frees resources on the CV
- This behaviour can be customised by selecting a "suspend strategy“
- Pre-defined set of strategies available appropriate for particular types of application.

Java Access to CA-IDMS Data at BT
Lessons Learnt – CA-IDMS Active Tasks

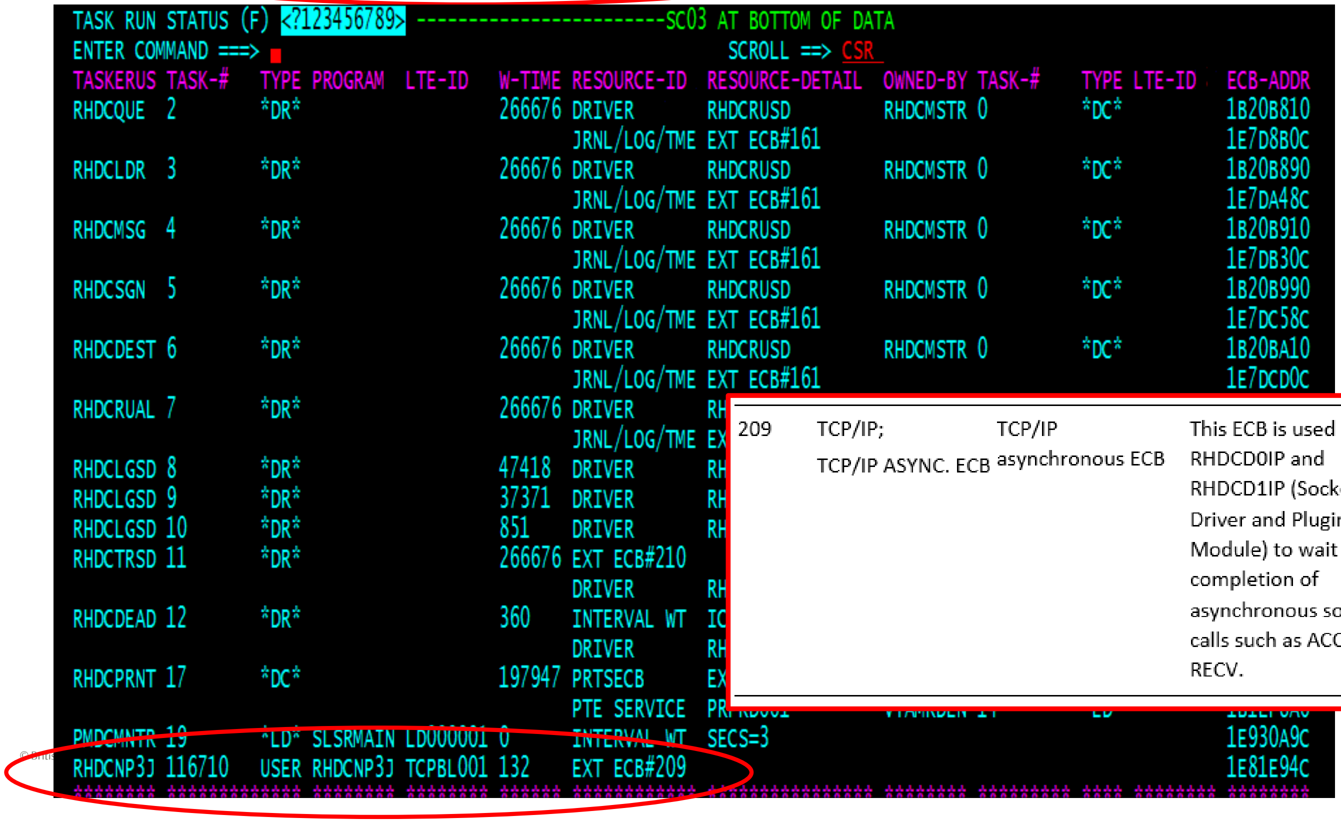
- Interactive—Intended for use by applications with a user interface, in which database activity and user input are intermixed.
  Task remains active on connect. Suspends when the transaction is committed.
  This is the default strategy for ODBC and non-pooled JDBC connections.
- Service—Intended for use by JDBC applications that run in an application server that pools connections and allocates them temporarily to units of work that that access the database one or more times without waiting for user input.
  Task remains active on commit – until it determines the connection is idle.
  This is the default for pooled JDBC connections.
- Batch—Intended for use by applications access the database many times and terminate without waiting for user input. The driver does not use pseudo-conversational processing at all.
- Custom

Java Access to CA-IDMS Data at BT
Lessons Learnt – CA-IDMS Active Tasks

INTERACTIVE – after CONNECT – no SQL has been run



```
LINE TCPIPRD       <?123456789>  MT=0    SS=0    AT=21   (L1  -2   - C20 -142)
ENTER COMMAND ===>                                       SCROLL ==> CSR
PTE-ID   LTE-ID   LU/UTERM TASKERUS TYPE TASK-#    SIGNONID AVG-RESP STGK LAST-I/O RLE# I/O-TIME READS    RD% WRITES   WT% UCFSYSID
TCPBP001 TCPBL001          RHDCNP3J USER 116710    SMSRT03         0      00:00:00 0    00000:00 719      74% 284      89%
******** ******** ******** ******** **** ********* ******** ******** **** ******** **** ******** ******** *** ******** *** ********
```

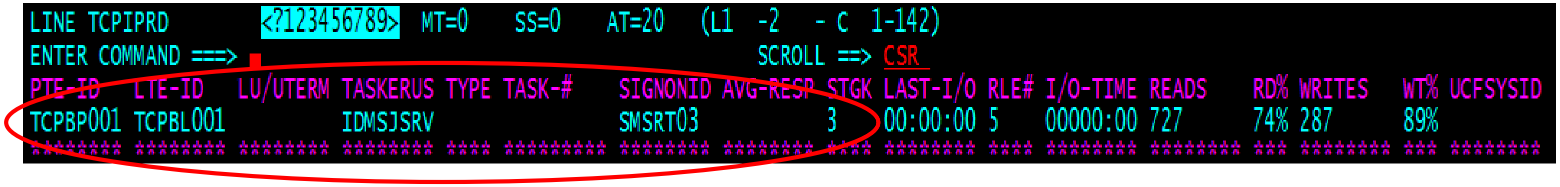


```
TASK RUN STATUS (F) <?123456789> ----------------------SC03 AT BOTTOM OF DATA
ENTER COMMAND ===>                                      SCROLL ==> CSR
TASKERUS TASK-#    TYPE PROGRAM  LTE-ID   W-TIME RESOURCE-ID RESOURCE-DETAIL OWNED-BY TASK-#    TYPE LTE-ID   ECB-ADDR
RHDCQUE  2         *DR*                   266676 DRIVER      RHDCRUSD        RHDCMSTR 0         *DC*          1B20B810
                                                 JRNL/LOG/TME EXT ECB#161                                     1E7D8B0C
RHDCLDR  3         *DR*                   266676 DRIVER      RHDCRUSD        RHDCMSTR 0         *DC*          1B20B890
                                                 JRNL/LOG/TME EXT ECB#161                                     1E7DA48C
RHDCMSG  4         *DR*                   266676 DRIVER      RHDCRUSD        RHDCMSTR 0         *DC*          1B20B910
                                                 JRNL/LOG/TME EXT ECB#161                                     1E7DB30C
RHDCSGN  5         *DR*                   266676 DRIVER      RHDCRUSD        RHDCMSTR 0         *DC*          1B20B990
                                                 JRNL/LOG/TME EXT ECB#161                                     1E7DC58C
RHDCDEST 6         *DR*                   266676 DRIVER      RHDCRUSD        RHDCMSTR 0         *DC*          1B20BA10
                                                 JRNL/LOG/TME EXT ECB#161                                     1E7DCD0C
RHDCRUAL 7         *DR*                   266676 DRIVER      RH
                                                 JRNL/LOG/TME EX
RHDCLGSD 8         *DR*                   47418  DRIVER      RH
RHDCLGSD 9         *DR*                   37371  DRIVER      RH
RHDCLGSD 10        *DR*                   851    DRIVER      RH
RHDCTRSD 11        *DR*                   266676 EXT ECB#210
                                                 DRIVER      RH
RHDCDEAD 12        *DR*                   360    INTERVAL WT IC
                                                 DRIVER      RH
RHDCPRNT 17        *DC*                   197947 PRTSECB     EX
                                                 PTE SERVICE PR
RHDCMNTR 19        *LD* SLSRMAIN LD000001 0      INTERVAL WT SECS=3                                           1E930A9C
RHDCNP3J 116710    USER RHDCNP3J TCPBL001 132    EXT ECB#209                                                  1E81E94C
******** ********* ******** ******** ****** ************ *************** ******** ********* **** ******** ********
```

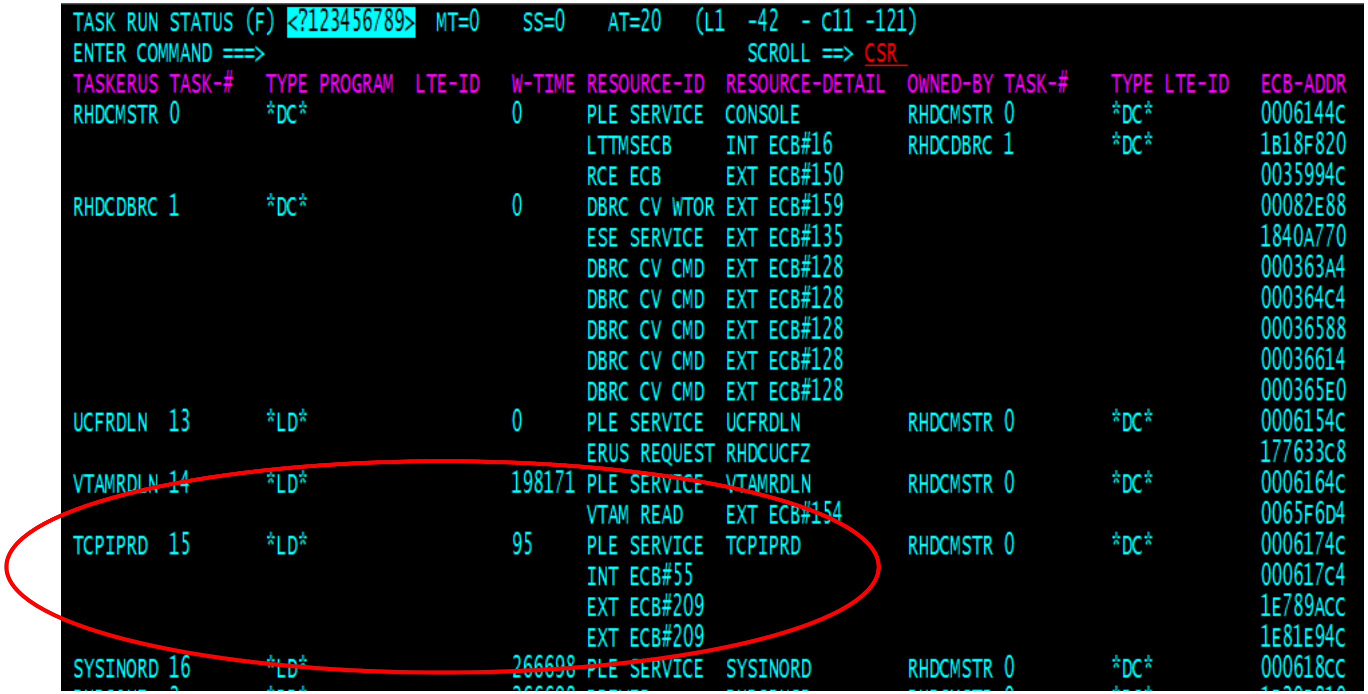
209	TCP/IP; TCP/IP ASYNC. ECB	TCP/IP asynchronous ECB	This ECB is used by RHDCD0IP and RHDCD1IP (Socket Line Driver and Plugin Module) to wait for completion of asynchronous socket calls such as ACCEPT and RECV.

Java Access to CA-IDMS Data at BT
Lessons Learnt – CA-IDMS Active Tasks

INTERACTIVE – after first SQL statement has been run



```
LINE TCPIPRD        <?123456789> MT=0    SS=0    AT=20    (L1  -2   - C  1-142)
ENTER COMMAND ===>                                     SCROLL ==> CSR
PTE-ID   LTE-ID   LU/UTERM TASKERUS TYPE TASK-#    SIGNONID AVG-RESP STGK LAST-I/O RLE# I/O-TIME READS    RD% WRITES   WT% UCFSYSID
TCPBP001 TCPBL001          IDMSJSRV                SMSRT03              3 00:00:00 5    00000:00 727      74% 287      89%
******** ******** ******** ******** **** ********* ******** ******** **** ******** **** ******** ******** *** ******** *** ********
```



```
TASK RUN STATUS (F) <?123456789> MT=0    SS=0    AT=20    (L1  -42  - c11 -121)
ENTER COMMAND ===>                               SCROLL ==> CSR
TASKERUS TASK-#    TYPE PROGRAM  LTE-ID   W-TIME RESOURCE-ID  RESOURCE-DETAIL OWNED-BY TASK-#    TYPE LTE-ID   ECB-ADDR
RHDCMSTR 0         *DC*                   0      PLE SERVICE  CONSOLE         RHDCMSTR 0         *DC*          0006144C
                                                 LTTMSECB     INT ECB#16      RHDCDBRC 1         *DC*          1B18F820
                                                 RCE ECB      EXT ECB#150                                      0035994C
RHDCDBRC 1         *DC*                   0      DBRC CV WTOR EXT ECB#159                                      00082E88
                                                 ESE SERVICE  EXT ECB#135                                      1840A770
                                                 DBRC CV CMD  EXT ECB#128                                      000363A4
                                                 DBRC CV CMD  EXT ECB#128                                      000364C4
                                                 DBRC CV CMD  EXT ECB#128                                      00036588
                                                 DBRC CV CMD  EXT ECB#128                                      00036614
                                                 DBRC CV CMD  EXT ECB#128                                      000365E0
UCFRDLN  13        *LD*                   0      PLE SERVICE  UCFRDLN         RHDCMSTR 0         *DC*          0006154C
                                                 ERUS REQUEST RHDCUCFZ                                         177633C8
VTAMRDLN 14        *LD*                   198171 PLE SERVICE  VTAMRDLN        RHDCMSTR 0         *DC*          0006164C
                                                 VTAM READ    EXT ECB#154                                      0065F6D4
TCPIPRD  15        *LD*                   95     PLE SERVICE  TCPIPRD         RHDCMSTR 0         *DC*          0006174C
                                                 INT ECB#55                                                    000617C4
                                                 EXT ECB#209                                                   1E789ACC
                                                 EXT ECB#209                                                   1E81E94C
SYSINORD 16        *LD*                   266698 PLE SERVICE  SYSINORD        RHDCMSTR 0         *DC*          000618CC
```

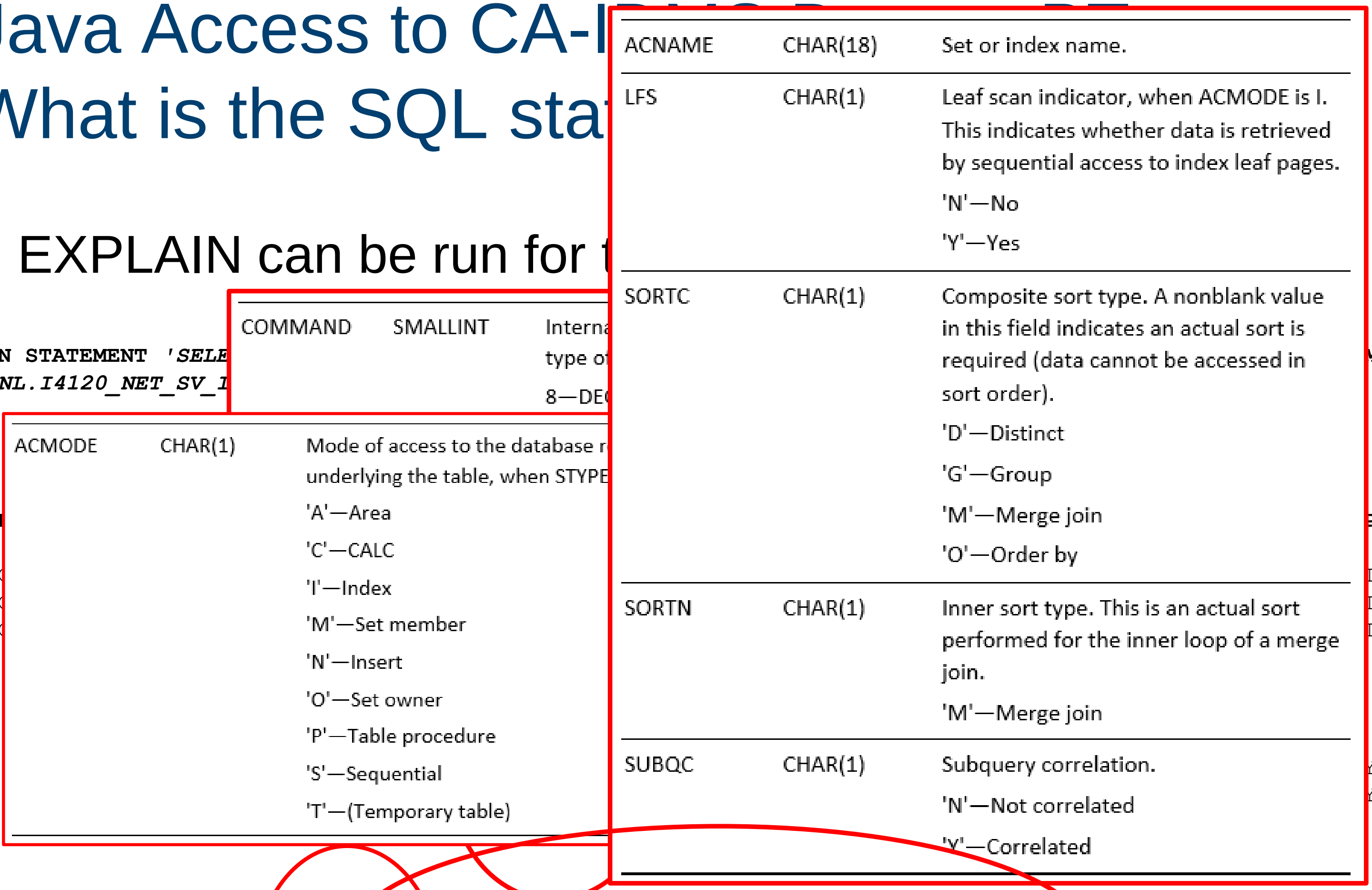
Java Access to CA-IDMS Data at BT
Lessons Learnt – User-ID Expiring

- CV uses external signon security - RACF
- JCSS connects to IDMS using a dedicated userid/password that is defined to RACF.
- We have multiple connections into the CV and the userid used is *always* active
- If CA-IDMS already has the signon control block for a userid it does not go back out to RACF to revalidate the userid/password
- This resulted in the userid/password being marked as inactive in RACF and getting revoked.
- Issue only comes to light when the next signon is attempted e.g. when CV is next cycled.

Java Access to CA-I
What is the SQL sta

- EXPLAIN can be run for t

```
EXPLAIN STATEMENT 'SELE                                                         M
WHERE NL.I4120_NET_SV_I

SELECT

 DBNAM                                                                          PROGRAM    PVERSION

 SQLDI                                                                          IDMSEXPL          0
 SQLDI                                                                          IDMSEXPL          0
 SQLDI                                                                          IDMSEXPL          0

 PDICT                                                                                    TABLE

                                                                                Y         NSL
                                                                                Y         NSL-COMMENT

 TSTAMP                     ACMODE  ACNAME               LFS  SORTC  SORTN  SUBQC

 0001-01-01-00.00.00.000000  C
 0001-01-01-00.00.00.000000  M      S-NSL-NSLCOMM        N
 2016-09-20-10.51.19.117245
```



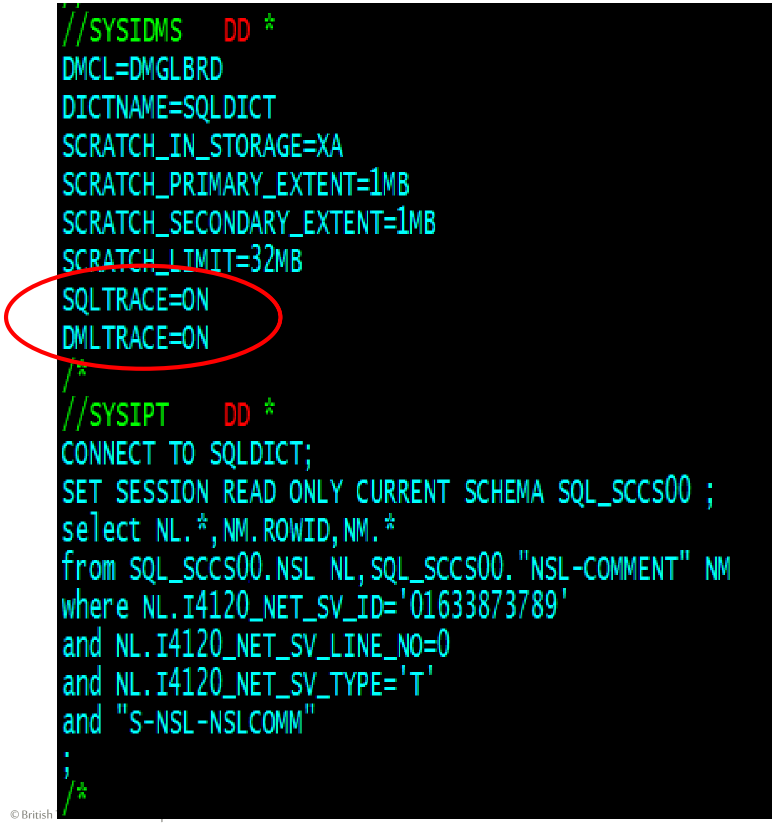
COMMAND	SMALLINT	Interna type o 8—DE

ACMODE	CHAR(1)	Mode of access to the database r underlying the table, when STYPE 'A'—Area 'C'—CALC 'I'—Index 'M'—Set member 'N'—Insert 'O'—Set owner 'P'—Table procedure 'S'—Sequential 'T'—(Temporary table)

ACNAME	CHAR(18)	Set or index name.
LFS	CHAR(1)	Leaf scan indicator, when ACMODE is I. This indicates whether data is retrieved by sequential access to index leaf pages. 'N'—No 'Y'—Yes
SORTC	CHAR(1)	Composite sort type. A nonblank value in this field indicates an actual sort is required (data cannot be accessed in sort order). 'D'—Distinct 'G'—Group 'M'—Merge join 'O'—Order by
SORTN	CHAR(1)	Inner sort type. This is an actual sort performed for the inner loop of a merge join. 'M'—Merge join
SUBQC	CHAR(1)	Subquery correlation. 'N'—Not correlated 'Y'—Correlated

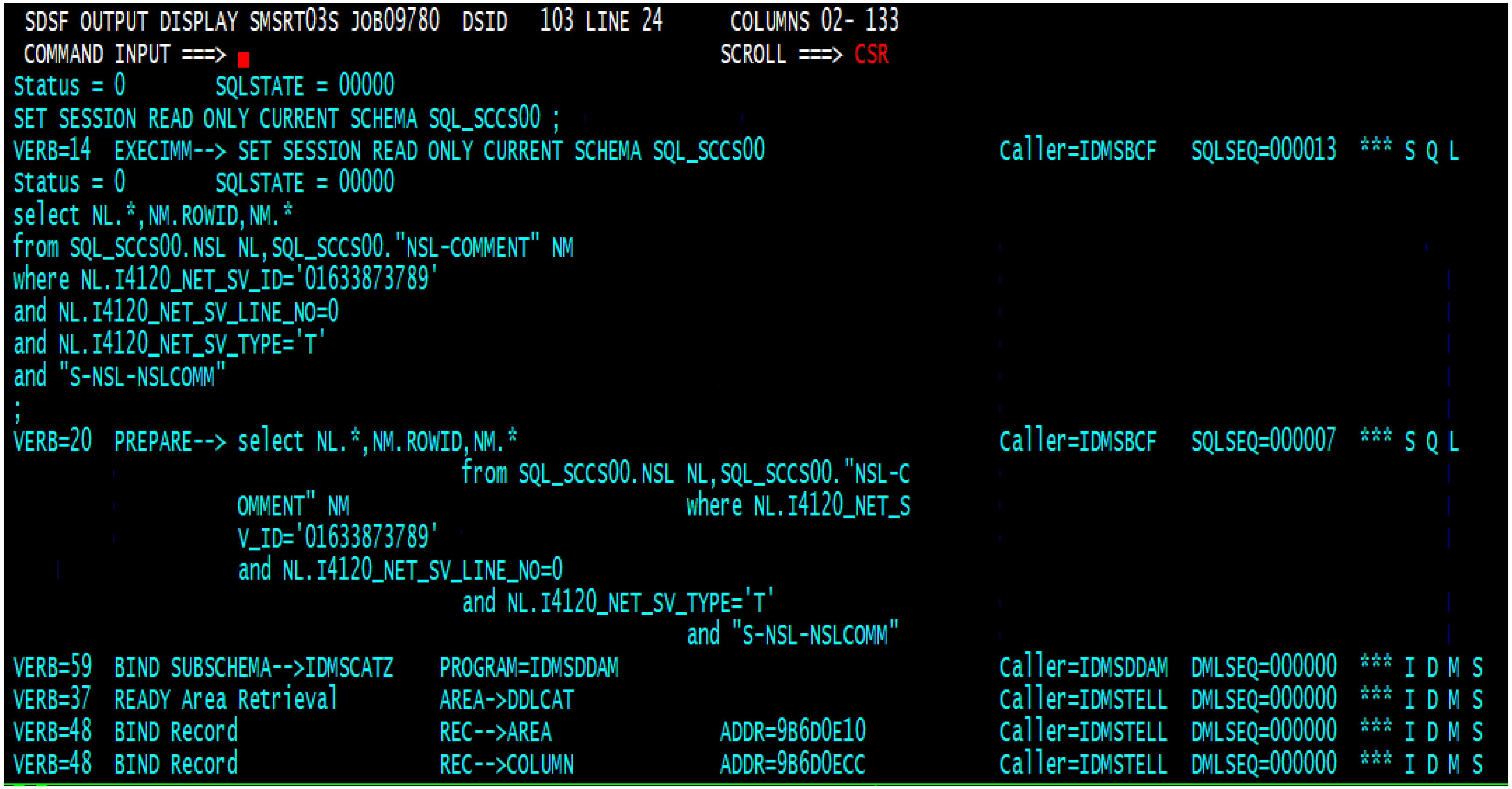
Java Access to CA-IDMS Data at BT
What is the SQL statement actually doing?

- Quick way to determine the DML that an SQL statement will drive
- Run a local mode IDMSBCF job using DMLTRACE and SQLTRACE



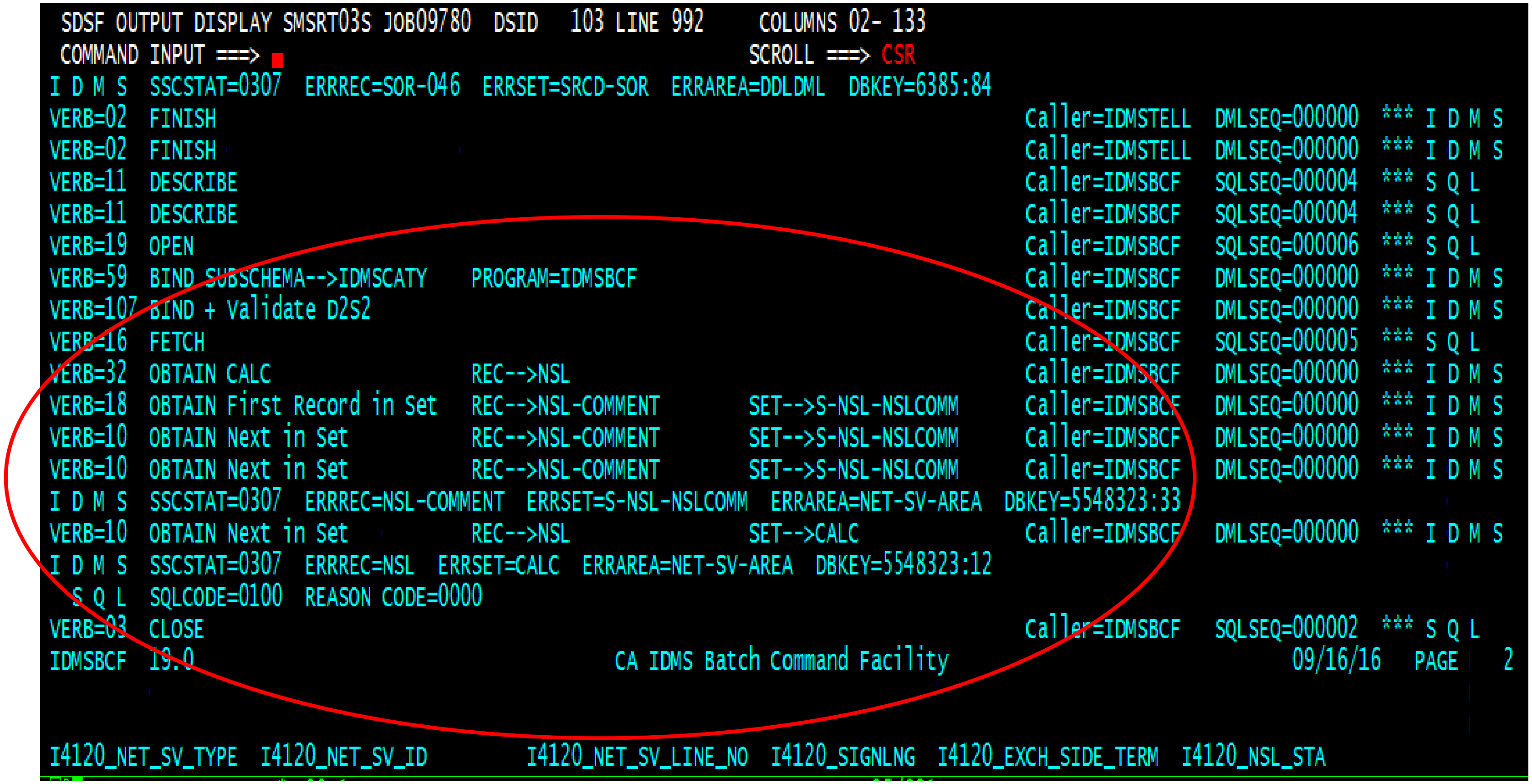
```
//SYSIDMS   DD *
DMCL=DMGLBRD
DICTNAME=SQLDICT
SCRATCH_IN_STORAGE=XA
SCRATCH_PRIMARY_EXTENT=1MB
SCRATCH_SECONDARY_EXTENT=1MB
SCRATCH_LIMIT=32MB
SQLTRACE=ON
DMLTRACE=ON
/*
//SYSIPT    DD *
CONNECT TO SQLDICT;
SET SESSION READ ONLY CURRENT SCHEMA SQL_SCCS00 ;
select NL.*,NM.ROWID,NM.*
from SQL_SCCS00.NSL NL,SQL_SCCS00."NSL-COMMENT" NM
where NL.I4120_NET_SV_ID='01633873789'
and NL.I4120_NET_SV_LINE_NO=0
and NL.I4120_NET_SV_TYPE='T'
and "S-NSL-NSLCOMM"
;
/*
```

Java Access to CA-IDMS Data at BT
What is the SQL statement actually doing?



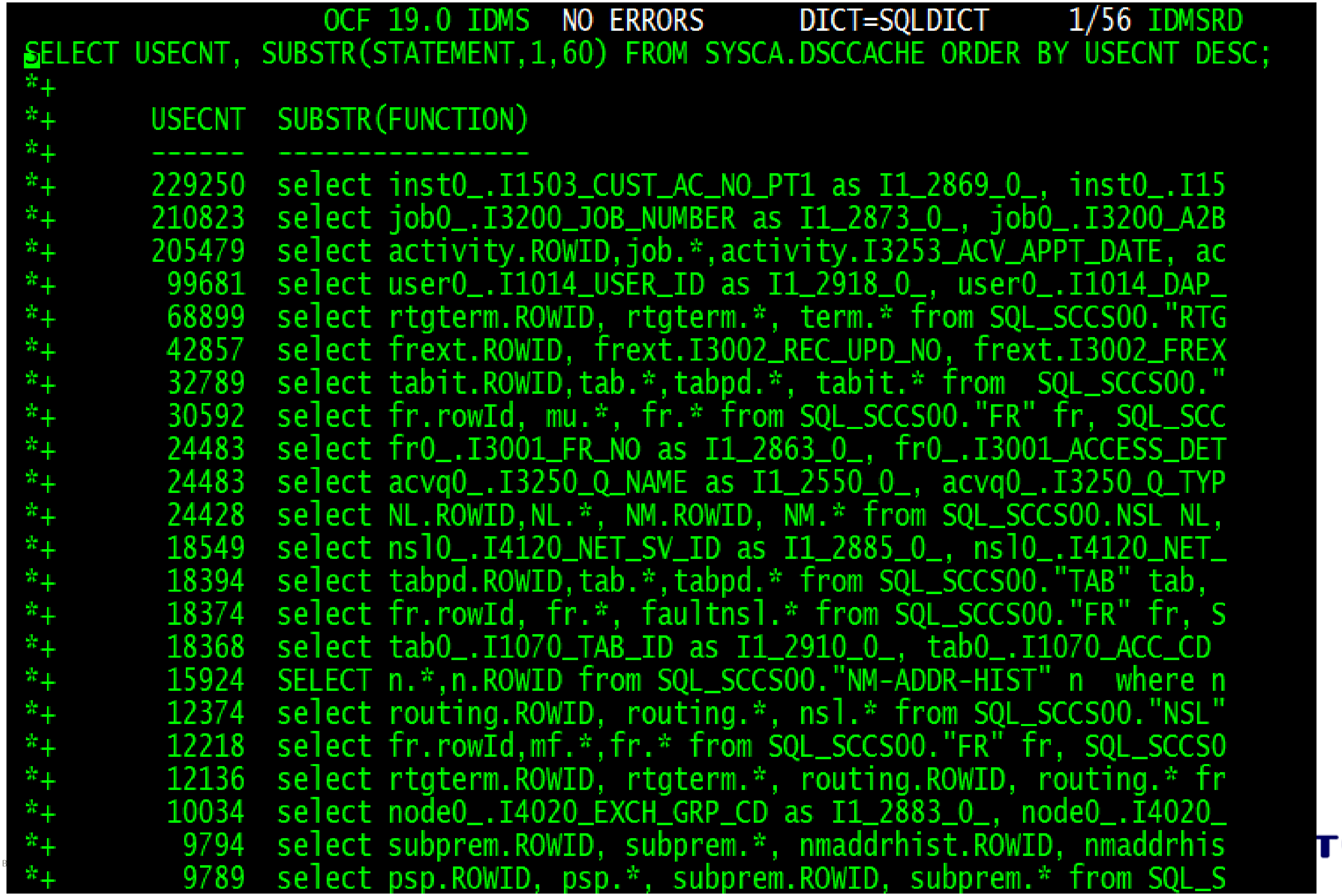
```
 SDSF OUTPUT DISPLAY SMSRT03S JOB09780  DSID   103 LINE 24      COLUMNS 02- 133
 COMMAND INPUT ===>                                            SCROLL ===> CSR
Status = 0        SQLSTATE = 00000
SET SESSION READ ONLY CURRENT SCHEMA SQL_SCCS00 ;
VERB=14  EXECIMM--> SET SESSION READ ONLY CURRENT SCHEMA SQL_SCCS00                     Caller=IDMSBCF   SQLSEQ=000013  *** S Q L
Status = 0        SQLSTATE = 00000
select NL.*,NM.ROWID,NM.*
from SQL_SCCS00.NSL NL,SQL_SCCS00."NSL-COMMENT" NM
where NL.I4120_NET_SV_ID='01633873789'
and NL.I4120_NET_SV_LINE_NO=0
and NL.I4120_NET_SV_TYPE='T'
and "S-NSL-NSLCOMM"
;
VERB=20  PREPARE--> select NL.*,NM.ROWID,NM.*                                           Caller=IDMSBCF   SQLSEQ=000007  *** S Q L
                                        from SQL_SCCS00.NSL NL,SQL_SCCS00."NSL-C
                    OMMENT" NM                              where NL.I4120_NET_S
                    V_ID='01633873789'
                    and NL.I4120_NET_SV_LINE_NO=0
                                        and NL.I4120_NET_SV_TYPE='T'
                                                            and "S-NSL-NSLCOMM"
VERB=59  BIND SUBSCHEMA-->IDMSCATZ    PROGRAM=IDMSDDAM                                  Caller=IDMSDDAM  DMLSEQ=000000  *** I D M S
VERB=37  READY Area Retrieval         AREA->DDLCAT                                      Caller=IDMSTELL  DMLSEQ=000000  *** I D M S
VERB=48  BIND Record                  REC-->AREA               ADDR=9B6D0E10            Caller=IDMSTELL  DMLSEQ=000000  *** I D M S
VERB=48  BIND Record                  REC-->COLUMN             ADDR=9B6D0ECC            Caller=IDMSTELL  DMLSEQ=000000  *** I D M S
```

Java Access to CA-IDMS Data at BT
What is the SQL statement actually doing?



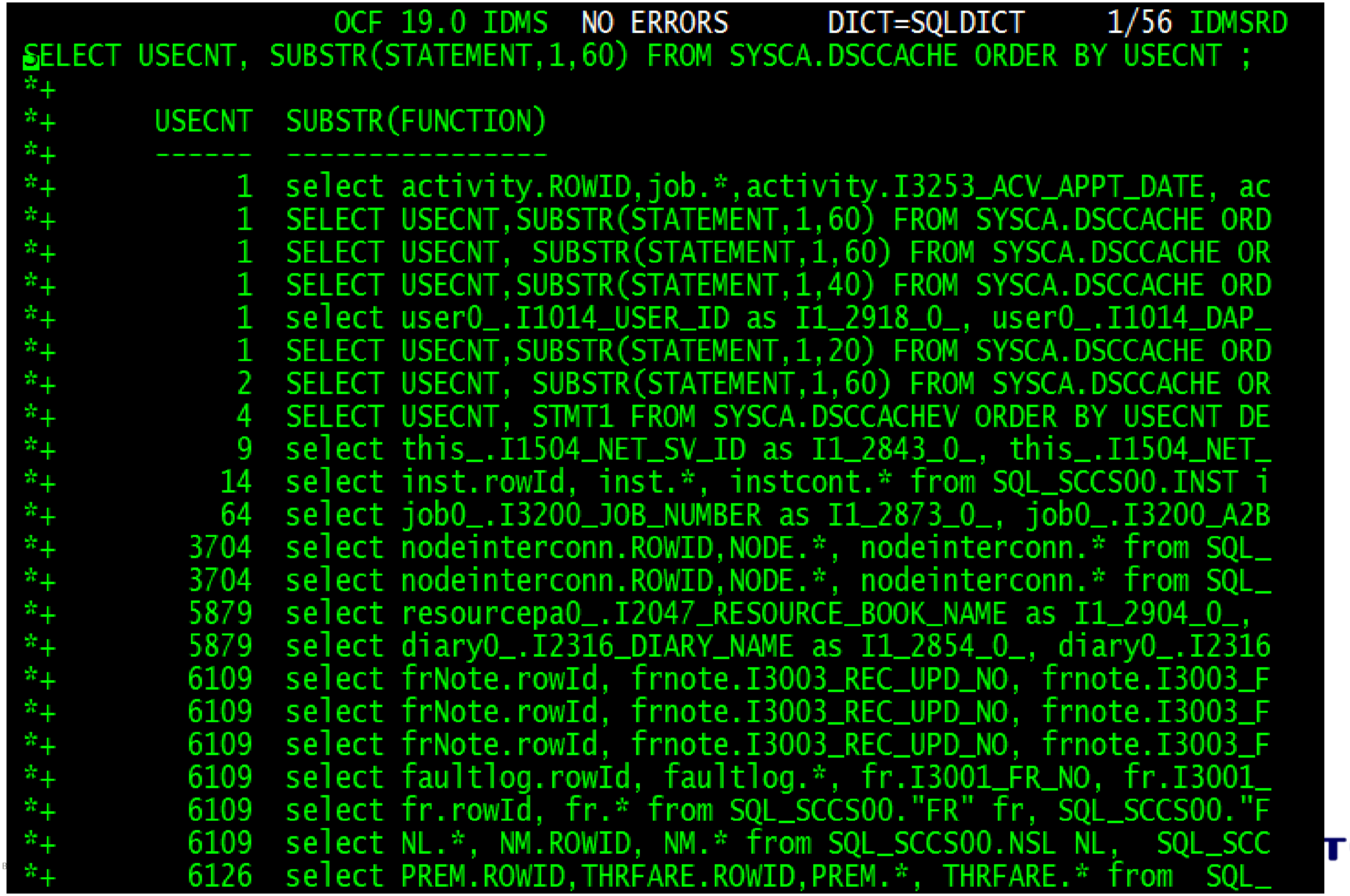
```
 SDSF OUTPUT DISPLAY SMSRT03S JOB09780  DSID   103 LINE 992      COLUMNS 02- 133
 COMMAND INPUT ===>                                         SCROLL ===> CSR
I D M S  SSCSTAT=0307  ERRREC=SOR-046  ERRSET=SRCD-SOR  ERRAREA=DDLDML  DBKEY=6385:84
VERB=02  FINISH                                                                  Caller=IDMSTELL  DMLSEQ=000000  *** I D M S
VERB=02  FINISH                                                                  Caller=IDMSTELL  DMLSEQ=000000  *** I D M S
VERB=11  DESCRIBE                                                                Caller=IDMSBCF   SQLSEQ=000004  *** S Q L
VERB=11  DESCRIBE                                                                Caller=IDMSBCF   SQLSEQ=000004  *** S Q L
VERB=19  OPEN                                                                    Caller=IDMSBCF   SQLSEQ=000006  *** S Q L
VERB=59  BIND SUBSCHEMA-->IDMSCATY    PROGRAM=IDMSBCF                            Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
VERB=107 BIND + Validate D2S2                                                    Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
VERB=16  FETCH                                                                   Caller=IDMSBCF   SQLSEQ=000005  *** S Q L
VERB=32  OBTAIN CALC                 REC-->NSL                                   Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
VERB=18  OBTAIN First Record in Set  REC-->NSL-COMMENT        SET-->S-NSL-NSLCOMM  Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
VERB=10  OBTAIN Next in Set          REC-->NSL-COMMENT        SET-->S-NSL-NSLCOMM  Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
VERB=10  OBTAIN Next in Set          REC-->NSL-COMMENT        SET-->S-NSL-NSLCOMM  Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
I D M S  SSCSTAT=0307  ERRREC=NSL-COMMENT  ERRSET=S-NSL-NSLCOMM  ERRAREA=NET-SV-AREA  DBKEY=5548323:33
VERB=10  OBTAIN Next in Set          REC-->NSL                SET-->CALC           Caller=IDMSBCF   DMLSEQ=000000  *** I D M S
I D M S  SSCSTAT=0307  ERRREC=NSL  ERRSET=CALC  ERRAREA=NET-SV-AREA  DBKEY=5548323:12
  S Q L  SQLCODE=0100  REASON CODE=0000
VERB=03  CLOSE                                                                   Caller=IDMSBCF   SQLSEQ=000002  *** S Q L
IDMSBCF  19.0                              CA IDMS Batch Command Facility                    09/16/16    PAGE    2



I4120_NET_SV_TYPE  I4120_NET_SV_ID       I4120_NET_SV_LINE_NO  I4120_SIGNLNG  I4120_EXCH_SIDE_TERM  I4120_NSL_STA
```

Java Access to CA-IDMS Data at BT
Ensure you have a big enough SQLCACHE



```
                  OCF 19.0 IDMS  NO ERRORS      DICT=SQLDICT     1/56 IDMSRD
SELECT USECNT, SUBSTR(STATEMENT,1,60) FROM SYSCA.DSCCACHE ORDER BY USECNT DESC;
*+
*+       USECNT  SUBSTR(FUNCTION)
*+       ------  ----------------
*+       229250  select inst0_.I1503_CUST_AC_NO_PT1 as I1_2869_0_, inst0_.I15
*+       210823  select job0_.I3200_JOB_NUMBER as I1_2873_0_, job0_.I3200_A2B
*+       205479  select activity.ROWID,job.*,activity.I3253_ACV_APPT_DATE, ac
*+        99681  select user0_.I1014_USER_ID as I1_2918_0_, user0_.I1014_DAP_
*+        68899  select rtgterm.ROWID, rtgterm.*, term.* from SQL_SCCS00."RTG
*+        42857  select frext.ROWID, frext.I3002_REC_UPD_NO, frext.I3002_FREX
*+        32789  select tabit.ROWID,tab.*,tabpd.*, tabit.* from  SQL_SCCS00."
*+        30592  select fr.rowId, mu.*, fr.* from SQL_SCCS00."FR" fr, SQL_SCC
*+        24483  select fr0_.I3001_FR_NO as I1_2863_0_, fr0_.I3001_ACCESS_DET
*+        24483  select acvq0_.I3250_Q_NAME as I1_2550_0_, acvq0_.I3250_Q_TYP
*+        24428  select NL.ROWID,NL.*, NM.ROWID, NM.* from SQL_SCCS00.NSL NL,
*+        18549  select nsl0_.I4120_NET_SV_ID as I1_2885_0_, nsl0_.I4120_NET_
*+        18394  select tabpd.ROWID,tab.*,tabpd.* from SQL_SCCS00."TAB" tab,
*+        18374  select fr.rowId, fr.*, faultnsl.* from SQL_SCCS00."FR" fr, S
*+        18368  select tab0_.I1070_TAB_ID as I1_2910_0_, tab0_.I1070_ACC_CD
*+        15924  SELECT n.*,n.ROWID from SQL_SCCS00."NM-ADDR-HIST" n  where n
*+        12374  select routing.ROWID, routing.*, nsl.* from SQL_SCCS00."NSL"
*+        12218  select fr.rowId,mf.*,fr.* from SQL_SCCS00."FR" fr, SQL_SCCS0
*+        12136  select rtgterm.ROWID, rtgterm.*, routing.ROWID, routing.* fr
*+        10034  select node0_.I4020_EXCH_GRP_CD as I1_2883_0_, node0_.I4020_
*+         9794  select subprem.ROWID, subprem.*, nmaddrhist.ROWID, nmaddrhis
*+         9789  select psp.ROWID, psp.*, subprem.ROWID, subprem.* from SQL_S
```

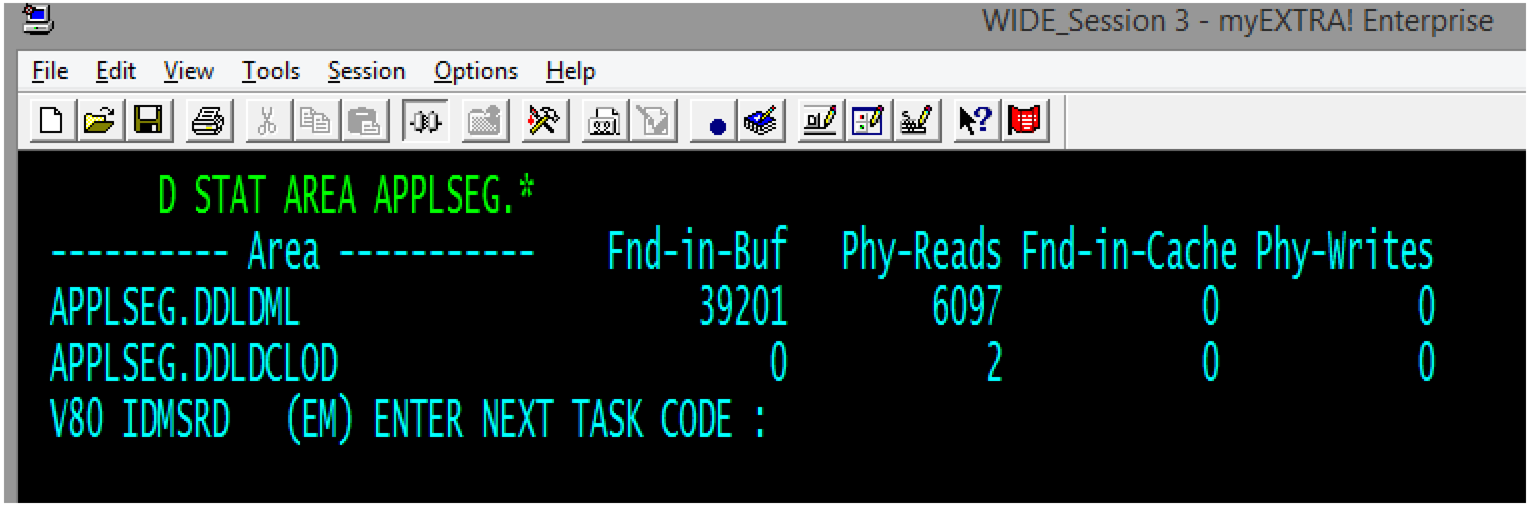
Java Access to CA-IDMS Data at BT
Ensure you have a big enough SQLCACHE



```
                   OCF 19.0 IDMS  NO ERRORS      DICT=SQLDICT     1/56 IDMSRD
SELECT USECNT, SUBSTR(STATEMENT,1,60) FROM SYSCA.DSCCACHE ORDER BY USECNT ;
*+
*+      USECNT  SUBSTR(FUNCTION)
*+      ------  ----------------
*+           1  select activity.ROWID,job.*,activity.I3253_ACV_APPT_DATE, ac
*+           1  SELECT USECNT,SUBSTR(STATEMENT,1,60) FROM SYSCA.DSCCACHE ORD
*+           1  SELECT USECNT, SUBSTR(STATEMENT,1,60) FROM SYSCA.DSCCACHE OR
*+           1  SELECT USECNT,SUBSTR(STATEMENT,1,40) FROM SYSCA.DSCCACHE ORD
*+           1  select user0_.I1014_USER_ID as I1_2918_0_, user0_.I1014_DAP_
*+           1  SELECT USECNT,SUBSTR(STATEMENT,1,20) FROM SYSCA.DSCCACHE ORD
*+           2  SELECT USECNT, SUBSTR(STATEMENT,1,60) FROM SYSCA.DSCCACHE OR
*+           4  SELECT USECNT, STMT1 FROM SYSCA.DSCCACHEV ORDER BY USECNT DE
*+           9  select this_.I1504_NET_SV_ID as I1_2843_0_, this_.I1504_NET_
*+          14  select inst.rowId, inst.*, instcont.* from SQL_SCCS00.INST i
*+          64  select job0_.I3200_JOB_NUMBER as I1_2873_0_, job0_.I3200_A2B
*+        3704  select nodeinterconn.ROWID,NODE.*, nodeinterconn.* from SQL_
*+        3704  select nodeinterconn.ROWID,NODE.*, nodeinterconn.* from SQL_
*+        5879  select resourcepa0_.I2047_RESOURCE_BOOK_NAME as I1_2904_0_,
*+        5879  select diary0_.I2316_DIARY_NAME as I1_2854_0_, diary0_.I2316
*+        6109  select frNote.rowId, frnote.I3003_REC_UPD_NO, frnote.I3003_F
*+        6109  select frNote.rowId, frnote.I3003_REC_UPD_NO, frnote.I3003_F
*+        6109  select frNote.rowId, frnote.I3003_REC_UPD_NO, frnote.I3003_F
*+        6109  select faultlog.rowId, faultlog.*, fr.I3001_FR_NO, fr.I3001_
*+        6109  select fr.rowId, fr.* from SQL_SCCS00."FR" fr, SQL_SCCS00."F
*+        6109  select NL.*, NM.ROWID, NM.* from SQL_SCCS00.NSL NL,  SQL_SCC
*+        6126  select PREM.ROWID,THRFARE.ROWID,PREM.*, THRFARE.* from  SQL_
```

Java Access to CA-IDMS Data at BT
Ensure you have a big enough SQLCACHE

- Keep an eye on I/O for the dictionary – if this rises it could indicate that the SQL cache is not effective



WIDE_Session 3 - myEXTRA! Enterprise
File Edit View Tools Session Options Help

```
      D STAT AREA APPLSEG.*
---------- Area -----------    Fnd-in-Buf   Phy-Reads Fnd-in-Cache Phy-Writes
APPLSEG.DDLDML                      39201        6097            0          0
APPLSEG.DDLDCLOD                        0           2            0          0
V80 IDMSRD   (EM) ENTER NEXT TASK CODE :
```

Java Access to CA-IDMS Data at BT Future…..

- Proof of Concept is going well…..
- ‘CSS Transaction’ DAJ – display activity for job implemented and 100% in use by one robotic application
- Two more transactions being rolled out
- Plan is to switch additional robotic applications / clients to the new service and increase the number of CSS Transactions / Data Items served
- Longer term the use of update SQL *may* be explored

Java Access to CA-IDMS Data at BT
Questions