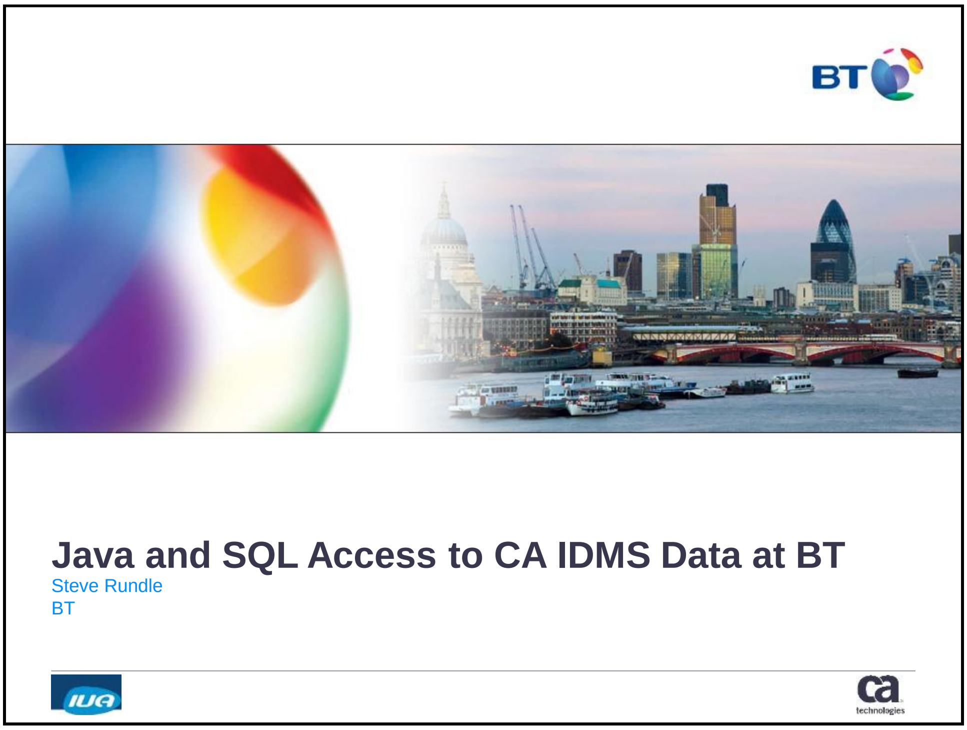
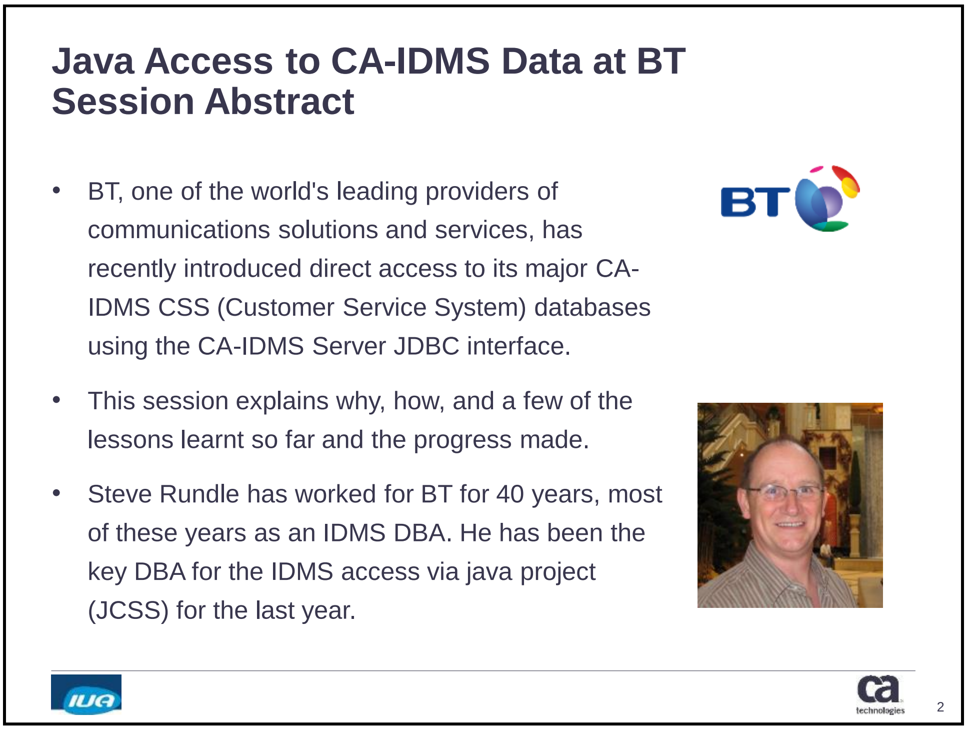


Java and SQL Access to CA IDMS Data at BT

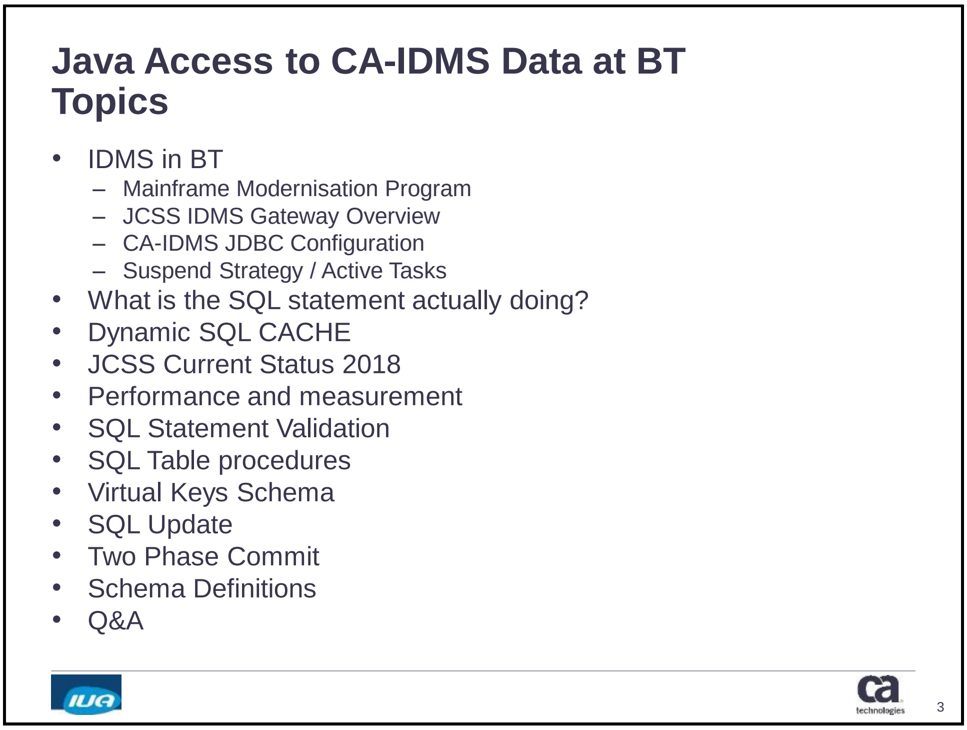
Steve Rundle
BT



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 and the progress made.
- Steve Rundle has worked for BT for 40 years, most of these years as an IDMS DBA. He has been the key DBA for the IDMS access via java project (JCSS) for the last year.

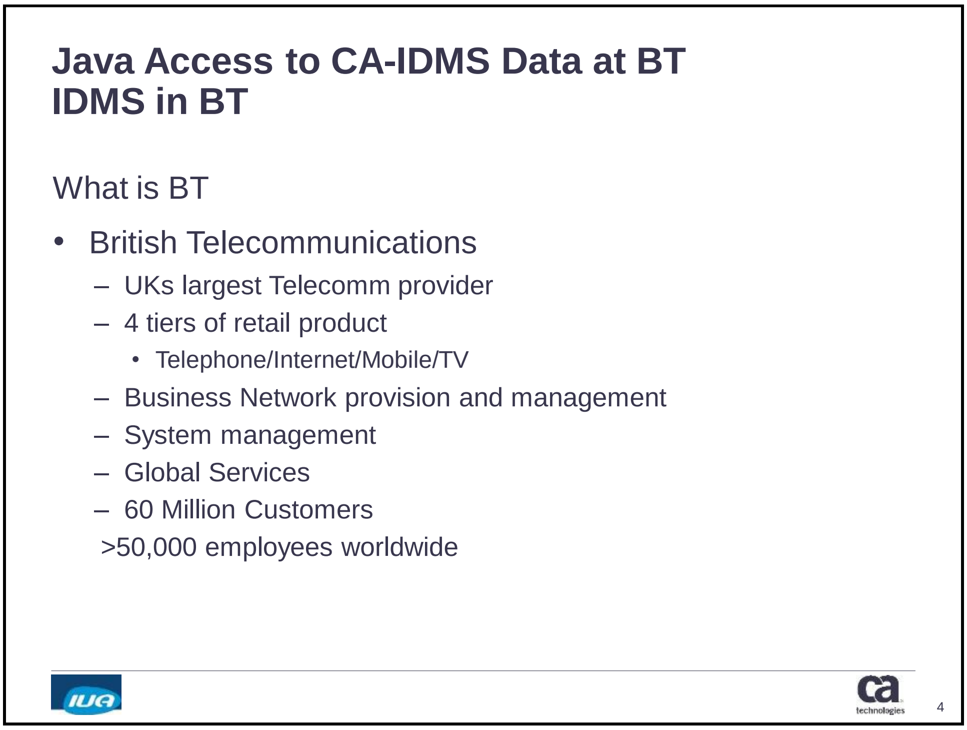
2



Java Access to CA-IDMS Data at BT
Topics

- IDMS in BT
  - Mainframe Modernisation Program
  - JCSS IDMS Gateway Overview
  - CA-IDMS JDBC Configuration
  - Suspend Strategy / Active Tasks
- What is the SQL statement actually doing?
- Dynamic SQL CACHE
- JCSS Current Status 2018
- Performance and measurement
- SQL Statement Validation
- SQL Table procedures
- Virtual Keys Schema
- SQL Update
- Two Phase Commit
- Schema Definitions
- Q&A

3



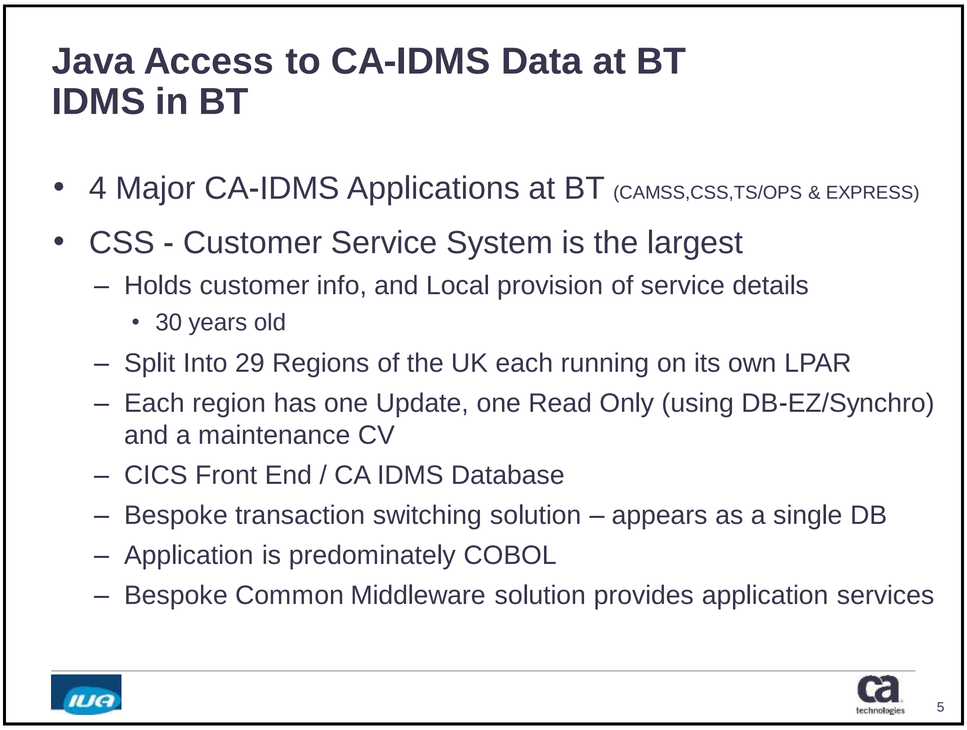
Java Access to CA-IDMS Data at BT
IDMS in BT

What is BT

- British Telecommunications
  - UKs largest Telecomm provider
  - 4 tiers of retail product
    - Telephone/Internet/Mobile/TV
  - Business Network provision and management
  - System management
  - Global Services
  - 60 Million Customers

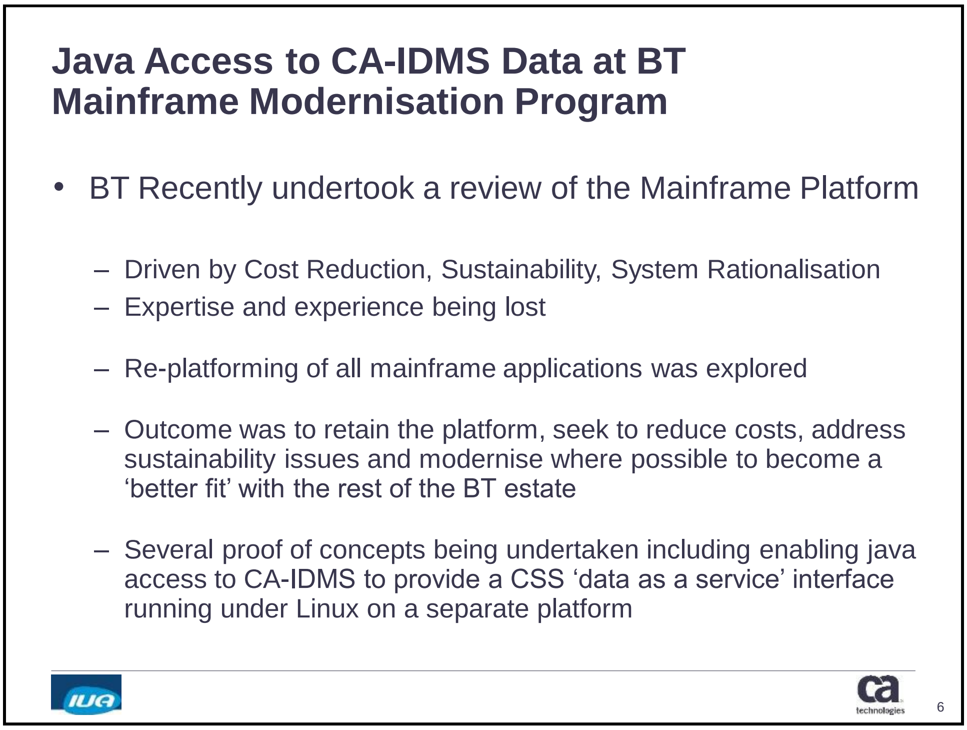
  >50,000 employees worldwide

4



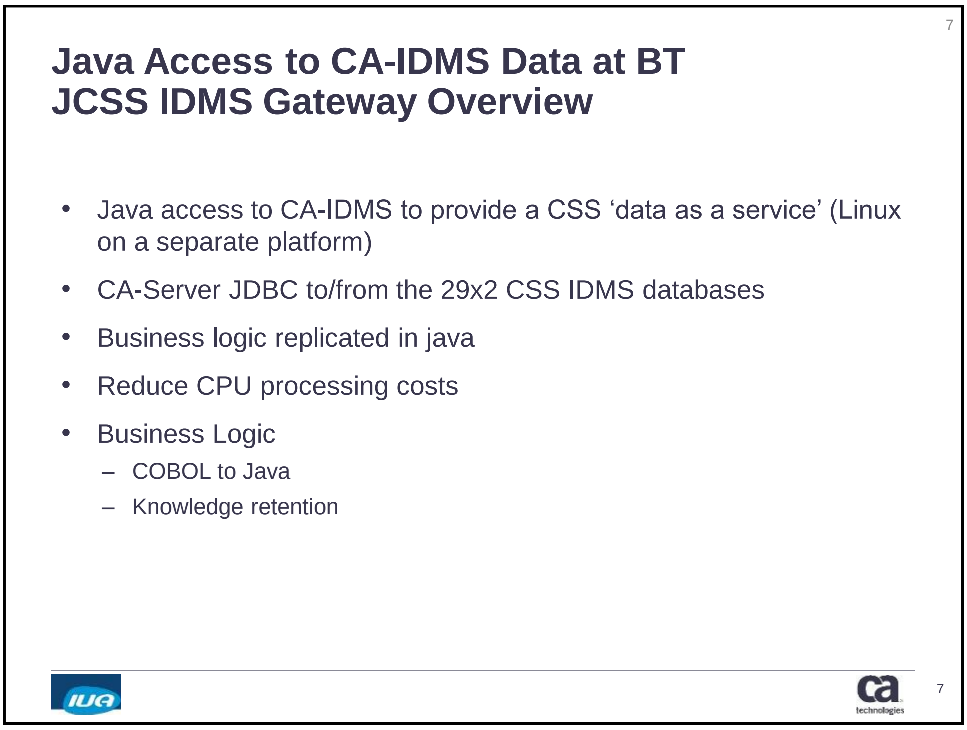
Java Access to CA-IDMS Data at BT
IDMS in BT

- 4 Major CA-IDMS Applications at BT (CAMSS,CSS,TS/OPS & EXPRESS)
- CSS - Customer Service System is the largest
  - Holds customer info, and Local provision of service details
    - 30 years old
  - 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



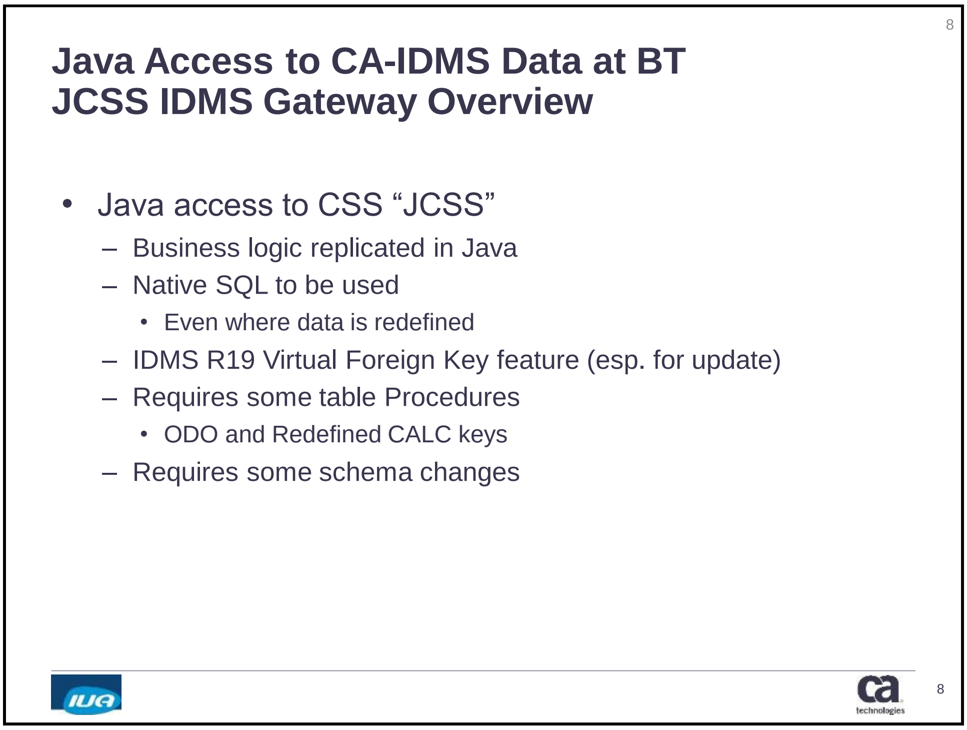
Java Access to CA-IDMS Data at BT
Mainframe Modernisation Program

- BT Recently undertook a review of the Mainframe Platform
  - Driven by Cost Reduction, Sustainability, System Rationalisation
  - Expertise and experience being lost
  - 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

- Java access to CA-IDMS to provide a CSS ‘data as a service’ (Linux on a separate platform)
- CA-Server JDBC to/from the 29x2 CSS IDMS databases
- Business logic replicated in java
- Reduce CPU processing costs
- Business Logic
  - COBOL to Java
  - Knowledge retention

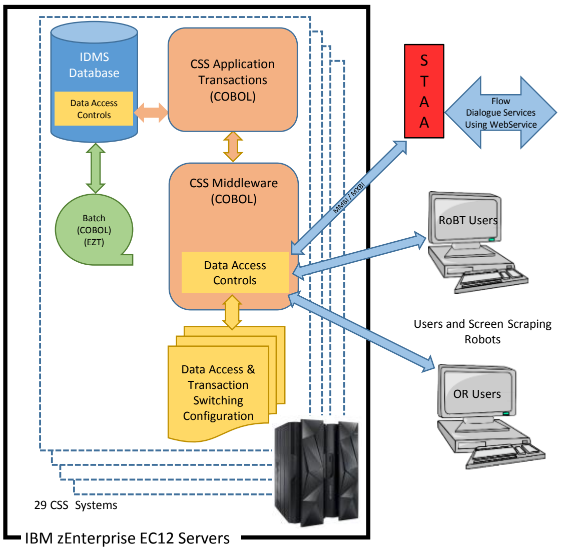


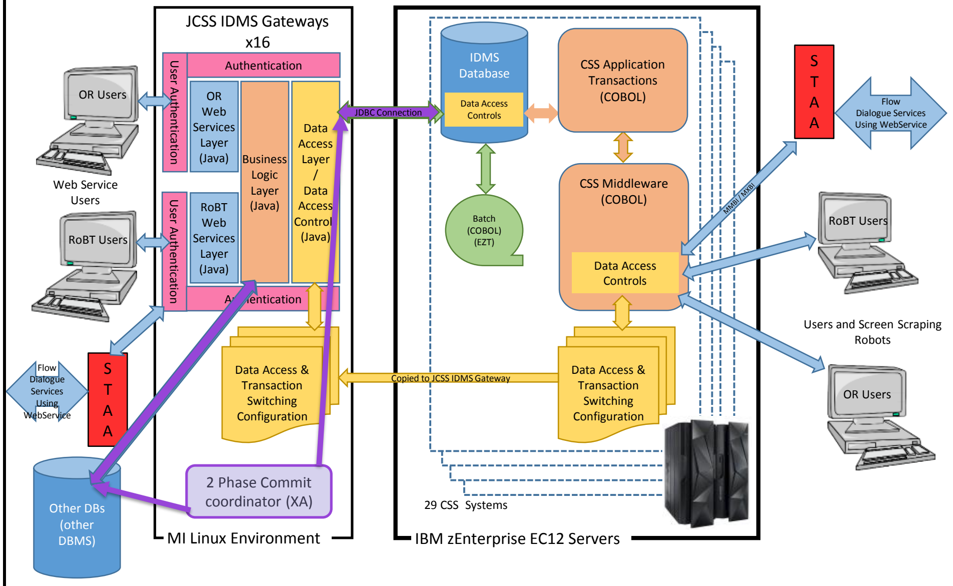
Java Access to CA-IDMS Data at BT JCSS IDMS Gateway Overview

- Java access to CSS “JCSS”
  - Business logic replicated in Java
  - Native SQL to be used
    - Even where data is redefined
  - IDMS R19 Virtual Foreign Key feature (esp. for update)
  - Requires some table Procedures
    - ODO and Redefined CALC keys
  - Requires some schema changes

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

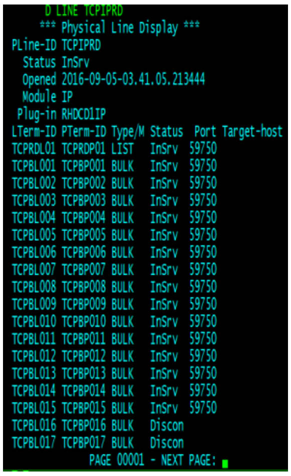
- Native SQL to be used whenever possible
  to avoid the overhead of maintaining table procedures
- We will be using the IDMS R19 Virtual Foreign Key feature
- Requires some schema changes – but not material changes
  - for example
  - FILLER redefined with alternative element name
  - Reordering redefined 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
  - Defining OM Indexes as MA.



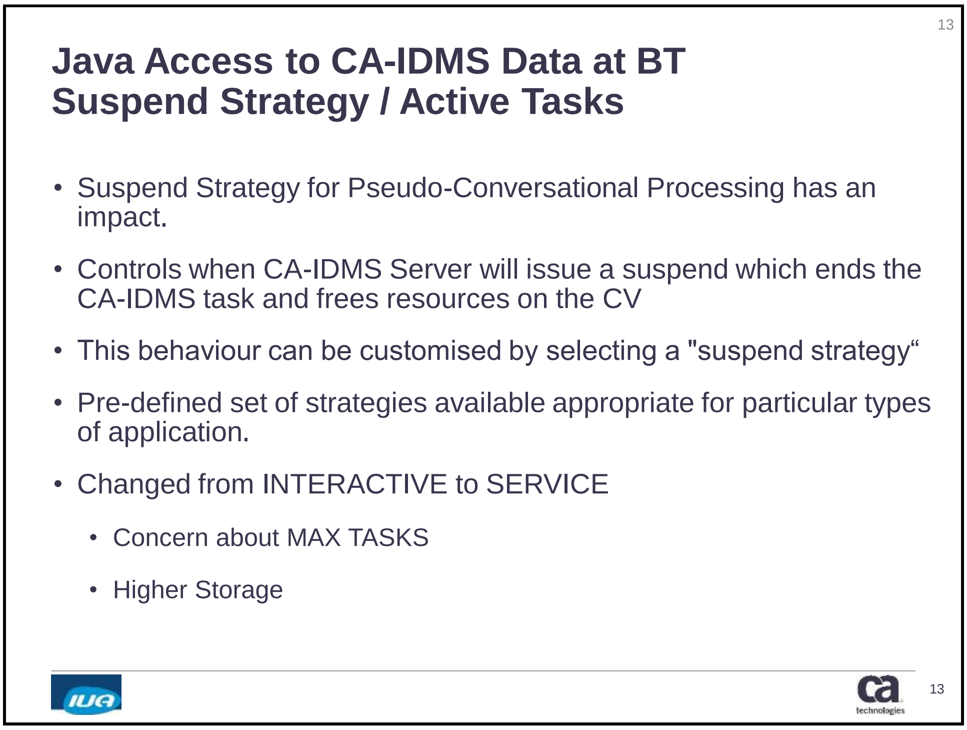

JCSS IDMS Gateways x16
Authentication
User Authentication
OR Web Services Layer (Java)
Business Logic Layer (Java)
Data Access Layer / Data Access Control (Java)
User Authentication
RoBT Web Services Layer (Java)
Authentication
OR Users
Web Service Users
RoBT Users
Flow Dialogue Services Using WebService
S T A A
Other DBs (other DBMS)
Data Access & Transaction Switching Configuration
2 Phase Commit coordinator (XA)
MI Linux Environment
JDBC Connection
Copied to JCSS IDMS Gateway
IDMS Database
Data Access Controls
CSS Application Transactions (COBOL)
Batch (COBOL) (EZT)
CSS Middleware (COBOL)
Data Access Controls
Data Access & Transaction Switching Configuration
29 CSS Systems
IBM zEnterprise EC12 Servers
S T A A
Flow Dialogue Services Using WebService
MMBI / MXBI
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 15 connections from each JCSS server to IDMS
  - *usually* 16 active connections to each CA-IDMS system
  - *could* get up to 240 active connections to each CA-IDMS system
- In CA-IDMS a TCPIP line is defined with 400 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……..

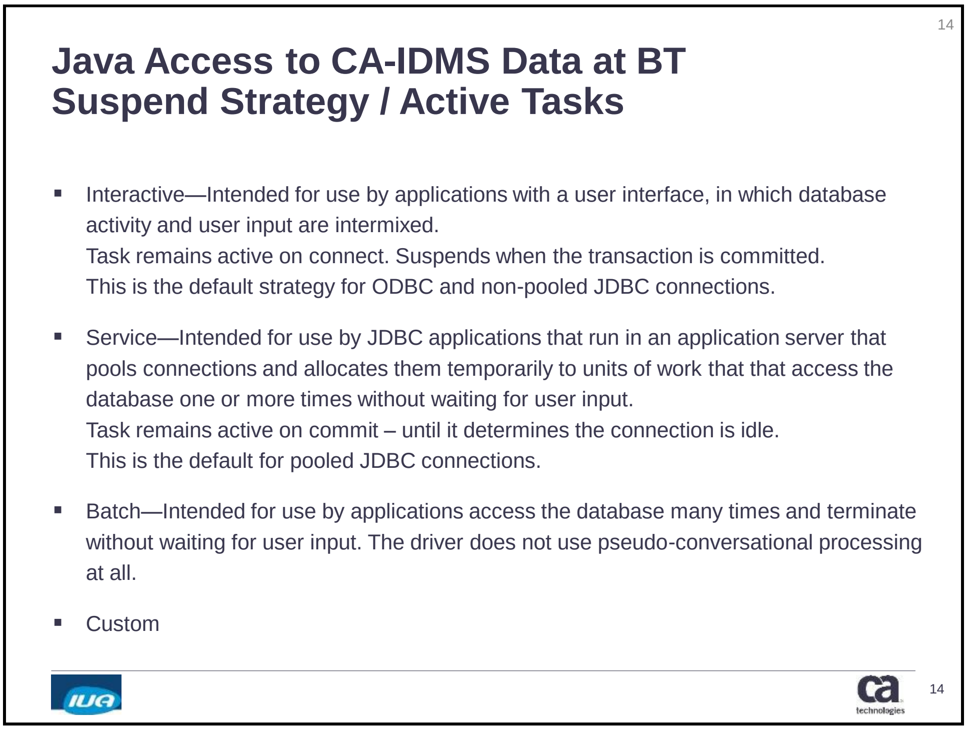


```
      D LINE TCPIPRD
     *** Physical Line Display ***
PLine-ID TCPIPRD
  Status InSrv
  Opened 2016-09-05-03.41.05.213444
  Module IP
 Plug-in RHDCD1IP
LTerm-ID PTerm-ID Type/M Status  Port Target-host
TCPRDL01 TCPRDP01 LIST   InSrv  59750
TCPBL001 TCPBP001 BULK   InSrv  59750
TCPBL002 TCPBP002 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
Suspend Strategy / 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.
- Changed from INTERACTIVE to SERVICE
  - Concern about MAX TASKS
  - Higher Storage

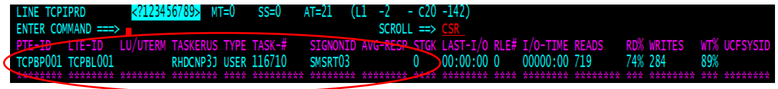


Java Access to CA-IDMS Data at BT
Suspend Strategy / Active Tasks

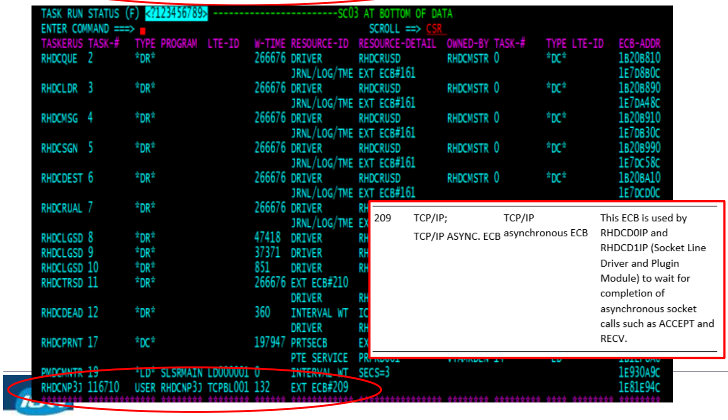
- Interactive—Intended for use by applications with a user interface, in which database activity and user input are intermixed.
  Task remains active on connect. Suspends when the transaction is committed.
  This is the default strategy for ODBC and non-pooled JDBC connections.
- Service—Intended for 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
Suspend Strategy / 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%
******** ******** ******** ******** **** ********* ******** ******** **** ******** **** ******** ******** *** ******** *** ********
```

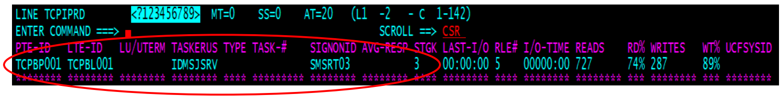


```
TASK RUN STATUS (F) <?123456789> ------------------------SC03 AT BOTTOM OF DATA
ENTER COMMAND ===>                                          SCROLL ==> CSR
TASKERUS TASK-#    TYPE PROGRAM  LTE-ID    W-TIME RESOURCE-ID RESOURCE-DETAIL OWNED-BY TASK-#    TYPE LTE-ID   ECB-ADDR
RHDCQUE  2         *DR*                    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
PWDCMNTR 19        *LD* SLSRMAIN LD000001 0       INTERVAL WT SECS=3                                           1E930A9C
RHDCNP3J 116710    USER RHDCNP3J TCPBL001 132     EXT ECB#209                                                  1E81E94C
******** ********* ******** ******** ****** ************ **************** ******** ********* **** ******** ********
```

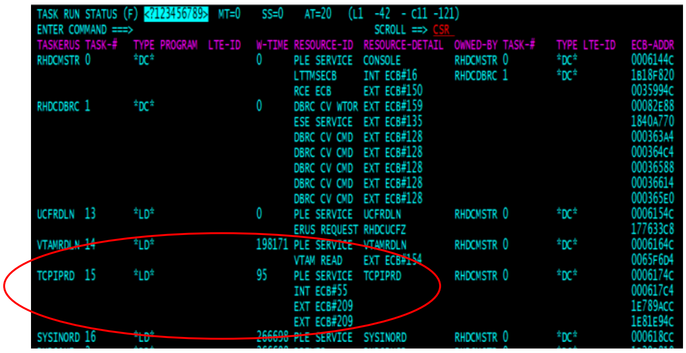
209	TCP/IP; TCP/IP ASYNC. ECB	TCP/IP asynchronous ECB	This ECB is used by RHDCD0IP and RHDCD1IP (Socket Line Driver and Plugin Module) to wait for completion of asynchronous socket calls such as ACCEPT and RECV.

Java Access to CA-IDMS Data at BT
Suspend Strategy / 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
```

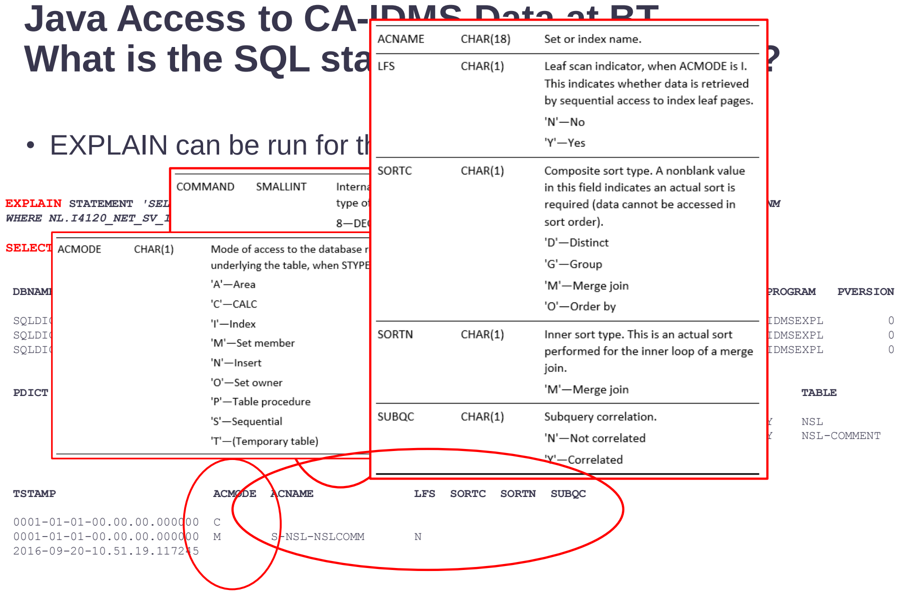
Java Access to CA-IDMS Data at BT

- What is the SQL statement doing in IDMS?
- Service affecting resource hungry SQL
- How do we know?

  Explain
  DML Trace
  Experience

Java Access to CA-IDMS Data at BT
What is the SQL sta ?

- EXPLAIN can be run for th



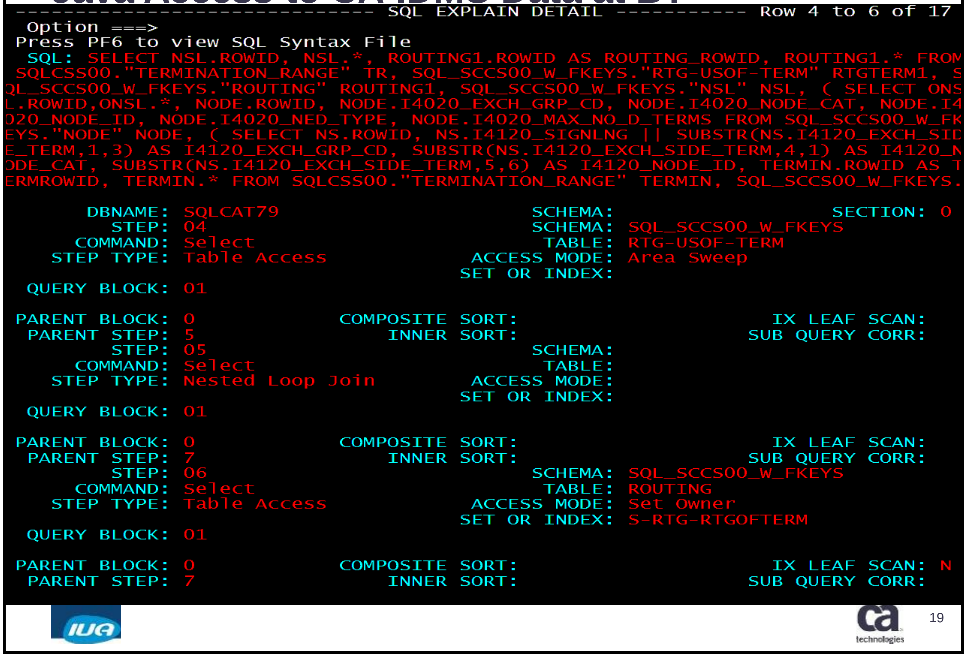
```
EXPLAIN STATEMENT 'SEL
WHERE NL.I4120_NET_SV_I                                                     NM

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

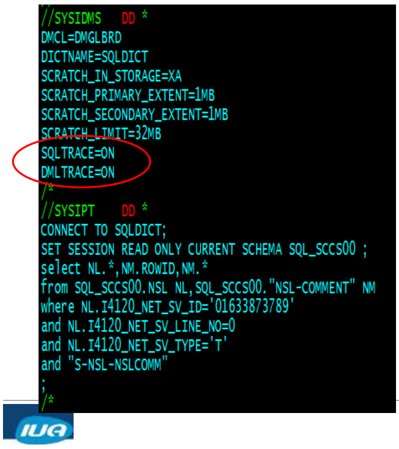
Java Access to CA-IDMS Data at BT



```
 ------------------------------ SQL EXPLAIN DETAIL ----------- Row 4 to 6 of 17
  Option ===>
 Press PF6 to view SQL Syntax File
  SQL: SELECT NSL.ROWID, NSL.*, ROUTING1.ROWID AS ROUTING_ROWID, ROUTING1.* FROM
 SQLCSS00."TERMINATION_RANGE" TR, SQL_SCCS00_W_FKEYS."RTG-USOF-TERM" RTGTERM1, S
QL_SCCS00_W_FKEYS."ROUTING" ROUTING1, SQL_SCCS00_W_FKEYS."NSL" NSL, ( SELECT ONS
L.ROWID,ONSL.*, NODE.ROWID, NODE.I4020_EXCH_GRP_CD, NODE.I4020_NODE_CAT, NODE.I4
020_NODE_ID, NODE.I4020_NED_TYPE, NODE.I4020_MAX_NO_D_TERMS FROM SQL_SCCS00_W_FK
EYS."NODE" NODE, ( SELECT NS.ROWID, NS.I4120_SIGNLNG || SUBSTR(NS.I4120_EXCH_SID
E_TERM,1,3) AS I4120_EXCH_GRP_CD, SUBSTR(NS.I4120_EXCH_SIDE_TERM,4,1) AS I4120_N
ODE_CAT, SUBSTR(NS.I4120_EXCH_SIDE_TERM,5,6) AS I4120_NODE_ID, TERMIN.ROWID AS T
ERMROWID, TERMIN.* FROM SQLCSS00."TERMINATION_RANGE" TERMIN, SQL_SCCS00_W_FKEYS.

       DBNAME: SQLCAT79                           SCHEMA:                SECTION: 0
         STEP: 04                                 SCHEMA: SQL_SCCS00_W_FKEYS
      COMMAND: Select                              TABLE: RTG-USOF-TERM
    STEP TYPE: Table Access                  ACCESS MODE: Area Sweep
                                            SET OR INDEX:
  QUERY BLOCK: 01

 PARENT BLOCK: 0             COMPOSITE SORT:                        IX LEAF SCAN:
  PARENT STEP: 5                 INNER SORT:                      SUB QUERY CORR:
         STEP: 05                                 SCHEMA:
      COMMAND: Select                              TABLE:
    STEP TYPE: Nested Loop Join              ACCESS MODE:
                                            SET OR INDEX:
  QUERY BLOCK: 01

 PARENT BLOCK: 0             COMPOSITE SORT:                        IX LEAF SCAN:
  PARENT STEP: 7                 INNER SORT:                      SUB QUERY CORR:
         STEP: 06                                 SCHEMA: SQL_SCCS00_W_FKEYS
      COMMAND: Select                              TABLE: ROUTING
    STEP TYPE: Table Access                  ACCESS MODE: Set Owner
                                            SET OR INDEX: S-RTG-RTGOFTERM
  QUERY BLOCK: 01

 PARENT BLOCK: 0             COMPOSITE SORT:                        IX LEAF SCAN: N
  PARENT STEP: 7                 INNER SORT:                      SUB QUERY CORR:
```

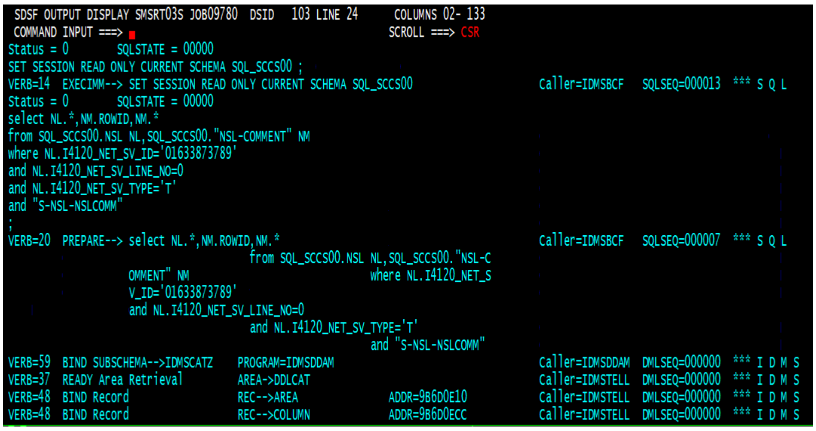
Java Access to CA-IDMS Data at BT
What is the SQL statement actually doing?

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



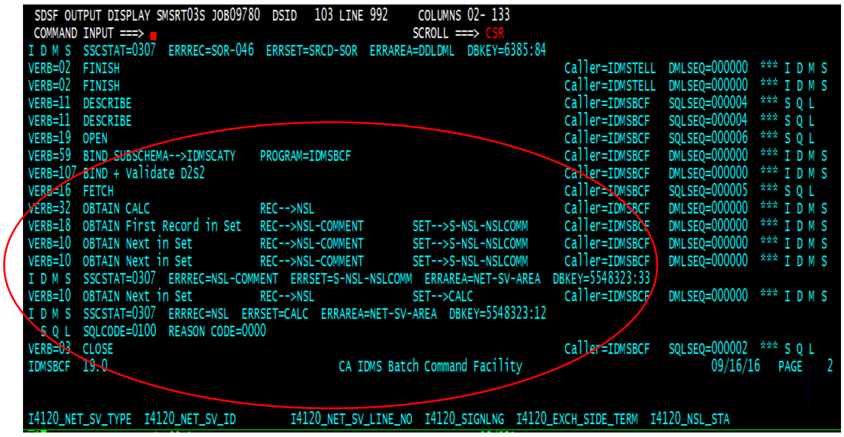
```
//SYSIDMS   DD *
DMCL=DMGLBRD
DICTNAME=SQLDICT
SCRATCH_IN_STORAGE=XA
SCRATCH_PRIMARY_EXTENT=1MB
SCRATCH_SECONDARY_EXTENT=1MB
SCRATCH_LIMIT=32MB
SQLTRACE=ON
DMLTRACE=ON
/*
//SYSIPT    DD *
CONNECT TO SQLDICT;
SET SESSION READ ONLY CURRENT SCHEMA SQL_SCCS00 ;
select NL.*,NM.ROWID,NM.*
from SQL_SCCS00.NSL 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?



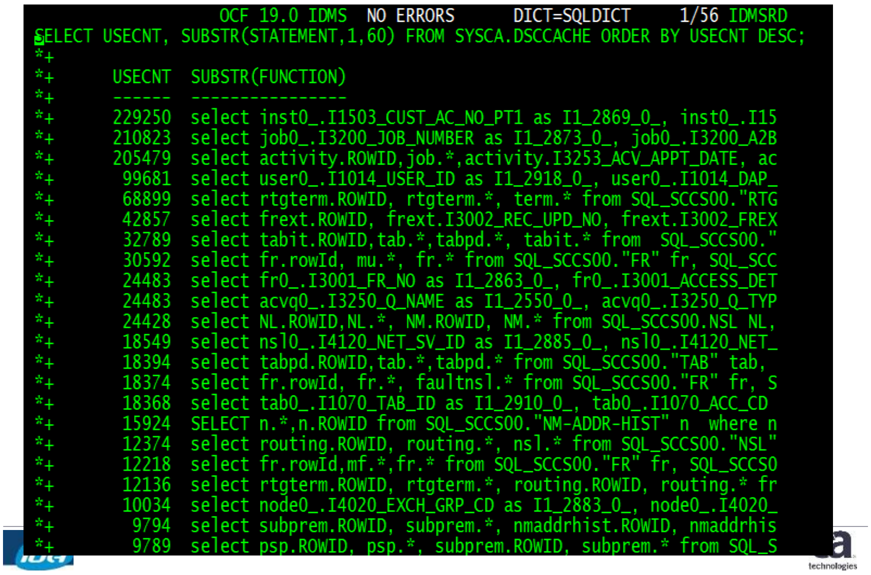
```
 SDSF OUTPUT DISPLAY SMSRT03S JOB09780  DSID   103 LINE 24      COLUMNS 02- 133
 COMMAND INPUT ===>                                            SCROLL ===> CSR
Status = 0       SQLSTATE = 00000
SET SESSION READ ONLY CURRENT SCHEMA SQL_SCCS00 ;
VERB=14  EXECIMM--> SET SESSION READ ONLY CURRENT SCHEMA 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?



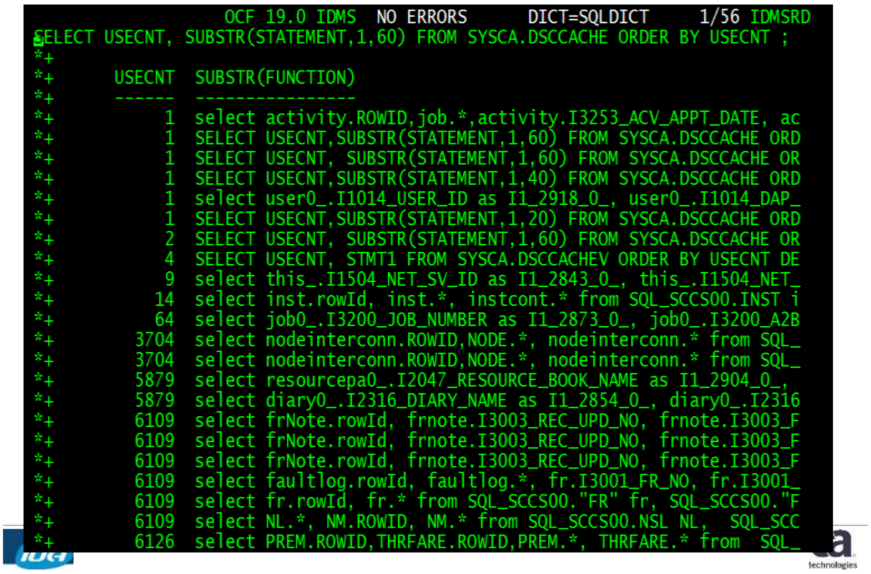
```
 SDSF OUTPUT DISPLAY SMSRT03S JOB09780  DSID   103 LINE 992     COLUMNS 02- 133
 COMMAND INPUT ===>                                            SCROLL ===> CSR
I D M S  SSCSTAT=0307  ERRREC=SOR-046  ERRSET=SRCD-SOR  ERRAREA=DDLDML  DBKEY=6385:84
VERB=02  FINISH                                                               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
Dynamic SQL CACHE



```
                  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
```

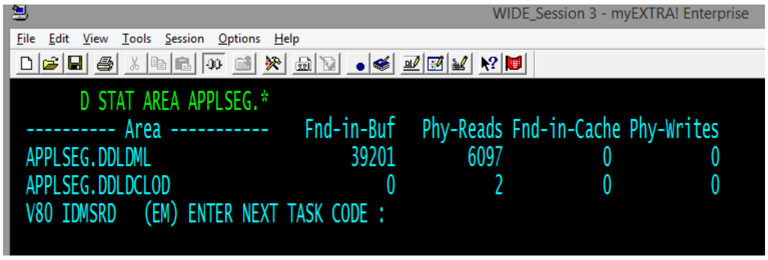
Java Access to CA-IDMS Data at BT
Dynamic SQL CACHE



```
                  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
Dynamic SQL CACHE

- 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
JCSS - Current Status 2018

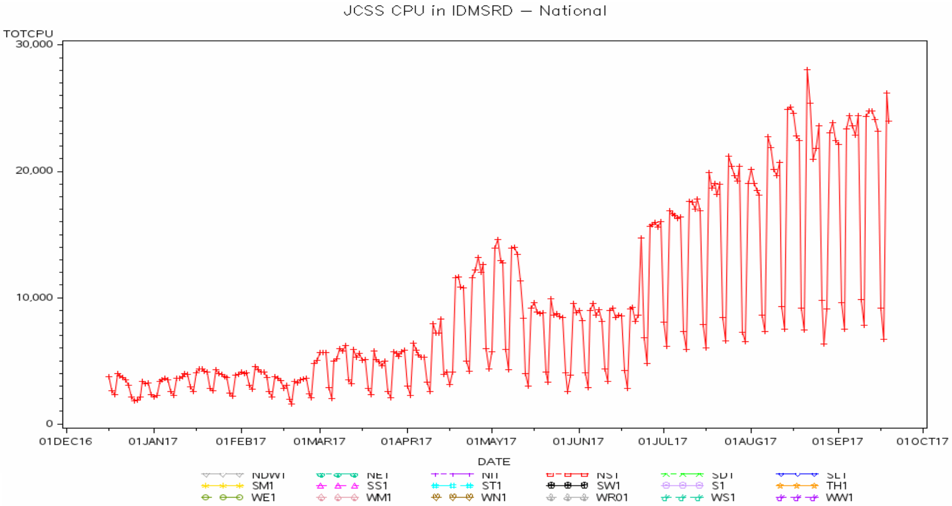
- 22 Transactions now developed and executing in production
- 37 Million Java Tasks executed per day
  (Not the same as Transactions - Approx 3 Java tasks per Xact)
- IDMS Update SQL analysed
- Monitoring and Measurement challenges
- Suspend Strategy to be changed to SERVICE
- Table Procedures developed
  Overcome IDMS SQL limitations
- 2 –phase commit possible

Java Access to CA-IDMS Data at BT Performance and Measurement

- Measured Detailed CICS transactions
  - Type/ Userid/ Usage/ Business Usage
  - We record and report on CPU/Storage/IOS for each one.
- JCSS we can only record JSRV TASKS
  - Not granular, only time and Place
- But we are looking for further details
  - JCSS Server records Transactions/type/user etc
  - We can look at SQL Cache for types of SQL queries
  - CA are looking at an API change

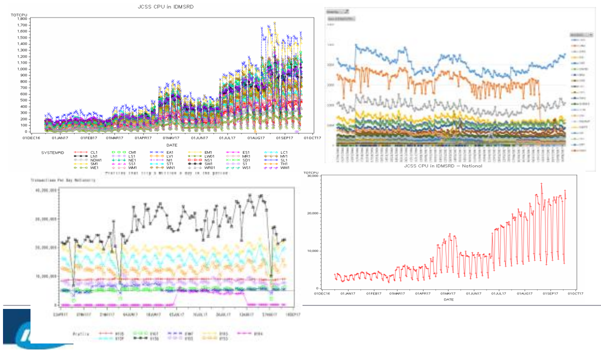
Java Access to CA-IDMS Data at BT Performance and Measurement

- Current measuring is for Transactions name and via user profiles and can be diagnosed



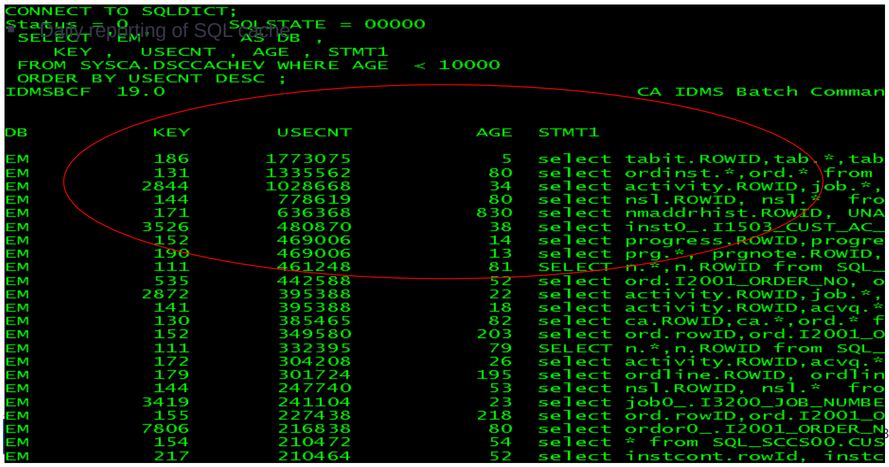
Java Access to CA-IDMS Data at BT
Performance and Measurement

- Current measuring is for Transactions name and via user profiles and can be diagnosed to individual users, But java tasks are not so easily monitored
- We are “flying blind” with only headline total Xactions and CPU usage to manage.
- We are looking to Coding the JAVA with a program name. So allow IDMS to use the program name
- SQL cache reporting is being examined.



Java Access to CA-IDMS Data at BT
Performance and Measurement
SQL Cache

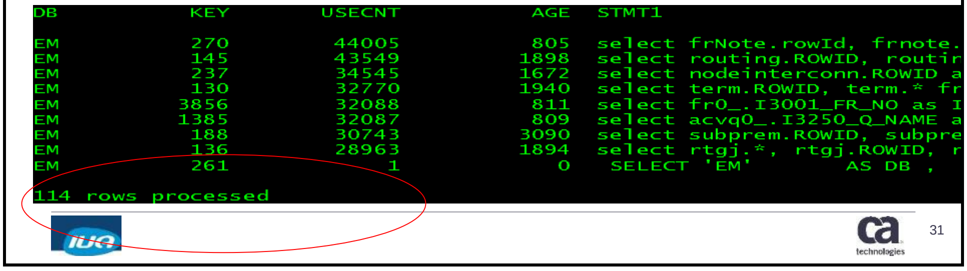
- Careful reporting on SQL Cache is possible to identify high quantity SQL queries
- SQL cache maintained against EACH of the 29 CVs.
- Daily reporting of SQL cache



```
CONNECT TO SQLDICT;
Status = 0        SQLSTATE = 00000
 SELECT 'EM'       AS DB  ,
    KEY ,  USECNT , AGE , STMT1
 FROM SYSCA.DSCCACHEV WHERE AGE  < 10000
 ORDER BY USECNT DESC ;
IDMSBCF  19.0                                              CA IDMS Batch Comman


DB           KEY        USECNT          AGE  STMT1

EM           186       1773075            5  select tabit.ROWID,tab.*,tab
EM           131       1335562           80  select ordinst.*,ord.* from
EM          2844       1028668           34  select activity.ROWID,job.*,
EM           144        778619           80  select nsl.ROWID, nsl.*  fro
EM           171        636368          830  select nmaddrhist.ROWID, UNA
EM          3526        480870           38  select inst0_.I1503_CUST_AC_
EM           152        469006           14  select progress.ROWID,progre
EM           190        469006           13  select prg.*, prgnote.ROWID,
EM           111        461248           81  SELECT n.*,n.ROWID from SQL_
EM           535        442588           52  select ord.I2001_ORDER_NO, o
EM          2872        395388           22  select activity.ROWID,job.*,
EM           141        395388           18  select activity.ROWID,acvq.*
EM           130        385465           82  select ca.ROWID,ca.*,ord.* f
EM           152        349580          203  select ord.rowID,ord.I2001_O
EM           111        332395           79  SELECT n.*,n.ROWID from SQL_
EM           172        304208           26  select activity.ROWID,acvq.*
EM           179        301724          195  select ordline.ROWID, ordlin
EM           144        247740           53  select nsl.ROWID, nsl.*  fro
EM          3419        241104           23  select job0_.I3200_JOB_NUMBE
EM           155        227438          218  select ord.rowID,ord.I2001_O
EM          7806        216838           80  select ordor0_.I2001_ORDER_N
EM           154        210472           54  select * from SQL_SCCS00.CUS
EM           217        210464           52  select instcont.rowId, instc
```

Java Access to CA-IDMS Data at BT
Performance and Measurement
SQL Cache

- Keeping an eye on these can show what SQL is being used most.
- Not easily aligned with business transactions because SQL Statements are re-used across Transactions. But still a good indicator of usage.
- We also keep an eye on the capacity (number of most recent frequently used Statements, (Selected by AGE)
- We use 200 CACHE entries. If it ever goes above this usage for frequently used, recent statements then performance will suffer.
  EG. SELECT from DSCCACHEV where age < 5000 and usecnt > 10000



```
DB            KEY        USECNT            AGE   STMT1

EM            270         44005            805   select frNote.rowId, frnote.
EM            145         43549           1898   select routing.ROWID, routir
EM            237         34545           1672   select nodeinterconn.ROWID a
EM            130         32770           1940   select term.ROWID, term.* fr
EM           3856         32088            811   select fr0_.I3001_FR_NO as I
EM           1385         32087            809   select acvq0_.I3250_Q_NAME a
EM            188         30743           3090   select subprem.ROWID, subpre
EM            136         28963           1894   select rtgj.*, rtgj.ROWID, r
EM            261             1              0    SELECT 'EM'          AS DB ,

114 rows processed
```

Java Access to CA-IDMS Data at BT
SQL Statement validation

- Validate Java SQL generated Queries for efficiency. Educate designers to IDMS
- SQL Queries are submitted to validation and authorisation
- Three processes
  - Eyeball the SQL
  - EXPLAIN
  - DML Trace

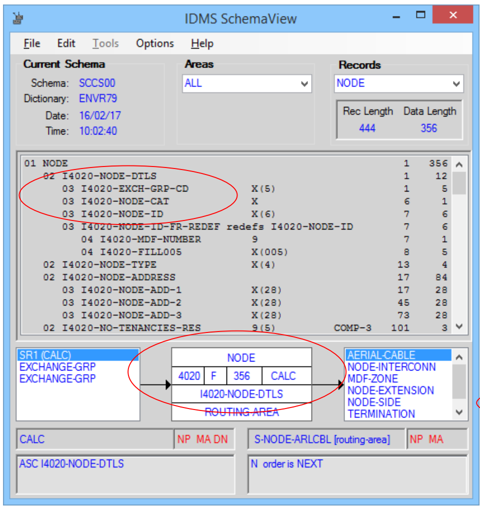
Java Access to CA-IDMS Data at BT
SQL validation - process

Currently, each TRANSACTION comprises around 20 SQL Statements.
Online system must be snappy response. DBAs validate the queries.

We receive SQL queries from designers:
EG:

- select llnk.ROWID, term.ROWID, llnk.*, term.*,node.* from SQL_SCCS00."**NODE**" node, SQL_SCCS00."**LOCAL-LINK**" llnk, SQL_SCCS00."**TERMINATION**" term where **node.I4020_EXCH_GRP_CD=? and node.I4020_NODE_CAT=? and node.I4020_NODE_ID=?** and "**S-NODE-SPR-TERM**" and "**S-TERM-UP-LL**"

Eyeball Check all the “FROM” statements for record names.
NODE, LOCAL-LINK, TERMINATION
and SETS: **S-NODE-SPR-TERM, S-TERM-UP-LL**
And where clause (Calc keys?)
Node..I4020_EXCH_GRP_CD, I4020_NODE_CAT, I4020_NODE_ID
Examine the Schema Diagram and apply experience



IDMS SchemaView

Current Schema
Schema: SCCS00
Dictionary: ENVR79
Date: 16/02/17
Time: 10:02:40

Areas: ALL

Records: NODE

Rec Length: 444
Data Length: 356

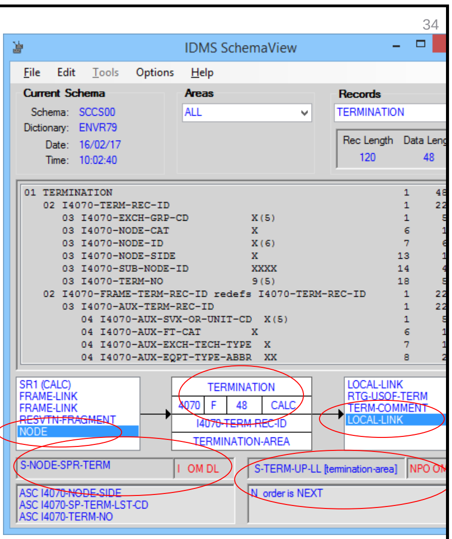
```
01 NODE                                                  1    356
  02 I4020-NODE-DTLS                                     1     12
    03 I4020-EXCH-GRP-CD          X(5)                   1      5
    03 I4020-NODE-CAT             X                      6      1
    03 I4020-NODE-ID              X(6)                   7      6
    03 I4020-NODE-ID-FR-REDEF redefs I4020-NODE-ID       7      6
      04 I4020-MDF-NUMBER         9                      7      1
      04 I4020-FILL005            X(005)                 8      5
  02 I4020-NODE-TYPE              X(4)                  13      4
  02 I4020-NODE-ADDRESS                                 17     84
    03 I4020-NODE-ADD-1           X(28)                 17     28
    03 I4020-NODE-ADD-2           X(28)                 45     28
    03 I4020-NODE-ADD-3           X(28)                 73     28
  02 I4020-NO-TENANCIES-RES       9(5)         COMP-3  101      3
```

SR1 (CALC)
EXCHANGE-GRP
EXCHANGE-GRP

NODE			
4020	F	356	CALC
I4020-NODE-DTLS			
ROUTING-AREA			

AERIAL-CABLE
NODE-INTERCONN
MDF-ZONE
NODE-EXTENSION
NODE-SIDE
TERMINATION

CALC | NP MA DN
S-NODE-ARLCBL [routing-area] | NP MA
ASC I4020-NODE-DTLS
N order is NEXT



IDMS SchemaView

Current Schema
Schema: SCCS00
Dictionary: ENVR79
Date: 16/02/17
Time: 10:02:40

Areas: ALL

Records: TERMINATION

Rec Length: 120
Data Leng: 48

```
01 TERMINATION                                                1   48
  02 I4070-TERM-REC-ID                                        1   22
    03 I4070-EXCH-GRP-CD          X(5)                        1   5
    03 I4070-NODE-CAT             X                           6   1
    03 I4070-NODE-ID              X(6)                        7   6
    03 I4070-NODE-SIDE            X                          13   1
    03 I4070-SUB-NODE-ID          XXXX                       14   4
    03 I4070-TERM-NO              9(5)                       18   5
  02 I4070-FRAME-TERM-REC-ID redefs I4070-TERM-REC-ID         1   22
    03 I4070-AUX-TERM-REC-ID                                  1   22
      04 I4070-AUX-SVX-OR-UNIT-CD  X(5)                       1   5
      04 I4070-AUX-FT-CAT          X                          6   1
      04 I4070-AUX-EXCH-TECH-TYPE  X                          7   1
      04 I4070-AUX-EQPT-TYPE-ABBR  XX                         8   2
```

SR1 (CALC)
FRAME-LINK
FRAME-LINK
RESVTN-FRAGMENT
NODE

TERMINATION			
4070	F	48	CALC
I4070-TERM-REC-ID			
TERMINATION-AREA			

LOCAL-LINK
RTG-USOF-TERM
TERM-COMMENT
LOCAL-LINK

S-NODE-SPR-TERM | I OM DL
S-TERM-UP-LL [termination-area] | NPO OM
ASC I4070-NODE-SIDE
ASC I4070-SP-TERM-LST-CD
ASC I4070-TERM-NO
N order is NEXT

select llnk.ROWID, term.ROWID, llnk.*, term.*,node.* from SQL_SCCS00."NODE" node, SQL_SCCS00."LOCAL-LINK" llnk, SQL_SCCS00."TERMINATION" term where node.I4020_EXCH_GRP_CD=? and node.I4020_NODE_CAT=? and node.I4020_NODE_ID=? and "S-NODE-SPR-TERM" and "S-TERM-UP-LL"

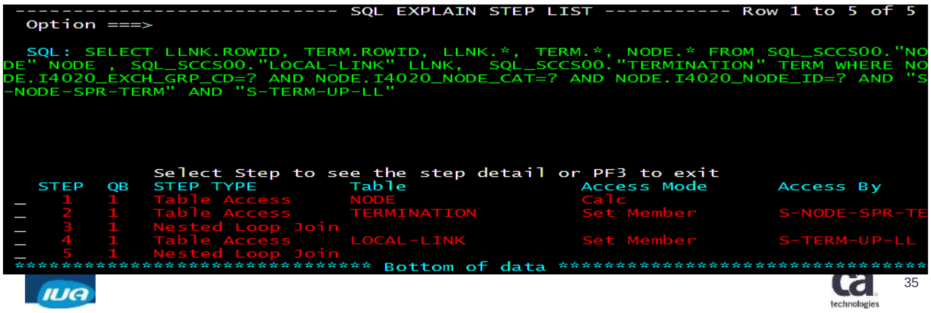
Java Access to CA-IDMS Data at BT
SQL validation - process

select llnk.ROWID, term.ROWID, llnk.*, term.*,node.* from SQL_SCCS00."**NODE**" node, SQL_SCCS00."**LOCAL-LINK**" llnk, SQL_SCCS00."**TERMINATION**" term where node.I4020_EXCH_GRP_CD=? and node.I4020_NODE_CAT=? and node.I4020_NODE_ID=? And "**S-NODE-SPR-TERM**" and "**S-TERM-UP-LL**"

Run the Statement through an EXPLAIN statement (Panel/Rexx)

NO AREA SWEEPS allowed

Set lengths may be an issue.



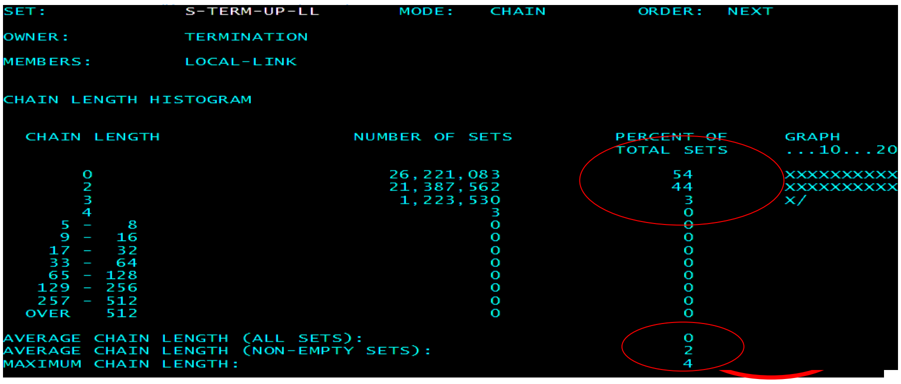
```
 ---------------------------- SQL EXPLAIN STEP LIST ----------- Row 1 to 5 of 5
  Option ===>

  SQL: SELECT LLNK.ROWID, TERM.ROWID, LLNK.*, TERM.*, NODE.* FROM SQL_SCCS00."NO
DE" NODE , SQL_SCCS00."LOCAL-LINK" LLNK,  SQL_SCCS00."TERMINATION" TERM WHERE NO
DE.I4020_EXCH_GRP_CD=? AND NODE.I4020_NODE_CAT=? AND NODE.I4020_NODE_ID=? AND "S
-NODE-SPR-TERM" AND "S-TERM-UP-LL"




             Select Step to see the step detail or PF3 to exit
   STEP  QB  STEP TYPE        Table               Access Mode      Access By
 _   1   1   Table Access     NODE                Calc
 _   2   1   Table Access     TERMINATION         Set Member       S-NODE-SPR-TE
 _   3   1   Nested Loop Join
 _   4   1   Table Access     LOCAL-LINK          Set Member       S-TERM-UP-LL
 _   5   1   Nested Loop Join
 ******************************* Bottom of data ********************************
```

Java Access to CA-IDMS Data at BT
SQL validation - process

Examine the DBAN reports for SET lengths

CALC sets

Set Members



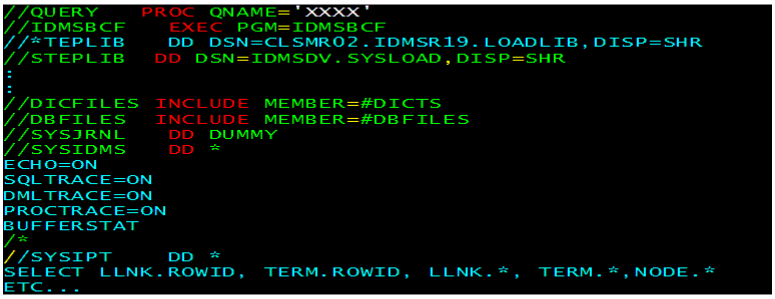
```
SET:                S-TERM-UP-LL       MODE:    CHAIN          ORDER:   NEXT

OWNER:              TERMINATION

MEMBERS:            LOCAL-LINK

CHAIN LENGTH HISTOGRAM

  CHAIN LENGTH                    NUMBER OF SETS          PERCENT OF     GRAPH
                                                          TOTAL SETS     ...10...20

       0                             26,221,083               54         XXXXXXXXXX
       2                             21,387,562               44         XXXXXXXXXX
       3                              1,223,530                3         X/
       4                                      3                0
     5 -   8                                  0                0
     9 -  16                                  0                0
    17 -  32                                  0                0
    33 -  64                                  0                0
    65 - 128                                  0                0
   129 - 256                                  0                0
   257 - 512                                  0                0
  OVER   512                                  0                0

AVERAGE CHAIN LENGTH (ALL SETS):                                   0
AVERAGE CHAIN LENGTH (NON-EMPTY SETS):                             2
MAXIMUM CHAIN LENGTH:                                              4
```

Java Access to CA-IDMS Data at BT
SQL validation – process

Run a Local Mode SQL BCF job with Sample data

Identify sample date with DMLO

Enter the Sample data into the JCL



```
//QUERY     PROC  QNAME='XXXX'
//IDMSBCF     EXEC  PGM=IDMSBCF
//*TEPLIB     DD  DSN=CLSMR02.IDMSR19.LOADLIB,DISP=SHR
//STEPLIB   DD  DSN=IDMSDV.SYSLOAD,DISP=SHR
:
:
//DICFILES  INCLUDE  MEMBER=#DICTS
//DBFILES   INCLUDE  MEMBER=#DBFILES
//SYSJRNL     DD  DUMMY
//SYSIDMS     DD  *
ECHO=ON
SQLTRACE=ON
DMLTRACE=ON
PROCTRACE=ON
BUFFERSTAT
/*
//SYSIPT      DD  *
SELECT  LLNK.ROWID,  TERM.ROWID,  LLNK.*,  TERM.*,NODE.*
ETC...
```

Java Access to CA-IDMS Data at BT
SQL validation - process

DML trace shows all the activity:

Is that what you expected?

```
VERB=19  OPEN
VERB=59  BIND SUBSCHEMA-->IDMSCATY    PROGRAM=IDMSBCF
VERB=107 BIND + Validate D2S2
VERB=107 BIND + Validate D2S2
VERB=16  FETCH
VERB=32  FIND   CALC                  REC-->NODE
VERB=18  FIND   First Record in Set   REC-->TERMINATION        SET-->S-NODE-SPR-
VERB=18  FIND   First Record in Set   REC-->LOCAL-LINK         SET-->S-TERM-UP-L
I D M S  SSCSTAT=0307  ERRREC=LOCAL-LINK  ERRSET=S-TERM-UP-LL  ERRAREA=TERMINATI
VERB=10  FIND   Next in Set           REC-->TERMINATION        SET-->S-NODE-SPR-
VERB=18  FIND   First Record in Set   REC-->LOCAL-LINK         SET-->S-TERM-UP-L
VERB=10  FIND   Next in Set           REC-->LOCAL-LINK         SET-->S-TERM-UP-L
I D M S  SSCSTAT=0307  ERRREC=LOCAL-LINK  ERRSET=S-TERM-UP-LL  ERRAREA=TERMINATI
VERB=10  FIND   Next in Set           REC-->TERMINATION        SET-->S-NODE-SPR-
I D M S  SSCSTAT=0307  ERRREC=TERMINATION  ERRSET=S-NODE-SPR-TERM  ERRAREA=ROUTI
  S Q L  SQLCODE=0100  REASON CODE=0000
VERB=03  CLOSE
```

Java Access to CA-IDMS Data at BT
SQL validation - process

Check the Buffer Stat for I/Os

Is that what you expected?

```
 *** DMCL ***          DMGLBRD  *** Name in Catalog *** DMGLBRD    *
  *** Buffer Name ***  BUF-TERMINATION    *** Pages ***        255 *
DB Page Requests                   107 Sequential Area Rqst          9
                                       Found in Buffer               9
                                       Found in Pref Buffer          0
Total DB Pages Read                 98 DB Pg Req:Tot Pg Read      1.0/1
Non-Prefetch I/O Rqst               98 Start I/O - Reads            98
Prefetch Requests                    0 Sequential Area Rqst          0
Start I/O - Prefetch                 0 Pages Read - Prefetch         0
                                       Pref Pages:Pref Strt I/O   0.0/1
```

Java Access to CA-IDMS Data at BT
SQL validation - proc

- Check the selected data colum

  Rowid And Virtual Foreign Keys

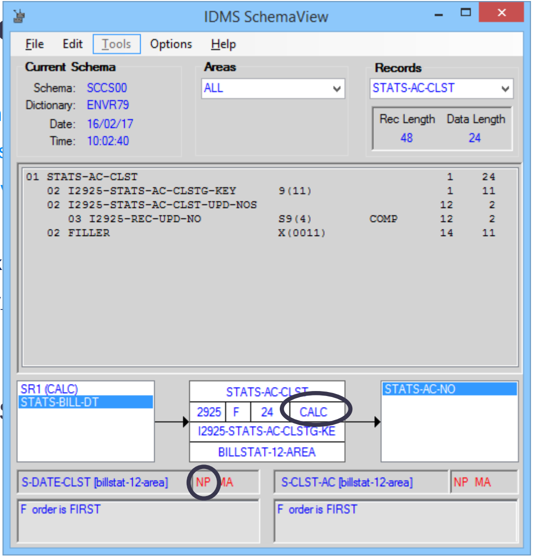
  Use of Select * can be expensi

EG: SELECT * from schema_w_fk

Produces extra IOs to return **Fkey**

Columns returned with SELECT *

- I2925_STATS-AC-CLS
- I2925-REC-UPD-NO
- Rowid
- Fkey_s_date_clst



Current Schema

Schema: SCCS00

Dictionary: ENVR79

Date: 16/02/17

Time: 10:02:40

Areas: ALL

Records: STATS-AC-CLST

Rec Length 48 Data Length 24

```
01 STATS-AC-CLST                                        1    24
   02 I2925-STATS-AC-CLSTG-KEY      9(11)               1    11
   02 I2925-STATS-AC-CLST-UPD-NOS                      12     2
      03 I2925-REC-UPD-NO           S9(4)      COMP    12     2
   02 FILLER                        X(0011)            14    11
```

Java Access to CA-IDMS Data at BT
SQL Table Procedures

- Some IDMS features are not supported by IDMSSQL
  - Occurs Depending On

    Only the Dependant column is exposed and available
  - Redefined Columns no exposed

    Only the Primary definition is available
  - FILLER

    FILLER is not available (was never recommended anyway but…)
  - Membership of System Index sets.

    MA System indexes are used for navigation.

    non-MA index sets are not.

Table procedure(s) can overcome these limitations

We used **CA SQL Quick Bridge**

Then rewrote our own TPs

Java Access to CA-IDMS
SQL Table Procedures

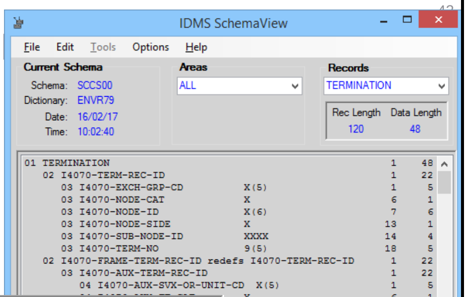
Example Redefined columns

Termination record

Problem:

redefined CALC Key around 10 times

CHAR (PIC X) to Numeric and Comp 3



Current Schema

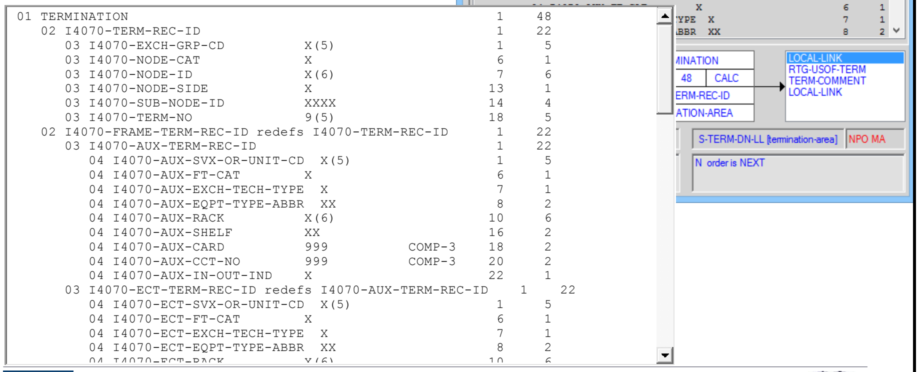
Schema: SCCS00
Dictionary: ENVR79
Date: 16/02/17
Time: 10:02:40

Areas: ALL

Records: TERMINATION

Rec Length: 120
Data Length: 48

```
01 TERMINATION                                          1    48
   02 I4070-TERM-REC-ID                                 1    22
      03 I4070-EXCH-GRP-CD          X(5)                1     5
      03 I4070-NODE-CAT             X                   6     1
      03 I4070-NODE-ID              X(6)                7     6
      03 I4070-NODE-SIDE            X                  13     1
      03 I4070-SUB-NODE-ID          XXXX               14     4
      03 I4070-TERM-NO              9(5)               18     5
   02 I4070-FRAME-TERM-REC-ID redefs I4070-TERM-REC-ID  1    22
      03 I4070-AUX-TERM-REC-ID                          1    22
         04 I4070-AUX-SVX-OR-UNIT-CD  X(5)              1     5
                                    X                   6     1
                                  YPE  X                7     1
                                  BBR  XX               8     2
```



```
01 TERMINATION                                          1    48
   02 I4070-TERM-REC-ID                                 1    22
      03 I4070-EXCH-GRP-CD          X(5)                1     5
      03 I4070-NODE-CAT             X                   6     1
      03 I4070-NODE-ID              X(6)                7     6
      03 I4070-NODE-SIDE            X                  13     1
      03 I4070-SUB-NODE-ID          XXXX               14     4
      03 I4070-TERM-NO              9(5)               18     5
   02 I4070-FRAME-TERM-REC-ID redefs I4070-TERM-REC-ID  1    22
      03 I4070-AUX-TERM-REC-ID                          1    22
         04 I4070-AUX-SVX-OR-UNIT-CD  X(5)              1     5
         04 I4070-AUX-FT-CAT        X                   6     1
         04 I4070-AUX-EXCH-TECH-TYPE  X                 7     1
         04 I4070-AUX-EQPT-TYPE-ABBR  XX                8     2
         04 I4070-AUX-RACK          X(6)               10     6
         04 I4070-AUX-SHELF         XX                 16     2
         04 I4070-AUX-CARD          999       COMP-3   18     2
         04 I4070-AUX-CCT-NO        999       COMP-3   20     2
         04 I4070-AUX-IN-OUT-IND    X                  22     1
      03 I4070-ECT-TERM-REC-ID redefs I4070-AUX-TERM-REC-ID   1    22
         04 I4070-ECT-SVX-OR-UNIT-CD  X(5)              1     5
         04 I4070-ECT-FT-CAT        X                   6     1
         04 I4070-ECT-EXCH-TECH-TYPE  X                 7     1
         04 I4070-ECT-EQPT-TYPE-ABBR  XX                8     2
         04 I4070-ECT-RACK          X(6)               10     6
```

Java Access to CA-IDMS Data at BT
SQL Table Procedures

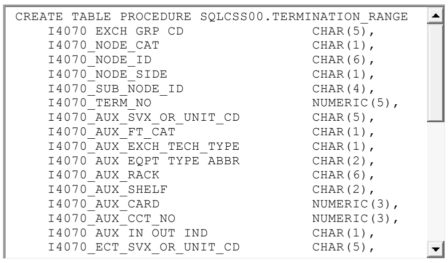
Defined a Table procedure to expose all columns, and the FKEY Columns

Accessed by any Calc Key combination

Or rowid

Or by the FKEY columns

Or by area sweep (for completeness)



```
CREATE TABLE PROCEDURE SQLCSS00.TERMINATION_RANGE
    I4070 EXCH GRP CD                   CHAR(5),
    I4070_NODE_CAT                      CHAR(1),
    I4070_NODE_ID                       CHAR(6),
    I4070_NODE_SIDE                     CHAR(1),
    I4070_SUB_NODE_ID                   CHAR(4),
    I4070_TERM_NO                       NUMERIC(5),
    I4070_AUX_SVX_OR_UNIT_CD            CHAR(5),
    I4070_AUX_FT_CAT                    CHAR(1),
    I4070_AUX_EXCH_TECH_TYPE            CHAR(1),
    I4070 AUX EQPT TYPE ABBR            CHAR(2),
    I4070_AUX_RACK                      CHAR(6),
    I4070_AUX_SHELF                     CHAR(2),
    I4070_AUX_CARD                      NUMERIC(3),
    I4070_AUX_CCT_NO                    NUMERIC(3),
    I4070 AUX IN OUT IND                CHAR(1),
    I4070_ECT_SVX_OR_UNIT_CD            CHAR(5),
```

Java Access to CA-IDMS Data at BT
SQL Table Procedures

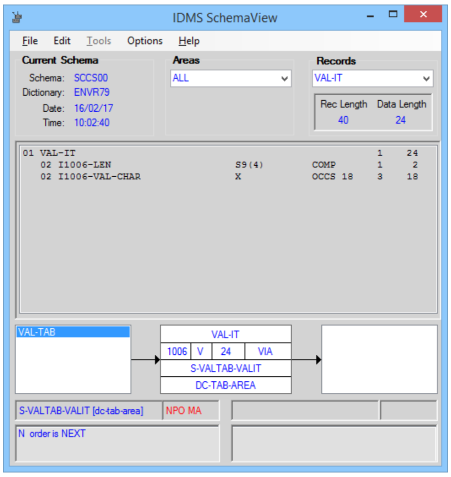
Example Occurs Depending On VAL-IT record

Problem: Occurs Depending On column of single chars.

- Table procedure written to expose each occurrence as a separate Column

- Will retrieve data by ROWID
- Or by Owner and all members of set.

Or By area Sweep (for completeness)



Schema: SCCS00
Dictionary: ENVR79
Date: 16/02/17
Time: 10:02:40

Areas: ALL

Records: VAL-IT

Rec Length: 40
Data Length: 24

```
01 VAL-IT                                                1    24
   02 I1006-LEN                  S9(4)       COMP        1     2
   02 I1006-VAL-CHAR             X           OCCS 18     3    18
```

S-VALTAB-VALIT [dc-tab-area] NPO MA

N order is NEXT

Java Access to CA-IDMS Data at BT
SQL Table Procedures

Example Occurs Depending On

VAL- record

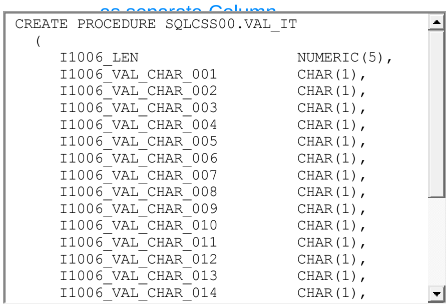
Problem:

Occurs Depending On Column

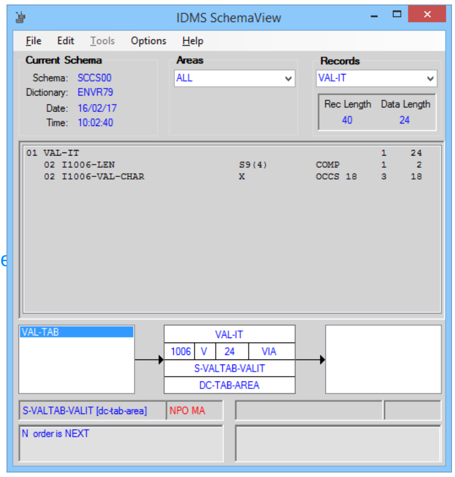
Single Character

Table procedure written to return each occurre

as separate Column



```
CREATE PROCEDURE SQLCSS00.VAL_IT
  (
    I1006_LEN                   NUMERIC(5),
    I1006_VAL_CHAR_001          CHAR(1),
    I1006_VAL_CHAR_002          CHAR(1),
    I1006_VAL_CHAR_003          CHAR(1),
    I1006_VAL_CHAR_004          CHAR(1),
    I1006_VAL_CHAR_005          CHAR(1),
    I1006_VAL_CHAR_006          CHAR(1),
    I1006_VAL_CHAR_007          CHAR(1),
    I1006_VAL_CHAR_008          CHAR(1),
    I1006_VAL_CHAR_009          CHAR(1),
    I1006_VAL_CHAR_010          CHAR(1),
    I1006_VAL_CHAR_011          CHAR(1),
    I1006_VAL_CHAR_012          CHAR(1),
    I1006_VAL_CHAR_013          CHAR(1),
    I1006_VAL_CHAR_014          CHAR(1),
```



Current Schema

Schema: SCCS00
Dictionary: ENVR79
Date: 16/02/17
Time: 10:02:40

Areas: ALL

Records: VAL-IT

Rec Length: 40
Data Length: 24

```
01 VAL-IT                                            1     24
   02 I1006-LEN                    S9(4)      COMP   1      2
   02 I1006-VAL-CHAR               X          OCCS 18 3    18
```

S-VALTAB-VALIT [dc-tab-area] NPO MA

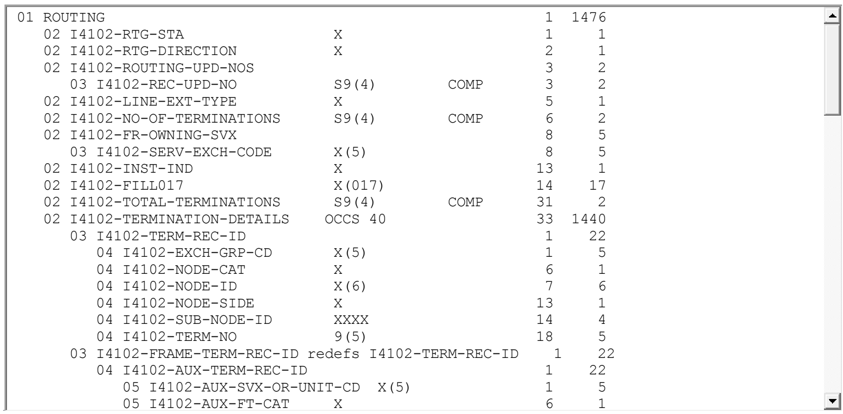
N order is NEXT

Java Access to CA-IDMS Data at BT
SQL Table Procedures

Example Occurs Depending On

ROUTING Record

Problem: Occurs Depending On but for Multiple Columns and (some are also redefined)



```
01 ROUTING                                                 1  1476
   02 I4102-RTG-STA                X                       1     1
   02 I4102-RTG-DIRECTION          X                       2     1
   02 I4102-ROUTING-UPD-NOS                                3     2
      03 I4102-REC-UPD-NO          S9(4)        COMP       3     2
   02 I4102-LINE-EXT-TYPE          X                       5     1
   02 I4102-NO-OF-TERMINATIONS     S9(4)        COMP       6     2
   02 I4102-FR-OWNING-SVX                                  8     5
      03 I4102-SERV-EXCH-CODE      X(5)                    8     5
   02 I4102-INST-IND               X                      13     1
   02 I4102-FILL017                X(017)                 14    17
   02 I4102-TOTAL-TERMINATIONS     S9(4)        COMP      31     2
   02 I4102-TERMINATION-DETAILS   OCCS 40                 33  1440
      03 I4102-TERM-REC-ID                                 1    22
         04 I4102-EXCH-GRP-CD      X(5)                    1     5
         04 I4102-NODE-CAT         X                       6     1
         04 I4102-NODE-ID          X(6)                    7     6
         04 I4102-NODE-SIDE        X                      13     1
         04 I4102-SUB-NODE-ID      XXXX                   14     4
         04 I4102-TERM-NO          9(5)                   18     5
      03 I4102-FRAME-TERM-REC-ID redefs I4102-TERM-REC-ID   1    22
         04 I4102-AUX-TERM-REC-ID                          1    22
            05 I4102-AUX-SVX-OR-UNIT-CD  X(5)              1     5
            05 I4102-AUX-FT-CAT    X                       6     1
```

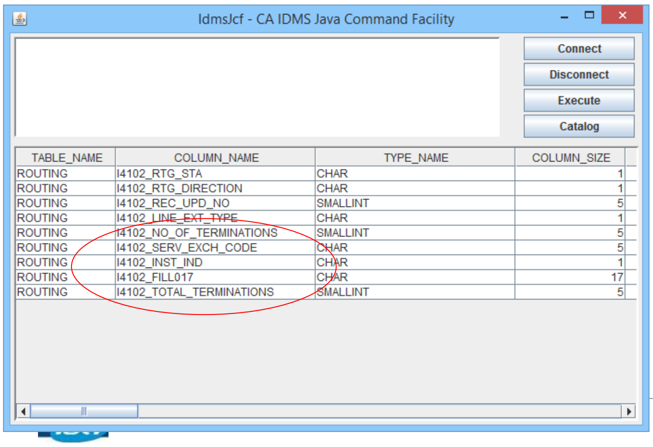
Java Access to CA-IDMS Data at BT
SQL Table Procedures

IDMS SQL only exposes the non occurrence columns in SQL_SCCS00."ROUTING" table.

ROUTING Record

Problem:

Occurs Depending On but for Multiple Columns



TABLE_NAME	COLUMN_NAME	TYPE_NAME	COLUMN_SIZE
ROUTING	I4102_RTG_STA	CHAR	1
ROUTING	I4102_RTG_DIRECTION	CHAR	1
ROUTING	I4102_REC_UPD_NO	SMALLINT	5
ROUTING	I4102_LINE_EXT_TYPE	CHAR	1
ROUTING	I4102_NO_OF_TERMINATIONS	SMALLINT	5
ROUTING	I4102_SERV_EXCH_CODE	CHAR	5
ROUTING	I4102_INST_IND	CHAR	1
ROUTING	I4102_FILL017	CHAR	17
ROUTING	I4102_TOTAL_TERMINATIONS	SMALLINT	5

Java Access to CA-IDMS Data at BT
SQL Table Procedures

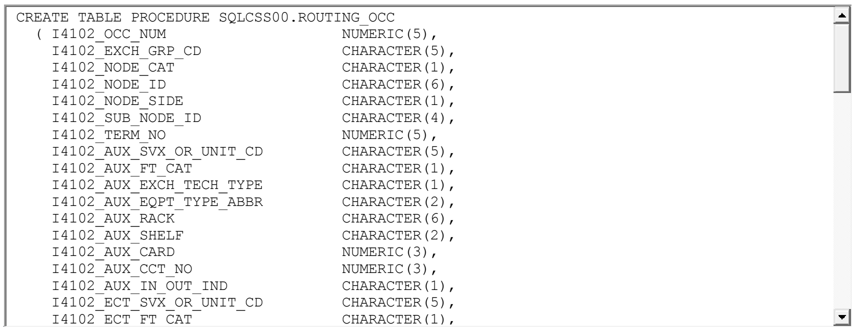
Table procedure ROUTING_OCC written to return each occurrence as separate ROW

Needed an additional Occurrence column.

Retrieved by Rowid

Or by Fkey Owner of set.

Or by Area Sweep



```
CREATE TABLE PROCEDURE SQLCSS00.ROUTING_OCC
  ( I4102_OCC_NUM                   NUMERIC(5),
    I4102_EXCH_GRP_CD               CHARACTER(5),
    I4102_NODE_CAT                  CHARACTER(1),
    I4102_NODE_ID                   CHARACTER(6),
    I4102_NODE_SIDE                 CHARACTER(1),
    I4102_SUB_NODE_ID               CHARACTER(4),
    I4102_TERM_NO                   NUMERIC(5),
    I4102_AUX_SVX_OR_UNIT_CD        CHARACTER(5),
    I4102_AUX_FT_CAT                CHARACTER(1),
    I4102_AUX_EXCH_TECH_TYPE        CHARACTER(1),
    I4102_AUX_EQPT_TYPE_ABBR        CHARACTER(2),
    I4102_AUX_RACK                  CHARACTER(6),
    I4102_AUX_SHELF                 CHARACTER(2),
    I4102_AUX_CARD                  NUMERIC(3),
    I4102_AUX_CCT_NO                NUMERIC(3),
    I4102_AUX_IN_OUT_IND            CHARACTER(1),
    I4102_ECT_SVX_OR_UNIT_CD        CHARACTER(5),
    I4102_ECT_FT_CAT                CHARACTER(1),
```

Java Access to CA-IDMS Data at BT
SQL Table Procedures

Things to watch with TP:

WHERE Clause columns only passed to TP if “=“ is specified

e.g. WHERE column > x is not seen by Table Procedure.

Whatever is supplied in WHERE column = must be returned exactly

eg: don’t be clever with data

Code with MODE = BATCH unless DC functions required

TP program is called at least 3 times in each query.

1- Initiate
2- Called each time to return a row
2a - return with no-more-rows
4 – Finish

We Treat TPs like Database tables, I.E. all data requests are valid

EG: cater for daft SQL requests and Area sweeps anyway

TP call does not pass columns in SELECT, So return all columns

Care taken to return FKEY columns. Could result in Multiple I/Os

We have used “WHERE FKEY = 0” to tell TP don’t return FKEY column.

Java Access to CA-IDMS Data at BT
VIRTUAL KEYS Schema

Existing SQL may not work as expected

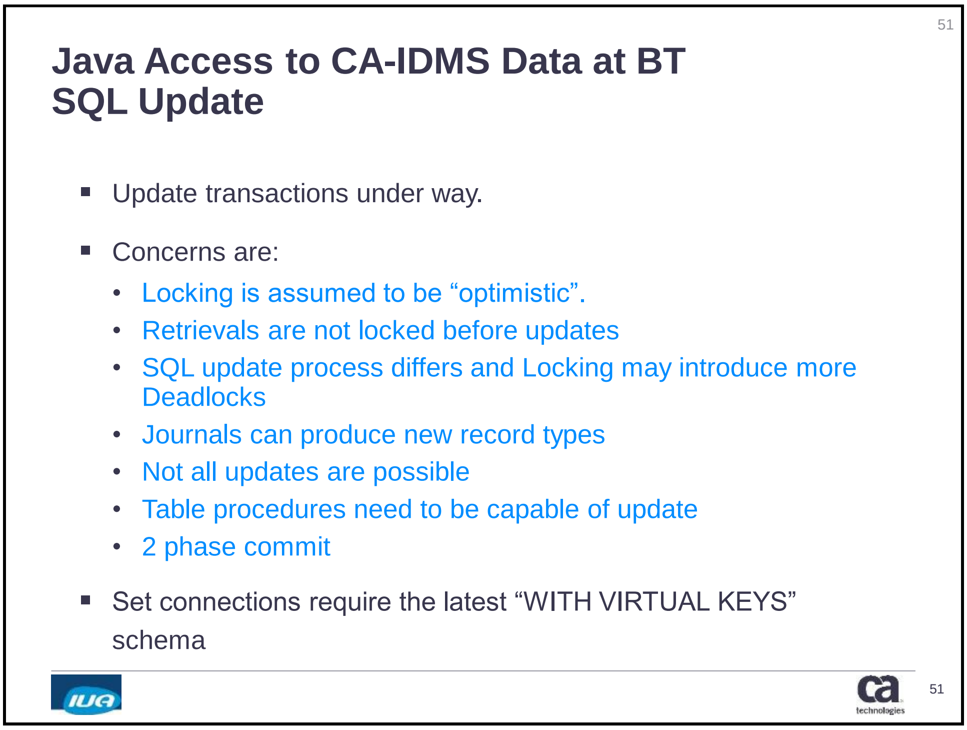
SELECT rowid , * from tablename returns all data columns and rowid

But against VFK Schema will return two ROWID columns, plus additional FKEY columns.

If Member records do not hold Owner Pointers, satisfying the implied request for OWNER will result in IDMS reading the set to the Owner.

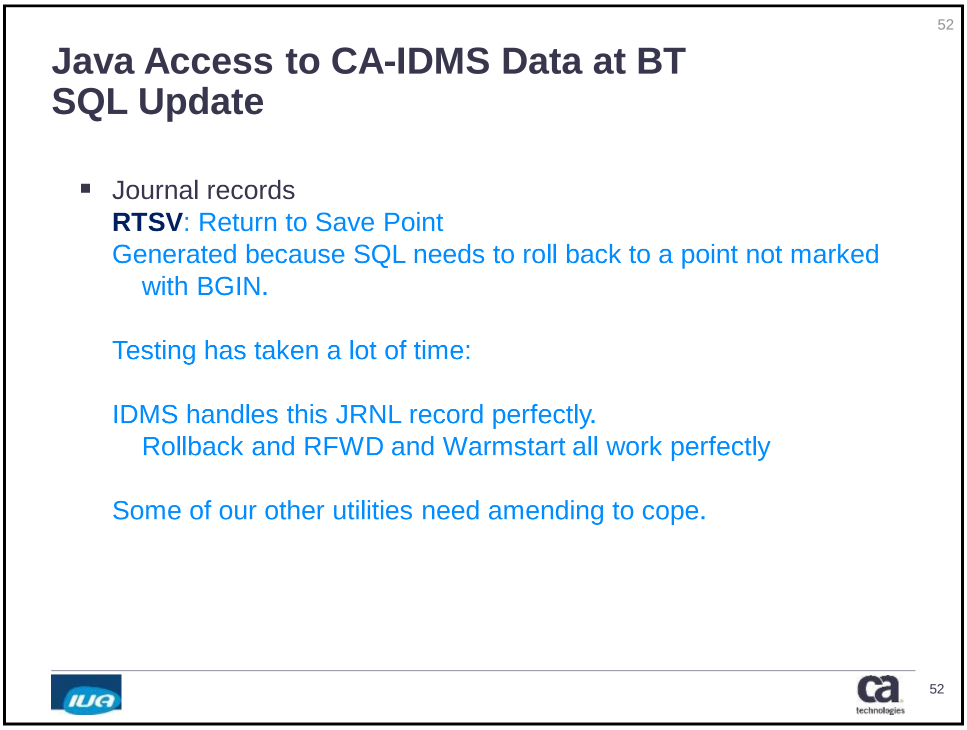
Many User Owned Index sets do not hold Owner pointers.
So inadvertently an SQL query one could result in additional I/Os.
extra I/O is expensive. Also MULTIPLE owners

- Rowid is accepted differently in the two Schemas
- EG: select xxx where rowid = x’0FB14401’ Works on NON-FK schema
- Does not work on VFK schema



Java Access to CA-IDMS Data at BT
SQL Update

- Update transactions under way.
- Concerns are:
  - Locking is assumed to be “optimistic”.
  - Retrievals are not locked before updates
  - SQL update process differs and Locking may introduce more Deadlocks
  - Journals can produce new record types
  - Not all updates are possible
  - Table procedures need to be capable of update
  - 2 phase commit
- Set connections require the latest “WITH VIRTUAL KEYS” schema



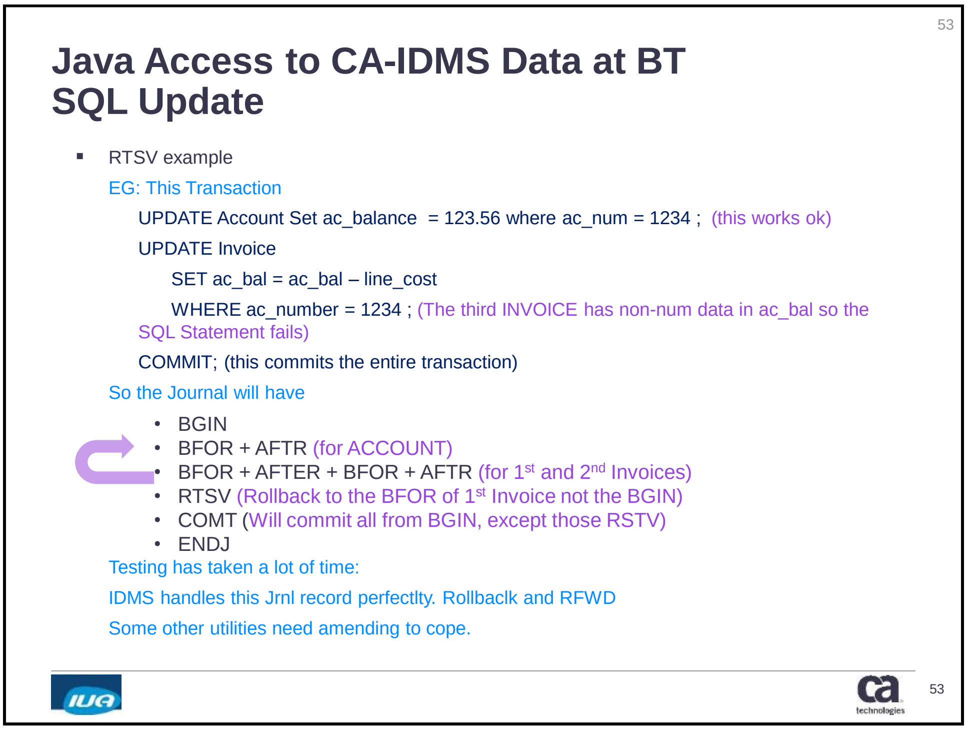
Java Access to CA-IDMS Data at BT
SQL Update

- Journal records
  RTSV: Return to Save Point
  Generated because SQL needs to roll back to a point not marked with BGIN.

  Testing has taken a lot of time:

  IDMS handles this JRNL record perfectly.
  Rollback and RFWD and Warmstart all work perfectly

  Some of our other utilities need amending to cope.



Java Access to CA-IDMS Data at BT
SQL Update

- RTSV example

EG: This Transaction

UPDATE Account Set ac_balance = 123.56 where ac_num = 1234 ; (this works ok)

UPDATE Invoice

SET ac_bal = ac_bal – line_cost

WHERE ac_number = 1234 ; (The third INVOICE has non-num data in ac_bal so the SQL Statement fails)

COMMIT; (this commits the entire transaction)

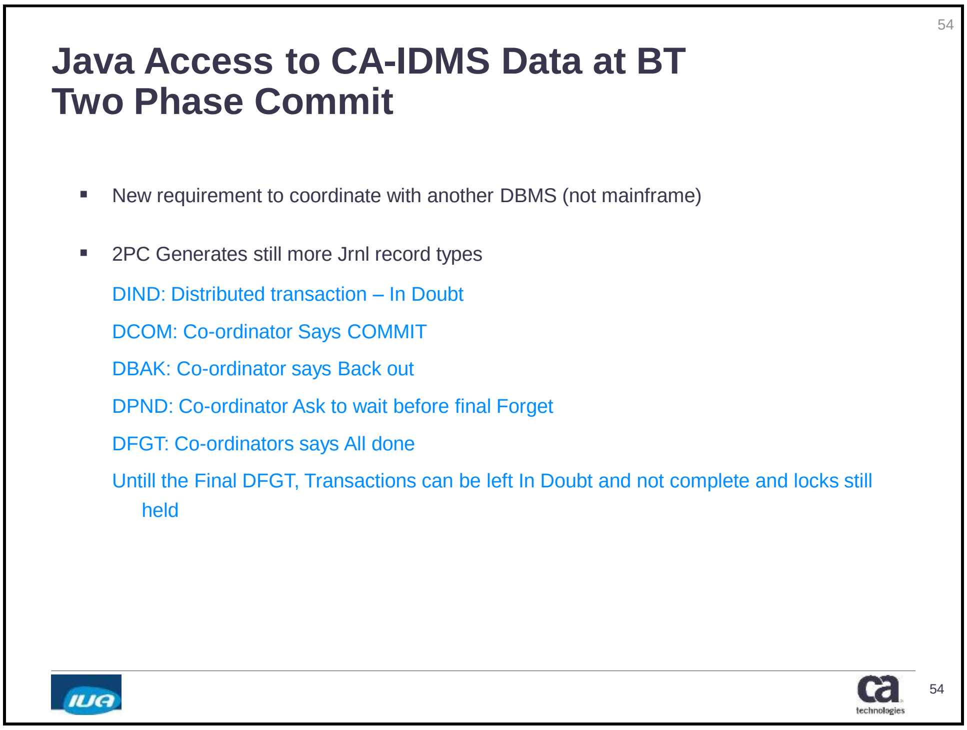
So the Journal will have

- BGIN
- BFOR + AFTR (for ACCOUNT)
- BFOR + AFTER + BFOR + AFTR (for 1st and 2nd Invoices)
- RTSV (Rollback to the BFOR of 1st Invoice not the BGIN)
- COMT (Will commit all from BGIN, except those RSTV)
- ENDJ

Testing has taken a lot of time:

IDMS handles this Jrnl record perfectlty. Rollbaclk and RFWD

Some other utilities need amending to cope.



Java Access to CA-IDMS Data at BT
Two Phase Commit

- New requirement to coordinate with another DBMS (not mainframe)
- 2PC Generates still more Jrnl record types

DIND: Distributed transaction – In Doubt

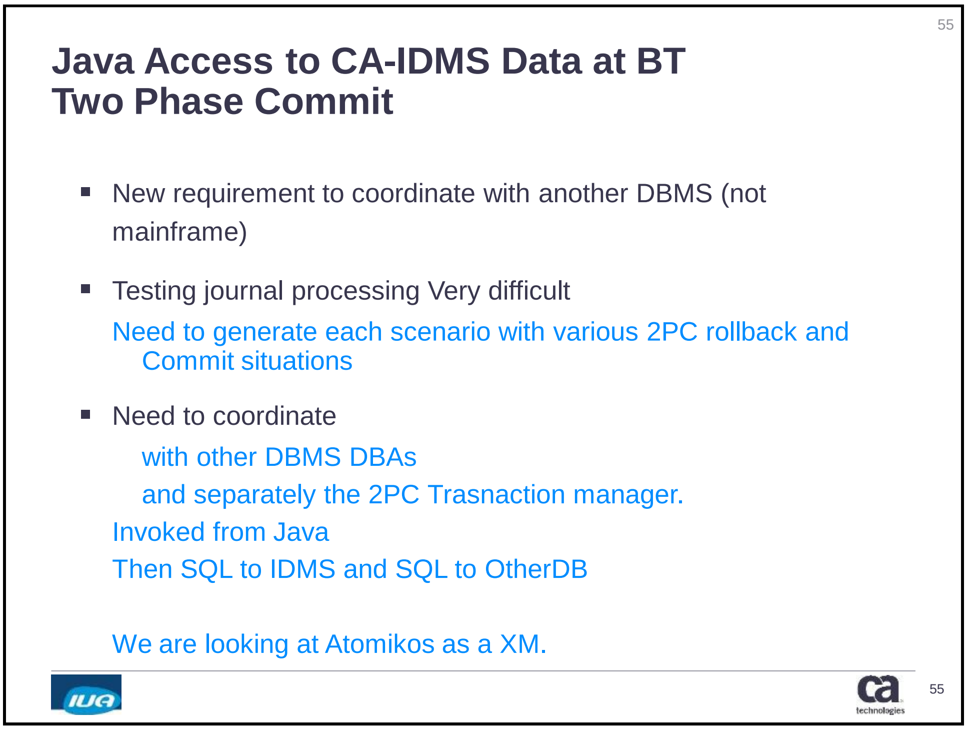
DCOM: Co-ordinator Says COMMIT

DBAK: Co-ordinator says Back out

DPND: Co-ordinator Ask to wait before final Forget

DFGT: Co-ordinators says All done

Untill the Final DFGT, Transactions can be left In Doubt and not complete and locks still held



Java Access to CA-IDMS Data at BT
Two Phase Commit

- New requirement to coordinate with another DBMS (not mainframe)
- Testing journal processing Very difficult

  Need to generate each scenario with various 2PC rollback and Commit situations
- Need to coordinate

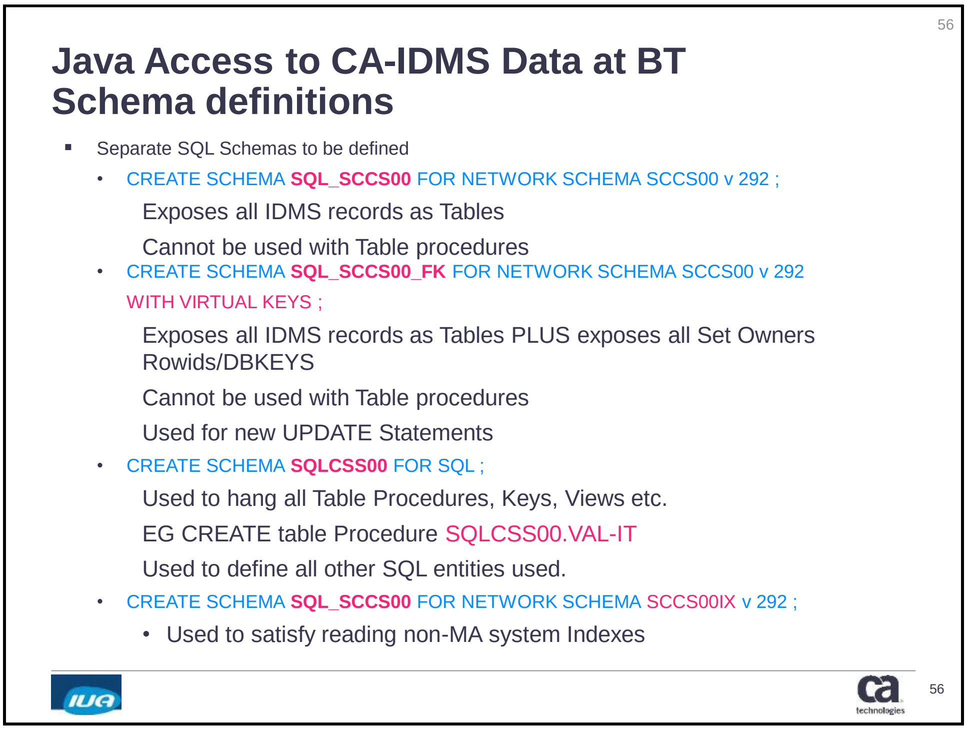
  with other DBMS DBAs

  and separately the 2PC Trasnaction manager.

  Invoked from Java

  Then SQL to IDMS and SQL to OtherDB

  We are looking at Atomikos as a XM.



Java Access to CA-IDMS Data at BT
Schema definitions

- Separate SQL Schemas to be defined
  - CREATE SCHEMA **SQL_SCCS00** FOR NETWORK SCHEMA SCCS00 v 292 ;

    Exposes all IDMS records as Tables

    Cannot be used with Table procedures
  - CREATE SCHEMA **SQL_SCCS00_FK** FOR NETWORK SCHEMA SCCS00 v 292 WITH VIRTUAL KEYS ;

    Exposes all IDMS records as Tables PLUS exposes all Set Owners Rowids/DBKEYS

    Cannot be used with Table procedures

    Used for new UPDATE Statements
  - CREATE SCHEMA **SQLCSS00** FOR SQL ;

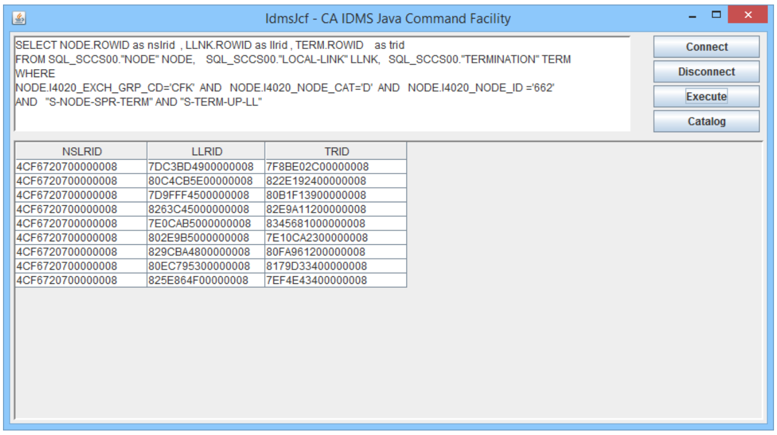
    Used to hang all Table Procedures, Keys, Views etc.

    EG CREATE table Procedure SQLCSS00.VAL-IT

    Used to define all other SQL entities used.
  - CREATE SCHEMA **SQL_SCCS00** FOR NETWORK SCHEMA SCCS00IX v 292 ;
    - Used to satisfy reading non-MA system Indexes

Java Access to CA-IDMS Data at BT
Other points

- Using JCFDEMO gives more confidence

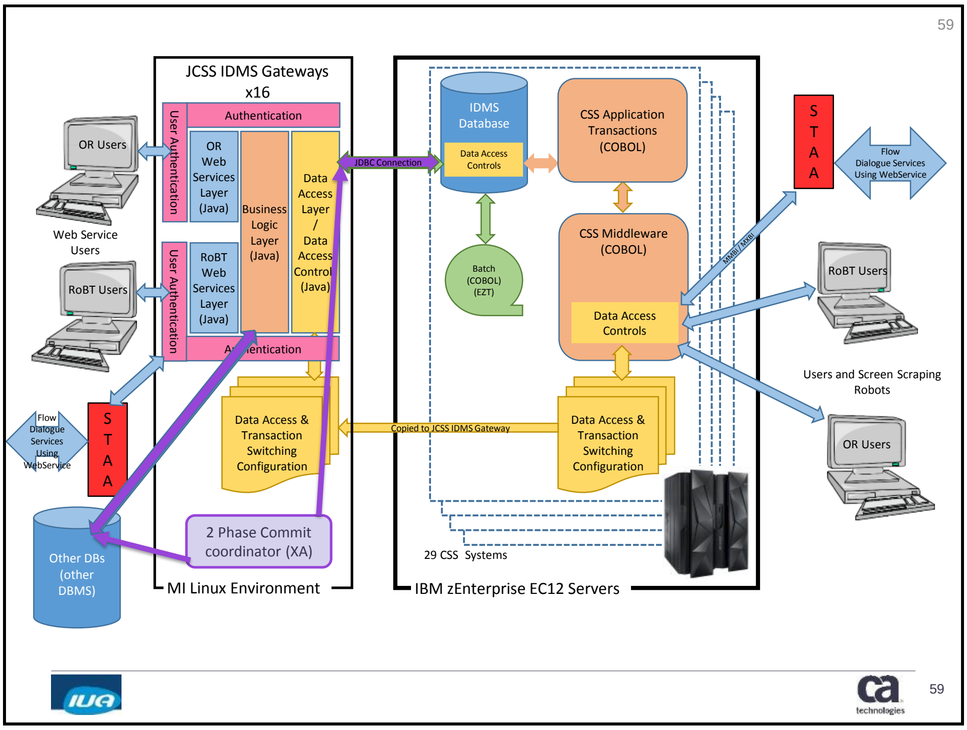


```
SELECT NODE.ROWID as nslrid  , LLNK.ROWID as llrid , TERM.ROWID    as trid
FROM SQL_SCCS00."NODE" NODE,    SQL_SCCS00."LOCAL-LINK" LLNK,   SQL_SCCS00."TERMINATION" TERM
WHERE
NODE.I4020_EXCH_GRP_CD='CFK'  AND   NODE.I4020_NODE_CAT='D'  AND   NODE.I4020_NODE_ID ='662'
AND   "S-NODE-SPR-TERM" AND "S-TERM-UP-LL"
```

NSLRID	LLRID	TRID
4CF6720700000008	7DC3BD4900000008	7F8BE02C00000008
4CF6720700000008	80C4CB5E00000008	822E192400000008
4CF6720700000008	7D9FFF4500000008	80B1F13900000008
4CF6720700000008	8263C45000000008	82E9A11200000008
4CF6720700000008	7E0CAB5000000008	8345681000000008
4CF6720700000008	802E9B5000000008	7E10CA2300000008
4CF6720700000008	829CBA4800000008	80FA961200000008
4CF6720700000008	80EC795300000008	8179D33400000008
4CF6720700000008	825E864F00000008	7EF4E43400000008

Java Access to CA-IDMS Data at BT
Summary

- BT are continuing to exploit the modernisation techniques provided in IDMS to great effect
- CA continue to enhance and exploit the SQL and Java drivers
- Update SQL is coming
- 2 phase Commit processing is coming
- One last thing…
  Remember that diagram?



JCSS IDMS Gateways x16

User Authentication

Authentication

OR Web Services Layer (Java)

RoBT Web Services Layer (Java)

Business Logic Layer (Java)

Data Access Layer / Data Access Control (Java)

Authentication

Data Access & Transaction Switching Configuration

2 Phase Commit coordinator (XA)

MI Linux Environment

OR Users

Web Service Users

RoBT Users

Flow Dialogue Services Using WebService

S T A A

Other DBs (other DBMS)

JDBC Connection

Copied to JCSS IDMS Gateway

IDMS Database

Data Access Controls

CSS Application Transactions (COBOL)

Batch (COBOL) (EZT)

CSS Middleware (COBOL)

Data Access Controls

Data Access & Transaction Switching Configuration

29 CSS Systems

IBM zEnterprise EC12 Servers

MMBI / MXBI

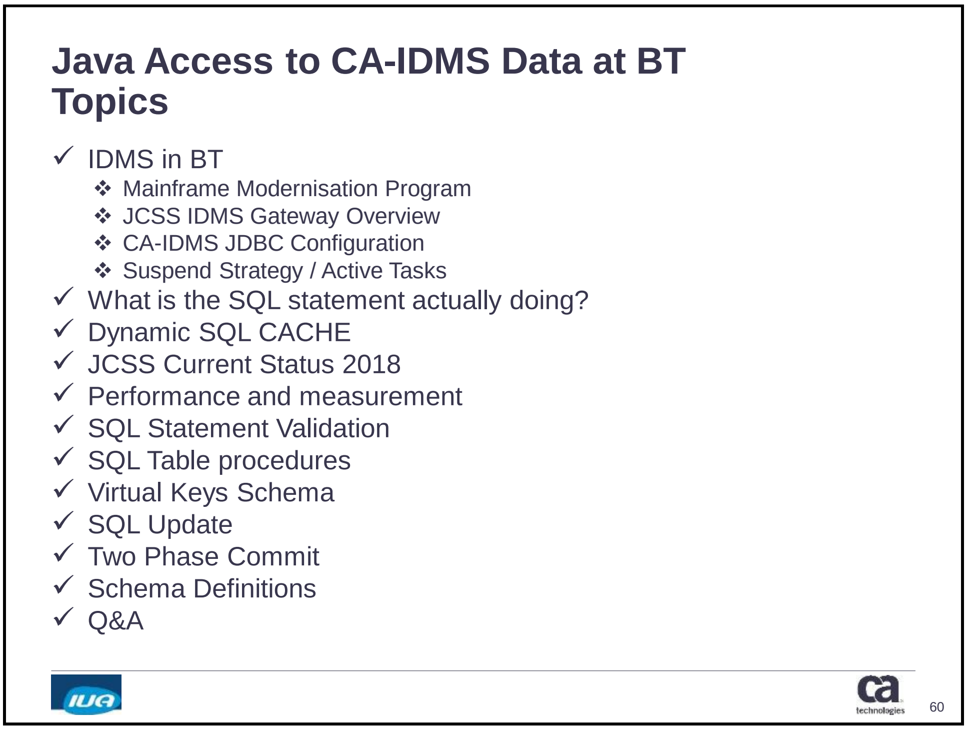
S T A A

Flow Dialogue Services Using WebService

RoBT Users

Users and Screen Scraping Robots

OR Users



Java Access to CA-IDMS Data at BT Topics

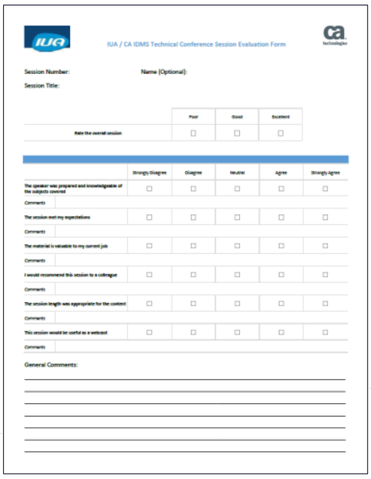
- ✓ IDMS in BT
  - ❖ Mainframe Modernisation Program
  - ❖ JCSS IDMS Gateway Overview
  - ❖ CA-IDMS JDBC Configuration
  - ❖ Suspend Strategy / Active Tasks
- ✓ What is the SQL statement actually doing?
- ✓ Dynamic SQL CACHE
- ✓ JCSS Current Status 2018
- ✓ Performance and measurement
- ✓ SQL Statement Validation
- ✓ SQL Table procedures
- ✓ Virtual Keys Schema
- ✓ SQL Update
- ✓ Two Phase Commit
- ✓ Schema Definitions
- ✓ Q&A

Java Access to CA-IDMS Data at BT

Please Complete a Session Evaluation Form

- The number for this session is **A04**

- After completing your session evaluation form, place it in the envelope at the front of the room



IUA / CA IDMS Technical Conference Session Evaluation Form

Session Number: Name (Optional):

Session Title:

	Poor	Good	Excellent
Rate the overall session	□	□	□

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The speaker was prepared and knowledgeable of the subjects covered	□	□	□	□	□
Comments					
The session met my expectations	□	□	□	□	□
Comments					
The material is valuable to my current job	□	□	□	□	□
Comments					
I would recommend this session to a colleague	□	□	□	□	□
Comments					
The session length was appropriate for the content	□	□	□	□	□
Comments					
This session would be useful as a webcast	□	□	□	□	□
Comments					

General Comments: