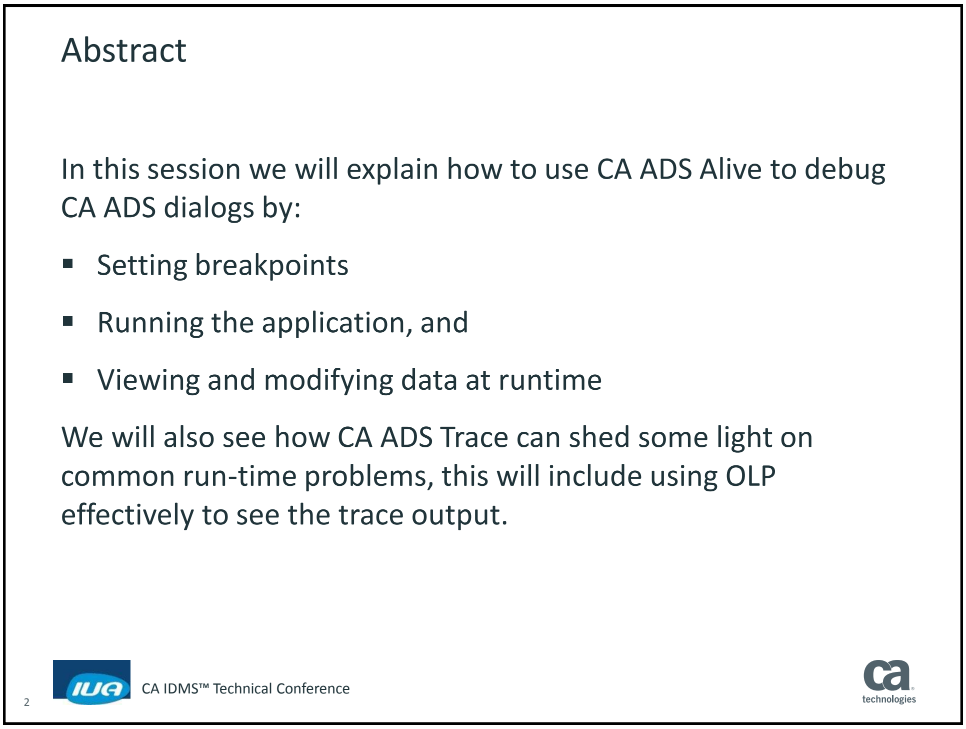
Debugging CA ADS Dialogs: Using CA ADS Alive, CA ADS Trace and OLP

Gary Cherlet
garty@cherlet.com

CA IDMS™ Technical Conference

Framingham MA
December 2-5, 2014

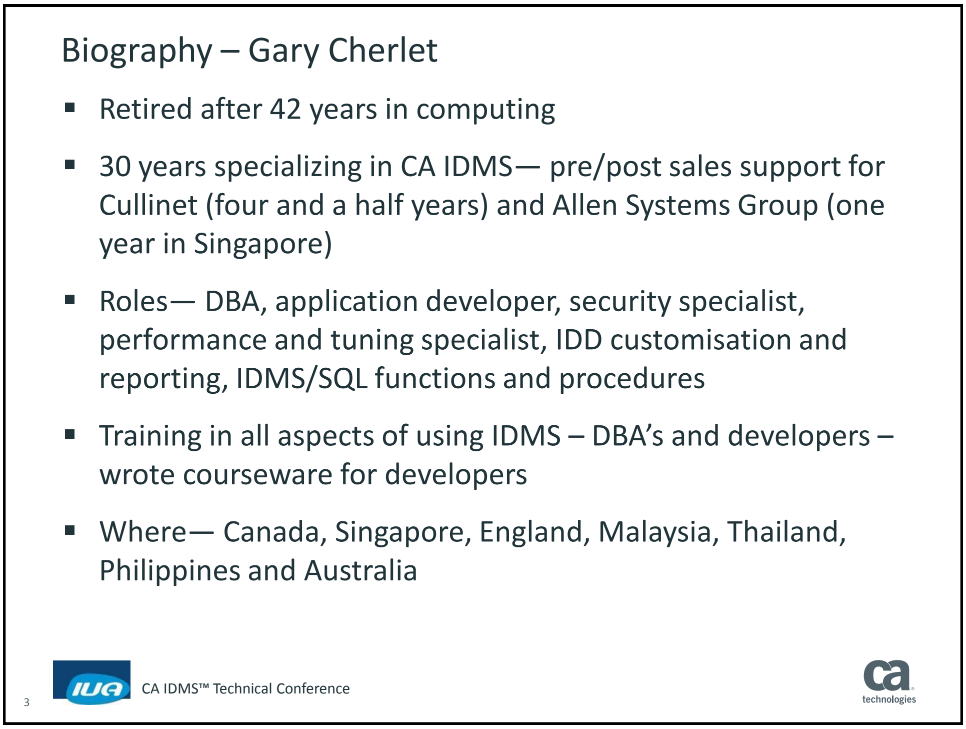


Abstract

In this session we will explain how to use CA ADS Alive to debug CA ADS dialogs by:

- Setting breakpoints
- Running the application, and
- Viewing and modifying data at runtime

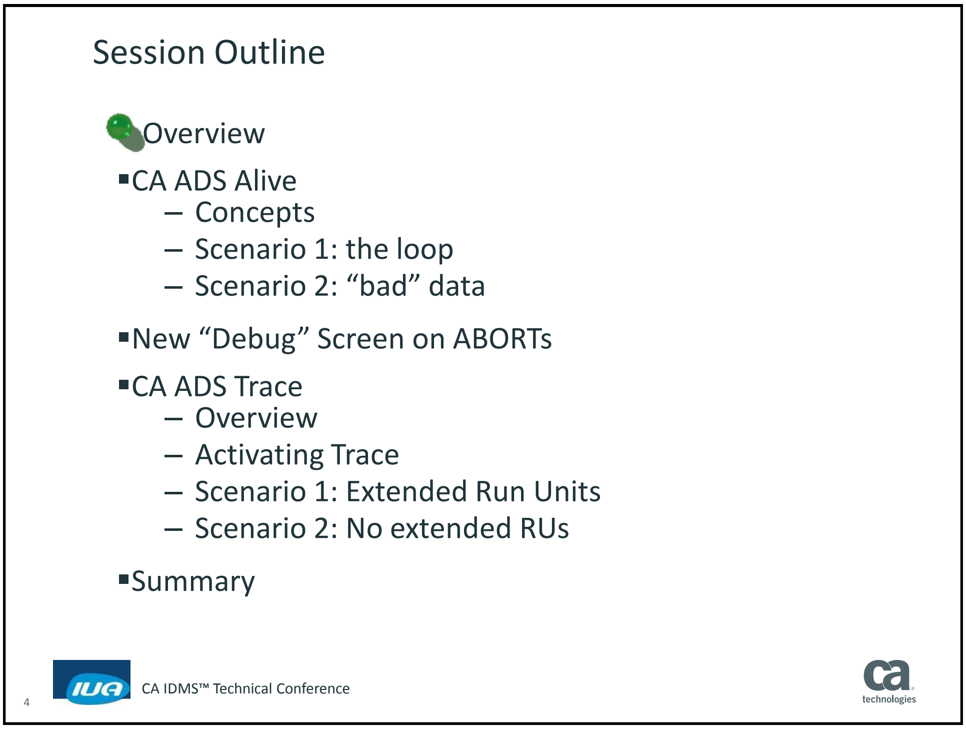
We will also see how CA ADS Trace can shed some light on common run-time problems, this will include using OLP effectively to see the trace output.



Biography – Gary Cherlet

- Retired after 42 years in computing
- 30 years specializing in CA IDMS— pre/post sales support for Cullinet (four and a half years) and Allen Systems Group (one year in Singapore)
- Roles— DBA, application developer, security specialist, performance and tuning specialist, IDD customisation and reporting, IDMS/SQL functions and procedures
- Training in all aspects of using IDMS – DBA's and developers – wrote courseware for developers
- Where— Canada, Singapore, England, Malaysia, Thailand, Philippines and Australia

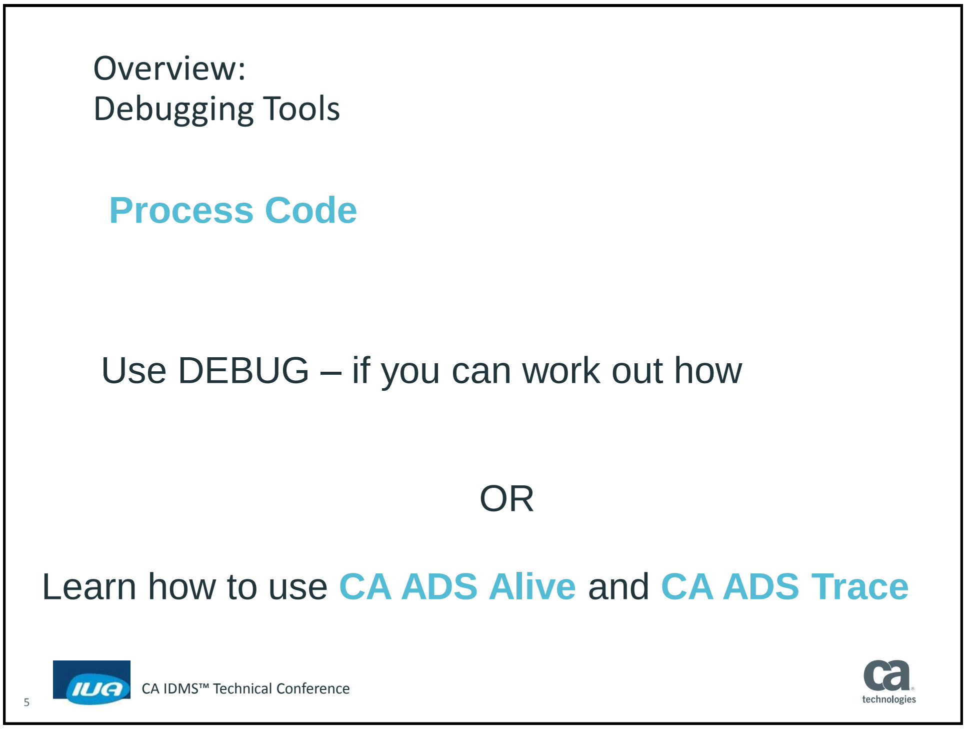
3 CA IDMS™ Technical Conference



Session Outline

- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: "bad" data
- New "Debug" Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary

4 CA IDMS™ Technical Conference



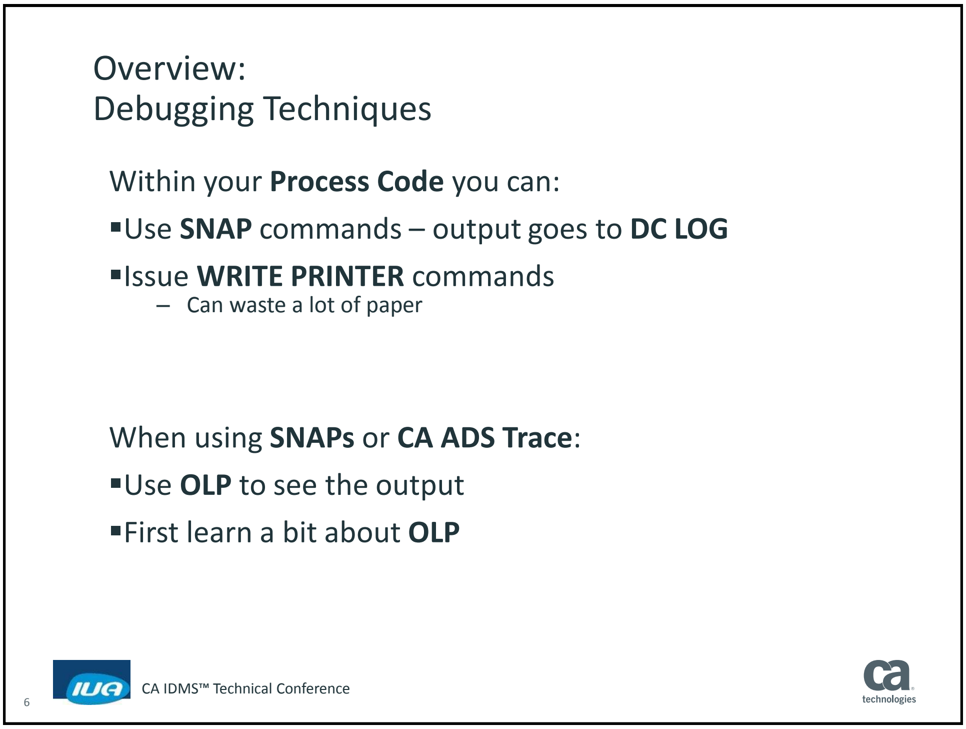
Overview:
Debugging Tools

Process Code

Use DEBUG – if you can work out how

OR

Learn how to use **CA ADS Alive** and **CA ADS Trace**



Overview:
Debugging Techniques

Within your **Process Code** you can:

- Use **SNAP** commands – output goes to **DC LOG**
- Issue **WRITE PRINTER** commands
  - Can waste a lot of paper

When using **SNAPs** or **CA ADS Trace**:

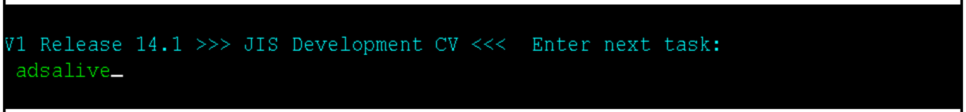
- Use **OLP** to see the output
- First learn a bit about **OLP**

Session Outline

- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary

CA ADS Alive: Concepts

- Set up Phase
  - Animating the dialog(s)
  - Setting breakpoints
- Execution (run-time) Phase
  - Encounter breakpoints
  - View and modify data
  - Add / Remove breakpoints
- Getting started:

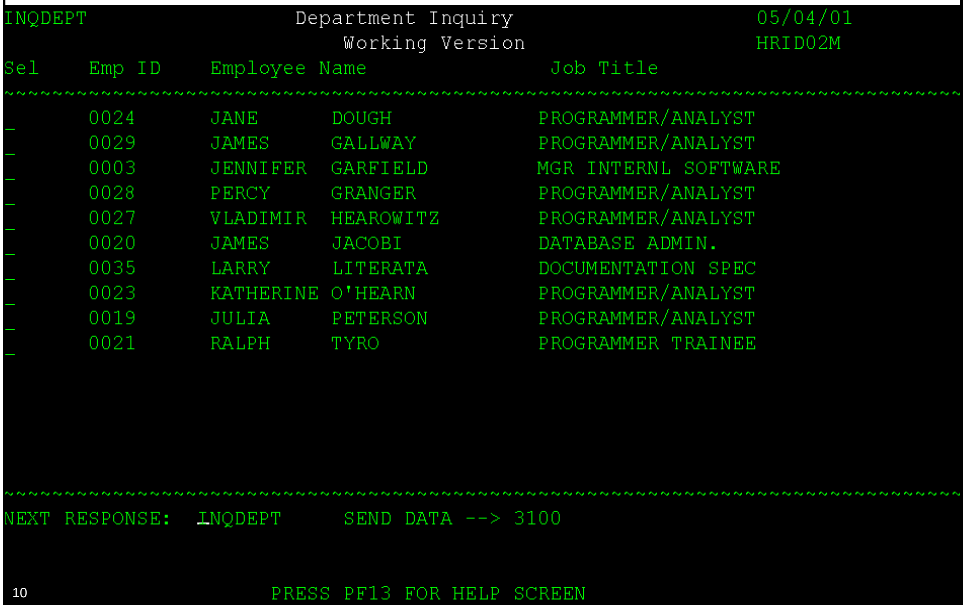


```
V1 Release 14.1 >>> JIS Development CV <<<  Enter next task:
 adsalive_
```

Session Outline

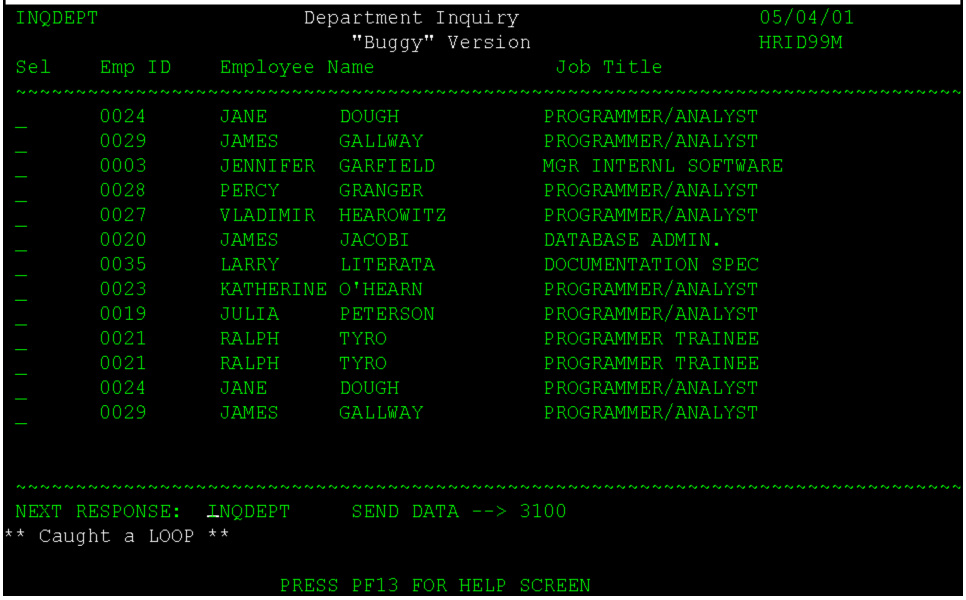
- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: "bad" data
- New "Debug" Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary

Scenario 1: the Loop
What we "should" be getting



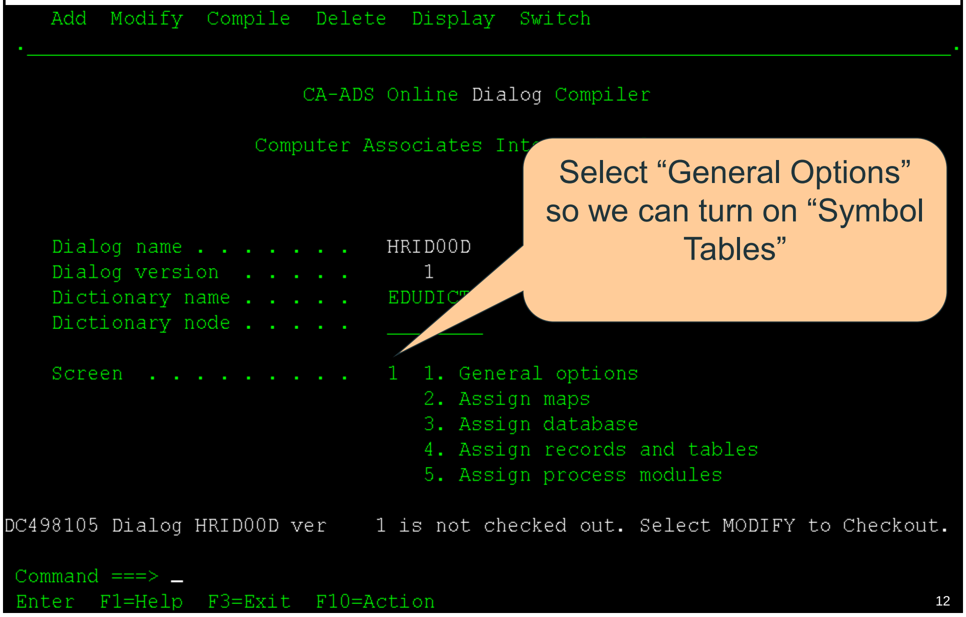
```
INQDEPT                   Department Inquiry                     05/04/01
                              Working Version                    HRID02M
Sel     Emp ID     Employee Name                Job Title
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
_       0024       JANE       DOUGH            PROGRAMMER/ANALYST
_       0029       JAMES      GALLWAY          PROGRAMMER/ANALYST
_       0003       JENNIFER   GARFIELD         MGR INTERNL SOFTWARE
_       0028       PERCY      GRANGER          PROGRAMMER/ANALYST
_       0027       VLADIMIR   HEAROWITZ        PROGRAMMER/ANALYST
_       0020       JAMES      JACOBI           DATABASE ADMIN.
_       0035       LARRY      LITERATA         DOCUMENTATION SPEC
_       0023       KATHERINE  O'HEARN          PROGRAMMER/ANALYST
_       0019       JULIA      PETERSON         PROGRAMMER/ANALYST
_       0021       RALPH      TYRO             PROGRAMMER TRAINEE






~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NEXT RESPONSE:  INQDEPT     SEND DATA --> 3100


                      PRESS PF13 FOR HELP SCREEN
```

Scenario 1: the Loop
What we are getting



```
INQDEPT                  Department Inquiry                    05/04/01
                             "Buggy" Version                   HRID99M
Sel    Emp ID    Employee Name              Job Title
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
_      0024      JANE      DOUGH            PROGRAMMER/ANALYST
_      0029      JAMES     GALLWAY          PROGRAMMER/ANALYST
_      0003      JENNIFER  GARFIELD         MGR INTERNL SOFTWARE
_      0028      PERCY     GRANGER          PROGRAMMER/ANALYST
_      0027      VLADIMIR  HEAROWITZ        PROGRAMMER/ANALYST
_      0020      JAMES     JACOBI           DATABASE ADMIN.
_      0035      LARRY     LITERATA         DOCUMENTATION SPEC
_      0023      KATHERINE O'HEARN          PROGRAMMER/ANALYST
_      0019      JULIA     PETERSON         PROGRAMMER/ANALYST
_      0021      RALPH     TYRO             PROGRAMMER TRAINEE
_      0021      RALPH     TYRO             PROGRAMMER TRAINEE
_      0024      JANE      DOUGH            PROGRAMMER/ANALYST
_      0029      JAMES     GALLWAY          PROGRAMMER/ANALYST
_


~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NEXT RESPONSE:  INQDEPT     SEND DATA --> 3100
** Caught a LOOP **

                          PRESS PF13 FOR HELP SCREEN
```

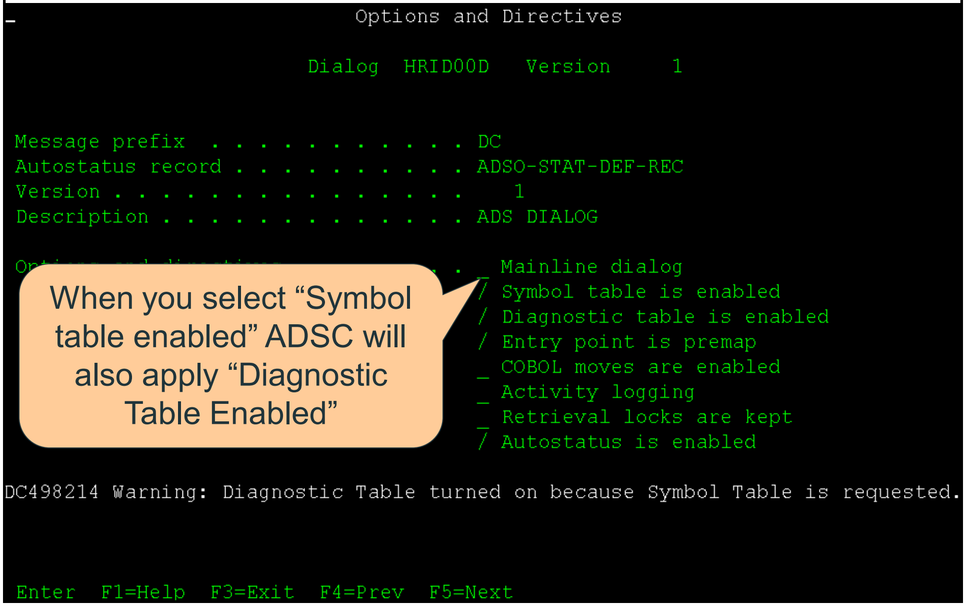
Scenario 1: the Loop
1) Turn on SYMBOL TABLES 1 of 2



```
   Add  Modify  Compile  Delete  Display  Switch
._____________________________________________________________________________.

                          CA-ADS Online Dialog Compiler

                      Computer Associates Int


   Dialog name . . . . . . . .    HRID00D
   Dialog version  . . . . . .       1
   Dictionary name . . . . . .    EDUDIC
   Dictionary node . . . . . .    ________

   Screen  . . . . . . . . . .    1  1. General options
                                     2. Assign maps
                                     3. Assign database
                                     4. Assign records and tables
                                     5. Assign process modules

DC498105 Dialog HRID00D ver    1 is not checked out. Select MODIFY to Checkout.

 Command ===> _
 Enter  F1=Help  F3=Exit  F10=Action                                       12
```

Select “General Options” so we can turn on “Symbol Tables”

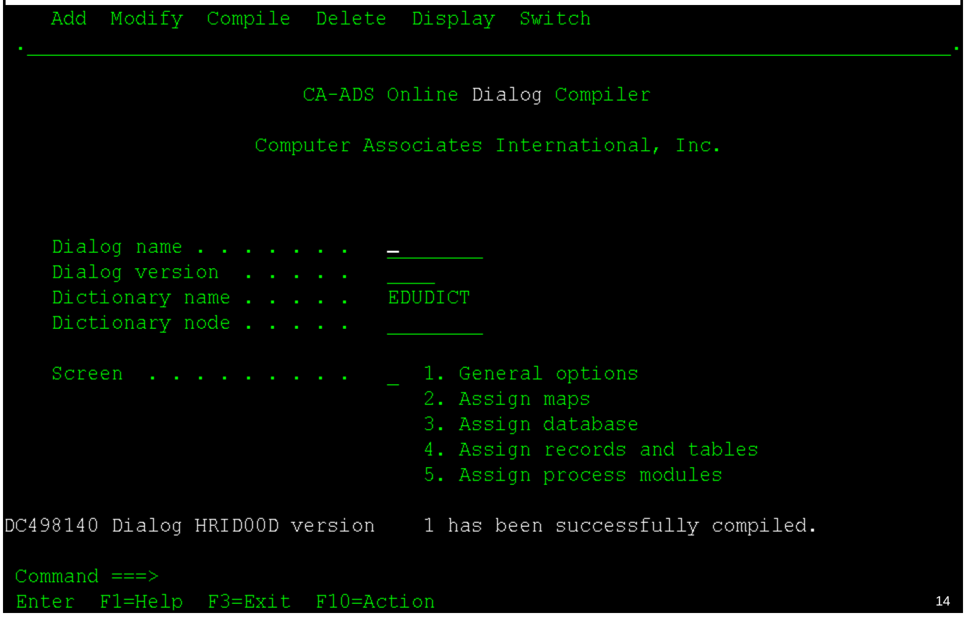
Scenario 1: the Loop
1) Turn on SYMBOL TABLES 2 of 2



```
                              Options and Directives

                          Dialog  HRIDOOD   Version     1


Message prefix  . . . . . . . . . . . . . DC
Autostatus record . . . . . . . . . . . . ADSO-STAT-DEF-REC
Version . . . . . . . . . . . . . . . . .    1
Description . . . . . . . . . . . . . . . ADS DIALOG

Options and directives . . . . . . . . . _ Mainline dialog
                                         / Symbol table is enabled
                                         / Diagnostic table is enabled
                                         / Entry point is premap
                                         _ COBOL moves are enabled
                                         _ Activity logging
                                         _ Retrieval locks are kept
                                         / Autostatus is enabled

DC498214 Warning: Diagnostic Table turned on because Symbol Table is requested.


 Enter  F1=Help  F3=Exit  F4=Prev  F5=Next
```

When you select "Symbol table enabled" ADSC will also apply "Diagnostic Table Enabled"

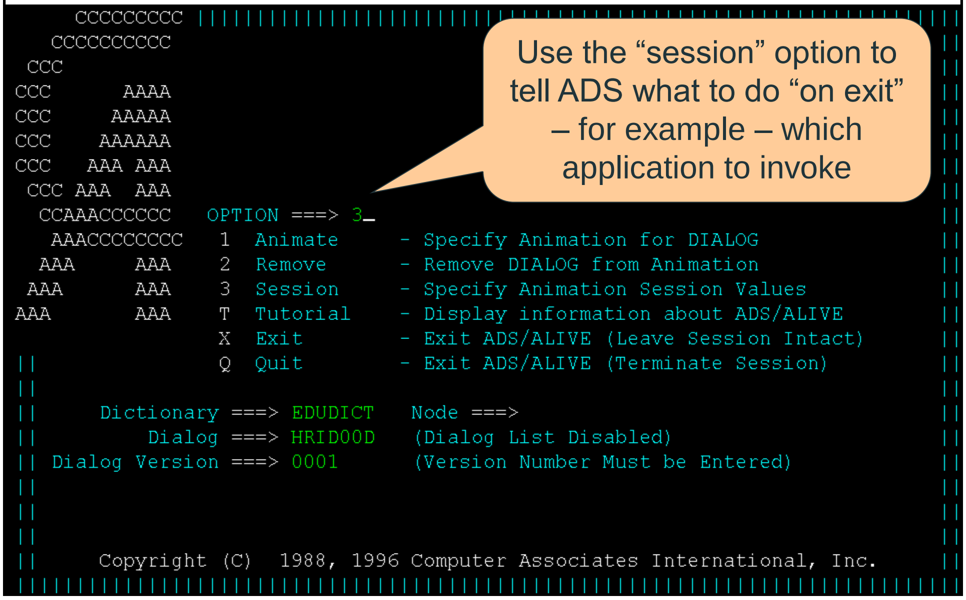
Scenario 1: the Loop
1) Turn on SYMBOL TABLES 3 of 3



```
   Add  Modify  Compile  Delete  Display  Switch
 .____________________________________________________________________________.

                         CA-ADS Online Dialog Compiler

                     Computer Associates International, Inc.


   Dialog name . . . . . . . .   ________
   Dialog version  . . . . . .   ____
   Dictionary name . . . . . .   EDUDICT
   Dictionary node . . . . . .   ________

   Screen  . . . . . . . . . .   _  1. General options
                                    2. Assign maps
                                    3. Assign database
                                    4. Assign records and tables
                                    5. Assign process modules

DC498140 Dialog HRIDOOD version    1 has been successfully compiled.

 Command ===>
 Enter  F1=Help  F3=Exit  F10=Action
```

14

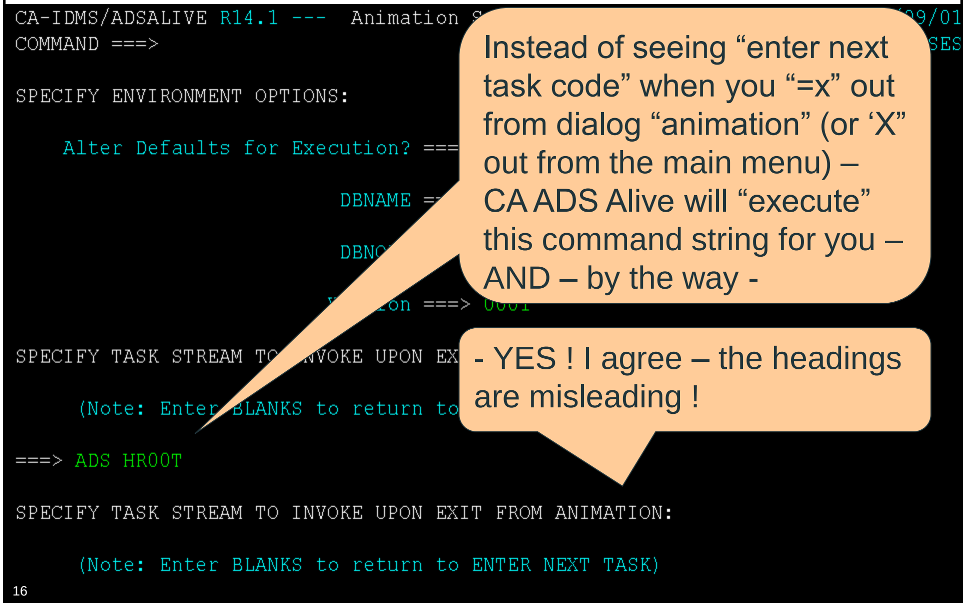
Scenario 1: the Loop
2) Animate the dialog with CA ADS Alive 1 of 8



```
     CCCCCCCCC |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
   CCCCCCCCCC                                                                     ||
 CCC                                                                              ||
CCC       AAAA                                                                    ||
CCC      AAAAA                                                                    ||
CCC     AAAAAA                                                                    ||
CCC    AAA AAA                                                                    ||
 CCC AAA  AAA                                                                     ||
  CCAAACCCCCC    OPTION ===> 3_                                                   ||
   AAACCCCCCCC    1  Animate      - Specify Animation for DIALOG                  ||
  AAA      AAA    2  Remove       - Remove DIALOG from Animation                  ||
 AAA       AAA    3  Session      - Specify Animation Session Values              ||
AAA        AAA    T  Tutorial     - Display information about ADS/ALIVE           ||
                  X  Exit         - Exit ADS/ALIVE (Leave Session Intact)         ||
||                Q  Quit         - Exit ADS/ALIVE (Terminate Session)            ||
||                                                                                ||
||     Dictionary ===> EDUDICT   Node ===>                                        ||
||         Dialog ===> HRIDOOD   (Dialog List Disabled)                           ||
|| Dialog Version ===> 0001      (Version Number Must be Entered)                 ||
||                                                                                ||
||                                                                                ||
||                                                                                ||
||     Copyright (C)  1988, 1996 Computer Associates International, Inc.          ||
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Use the "session" option to tell ADS what to do "on exit" – for example – which application to invoke

Scenario 1: the Loop
2) Animate the dialog with CA ADS Alive 2 of 8

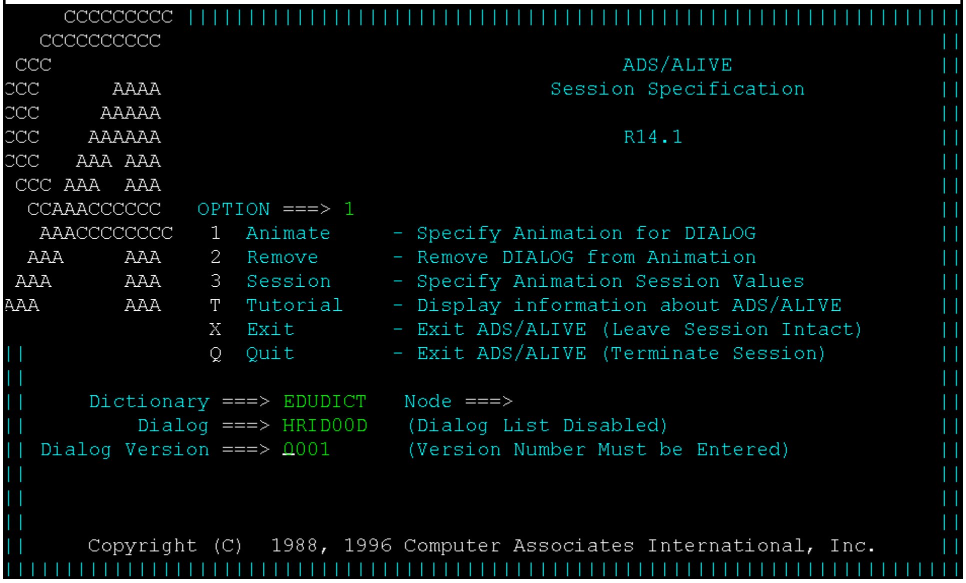


```
CA-IDMS/ADSALIVE R14.1 ---  Animation S                                09/01
COMMAND ===>                                                             SES

SPECIFY ENVIRONMENT OPTIONS:

    Alter Defaults for Execution? ===

                       DBNAME ==

                       DBNO

                      V   ion ===> 0001

SPECIFY TASK STREAM TO INVOKE UPON EX

     (Note: Enter BLANKS to return to

===> ADS HROOT

SPECIFY TASK STREAM TO INVOKE UPON EXIT FROM ANIMATION:

     (Note: Enter BLANKS to return to ENTER NEXT TASK)
16
```

Instead of seeing "enter next task code" when you "=x" out from dialog "animation" (or 'X" out from the main menu) – CA ADS Alive will "execute" this command string for you – AND – by the way -

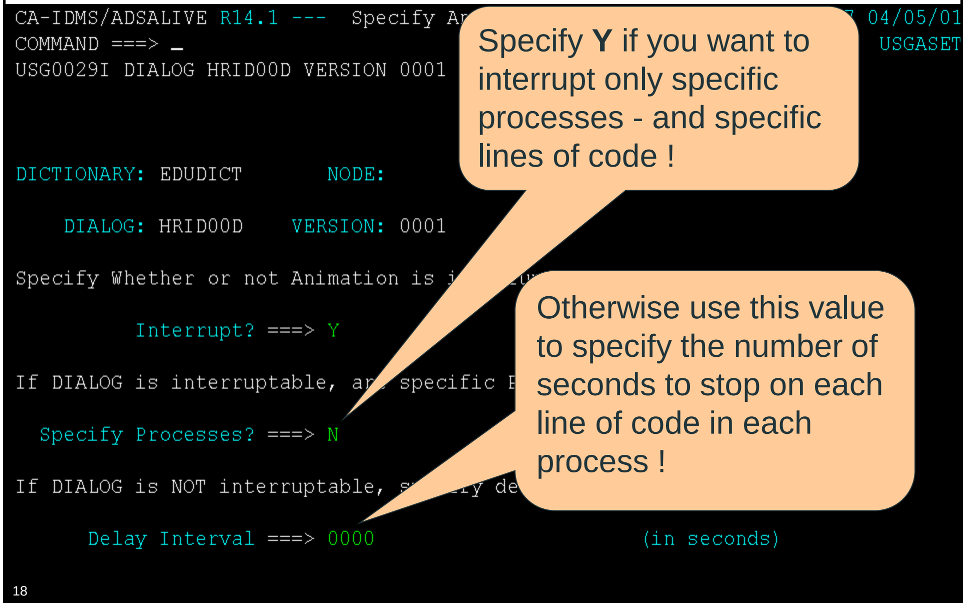
- YES ! I agree – the headings are misleading !

Scenario 1: the Loop
2) Animate the dialog with CA ADS Alive 3 of 8



```
    CCCCCCCCCC ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
  CCCCCCCCCCC                                                              ||
 CCC                                          ADS/ALIVE                    ||
CCC       AAAA                          Session Specification              ||
CCC      AAAAA                                                             ||
CCC     AAAAAA                                R14.1                        ||
CCC    AAA AAA                                                             ||
 CCC AAA   AAA                                                             ||
  CCAAACCCCCCC   OPTION ===> 1                                             ||
   AAACCCCCCCCC   1  Animate     - Specify Animation for DIALOG            ||
  AAA      AAA    2  Remove      - Remove DIALOG from Animation            ||
 AAA       AAA    3  Session     - Specify Animation Session Values        ||
AAA        AAA    T  Tutorial    - Display information about ADS/ALIVE     ||
                  X  Exit        - Exit ADS/ALIVE (Leave Session Intact)   ||
||                Q  Quit        - Exit ADS/ALIVE (Terminate Session)      ||
||                                                                         ||
||      Dictionary ===> EDUDICT    Node ===>                               ||
||          Dialog ===> HRID00D    (Dialog List Disabled)                  ||
|| Dialog Version ===> 0001        (Version Number Must be Entered)        ||
||                                                                         ||
||                                                                         ||
||                                                                         ||
||      Copyright (C)  1988, 1996 Computer Associates International, Inc.  ||
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Scenario 1: the Loop
2) Animate the dialog with CA ADS Alive 4 of 8

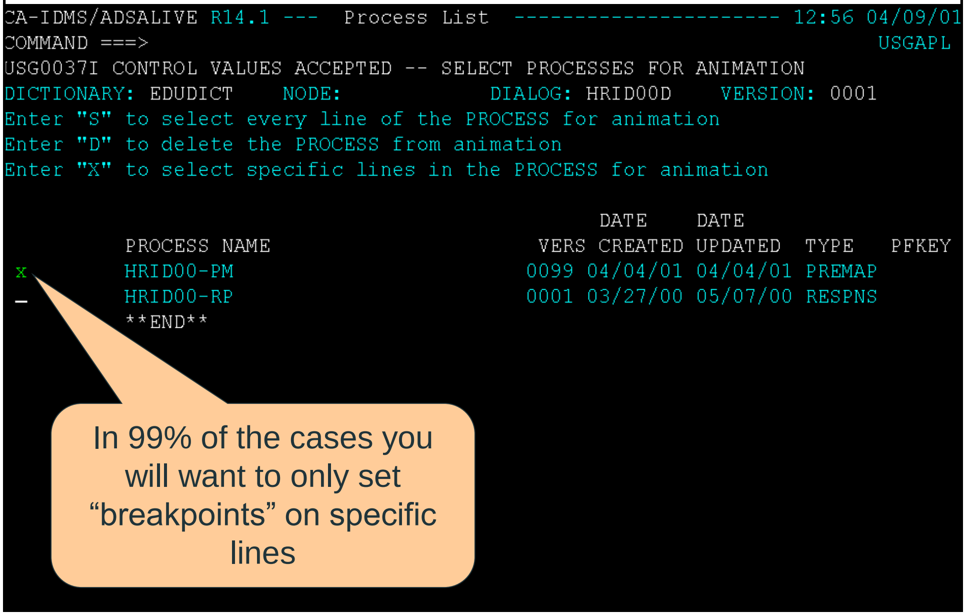


```
CA-IDMS/ADSALIVE R14.1 ---  Specify An                            7 04/05/01
COMMAND ===> _                                                       USGASET
USG0029I DIALOG HRID00D VERSION 0001


DICTIONARY: EDUDICT       NODE:

    DIALOG: HRID00D    VERSION: 0001

Specify Whether or not Animation is i   lu

          Interrupt? ===> Y

If DIALOG is interruptable, an  specific P

  Specify Processes? ===> N

If DIALOG is NOT interruptable, s    ify de

      Delay Interval ===> 0000                        (in seconds)

18
```

Specify **Y** if you want to interrupt only specific processes - and specific lines of code !

Otherwise use this value to specify the number of seconds to stop on each line of code in each process !

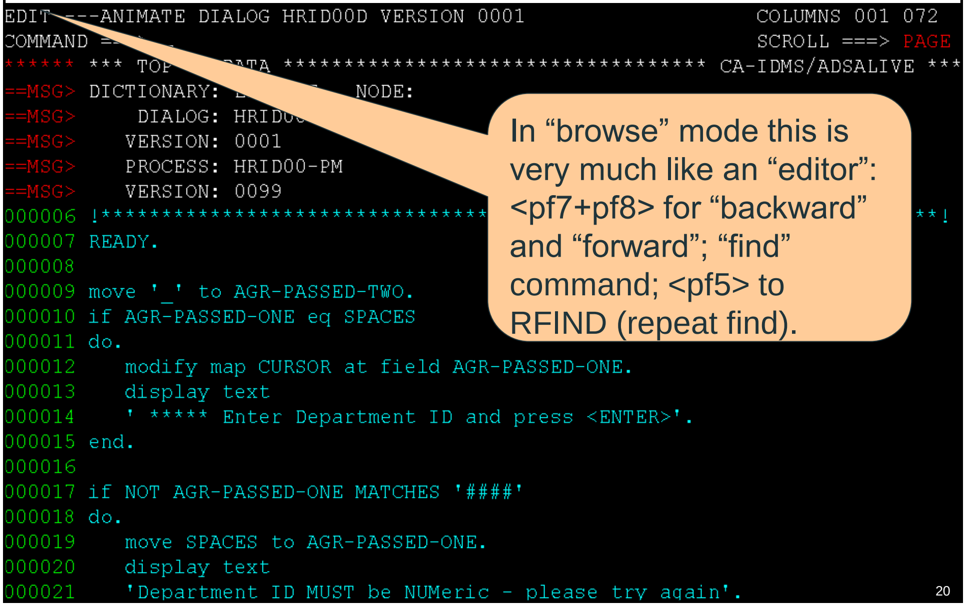
Scenario 1: the Loop
2) Animate the dialog with CA ADS Alive 5 of 8



```
CA-IDMS/ADSALIVE R14.1 ---  Process List  ---------------------- 12:56 04/09/01
COMMAND ===>                                                             USGAPL
USG0037I CONTROL VALUES ACCEPTED -- SELECT PROCESSES FOR ANIMATION
DICTIONARY: EDUDICT    NODE:             DIALOG: HRID00D    VERSION: 0001
Enter "S" to select every line of the PROCESS for animation
Enter "D" to delete the PROCESS from animation
Enter "X" to select specific lines in the PROCESS for animation

                                                 DATE     DATE
          PROCESS NAME                      VERS CREATED UPDATED  TYPE   PFKEY
 x        HRID00-PM                         0099 04/04/01 04/04/01 PREMAP
 _        HRID00-RP                         0001 03/27/00 05/07/00 RESPNS
          **END**
```

In 99% of the cases you will want to only set “breakpoints” on specific lines

Scenario 1: the Loop
2) Animate the dialog with CA ADS Alive 6 of 8

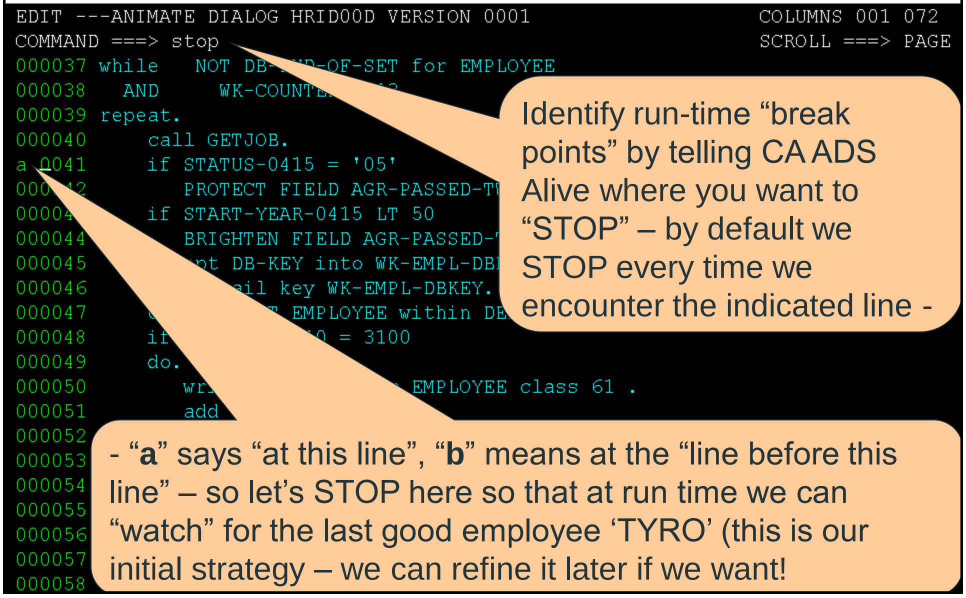


```
EDIT ---ANIMATE DIALOG HRID00D VERSION 0001                     COLUMNS 001 072
COMMAND ==                                                      SCROLL ===> PAGE
****** *** TOP    DATA ************************************ CA-IDMS/ADSALIVE ***
==MSG> DICTIONARY: E          NODE:
==MSG>     DIALOG: HRIDO
==MSG>    VERSION: 0001
==MSG>    PROCESS: HRID00-PM
==MSG>    VERSION: 0099
 !*********************************                                        **!
 READY.

 move '_' to AGR-PASSED-TWO.
 if AGR-PASSED-ONE eq SPACES
 do.
    modify map CURSOR at field AGR-PASSED-ONE.
    display text
    ' ***** Enter Department ID and press <ENTER>'.
 end.

 if NOT AGR-PASSED-ONE MATCHES '####'
 do.
    move SPACES to AGR-PASSED-ONE.
    display text
    'Department ID MUST be NUMeric - please try again'.
```

In “browse” mode this is very much like an “editor”: <pf7+pf8> for “backward” and “forward”; “find” command; <pf5> to RFIND (repeat find).

Scenario 1: the Loop

2) Animate the dialog with CA ADS Alive 7 of 8



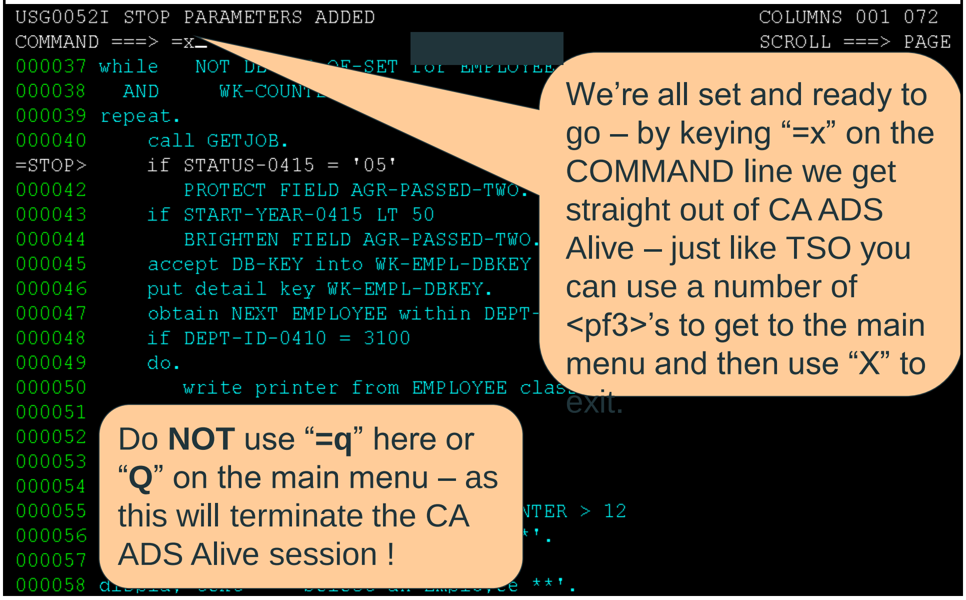
```
EDIT ---ANIMATE DIALOG HRIDOOD VERSION 0001                 COLUMNS 001 072
COMMAND ===> stop                                           SCROLL ===> PAGE
000037 while   NOT DB-   D-OF-SET for EMPLOYEE
000038   AND     WK-COUNTE
000039 repeat.
000040     call GETJOB.
a 0041     if STATUS-0415 = '05'
000 42        PROTECT FIELD AGR-PASSED-T
00004      if START-YEAR-0415 LT 50
000044        BRIGHTEN FIELD AGR-PASSED-
000045         pt DB-KEY into WK-EMPL-DB
000046          ail key WK-EMPL-DBKEY.
000047             T EMPLOYEE within DE
000048     if           0 = 3100
000049     do.
000050        wr             EMPLOYEE class 61 .
000051        add
000052
000053
000054
000055
000056
000057
000058
```

Identify run-time “break points” by telling CA ADS Alive where you want to “STOP” – by default we STOP every time we encounter the indicated line -

- “**a**” says “at this line”, “**b**” means at the “line before this line” – so let’s STOP here so that at run time we can “watch” for the last good employee ‘TYRO’ (this is our initial strategy – we can refine it later if we want!

Scenario 1: the Loop

2) Animate the dialog with CA ADS Alive 8 of 8

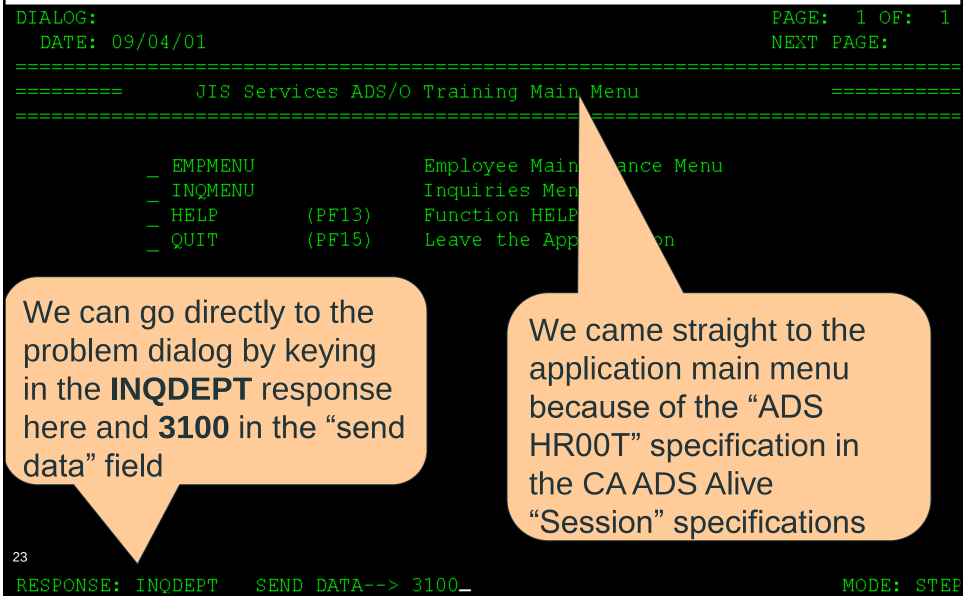


```
USG0052I STOP PARAMETERS ADDED                              COLUMNS 001 072
COMMAND ===> =x_                                            SCROLL ===> PAGE
000037 while   NOT DB      OF-SET for EMPLOYEE
000038   AND     WK-COUNT
000039 repeat.
000040     call GETJOB.
=STOP>     if STATUS-0415 = '05'
000042        PROTECT FIELD AGR-PASSED-TWO.
000043     if START-YEAR-0415 LT 50
000044        BRIGHTEN FIELD AGR-PASSED-TWO.
000045     accept DB-KEY into WK-EMPL-DBKEY
000046     put detail key WK-EMPL-DBKEY.
000047     obtain NEXT EMPLOYEE within DEPT-
000048     if DEPT-ID-0410 = 3100
000049     do.
000050        write printer from EMPLOYEE clas
000051
000052
000053
000054
000055                                     NTER > 12
000056                                     *'.
000057
000058 display                          ee **'.
```

We’re all set and ready to go – by keying “=x” on the COMMAND line we get straight out of CA ADS Alive – just like TSO you can use a number of <pf3>’s to get to the main menu and then use “X” to exit.

Do **NOT** use “**=q**” here or “**Q**” on the main menu – as this will terminate the CA ADS Alive session !

Scenario 1: the Loop
3) Run the application 1 of 18

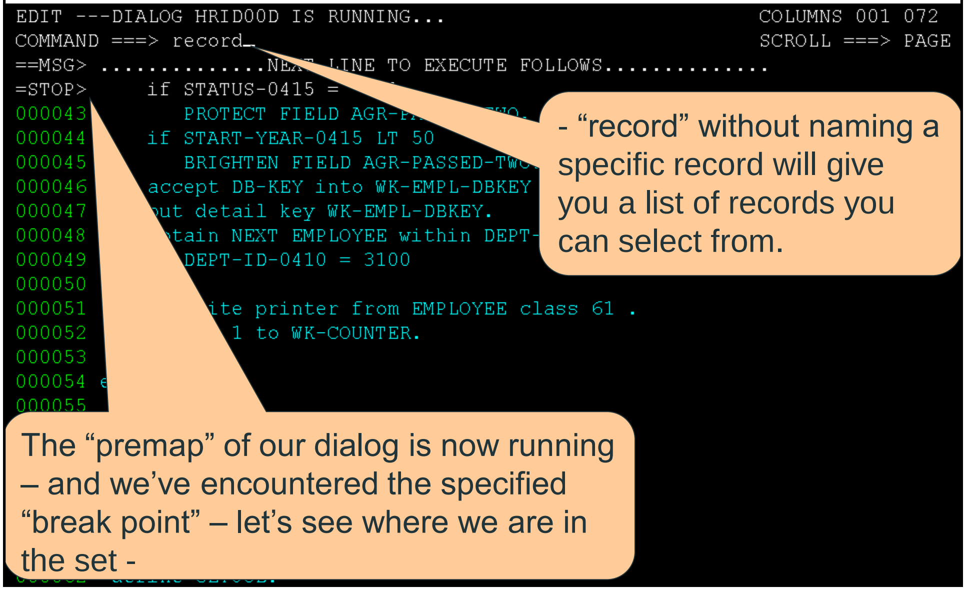


```
DIALOG:                                                              PAGE:   1 OF:   1
  DATE: 09/04/01                                                     NEXT PAGE:
=====================================================================================
=========      JIS Services ADS/O Training Main Menu                      ==========
=====================================================================================

           _ EMPMENU               Employee Main     ance Menu
           _ INQMENU               Inquiries Men
           _ HELP        (PF13)    Function HELP
           _ QUIT        (PF15)    Leave the App         on

23
RESPONSE: INQDEPT   SEND DATA--> 3100_                                      MODE: STEP
```

We can go directly to the problem dialog by keying in the **INQDEPT** response here and **3100** in the "send data" field

We came straight to the application main menu because of the "ADS HR00T" specification in the CA ADS Alive "Session" specifications

Scenario 1: the Loop
3) Run the application 2 of 18

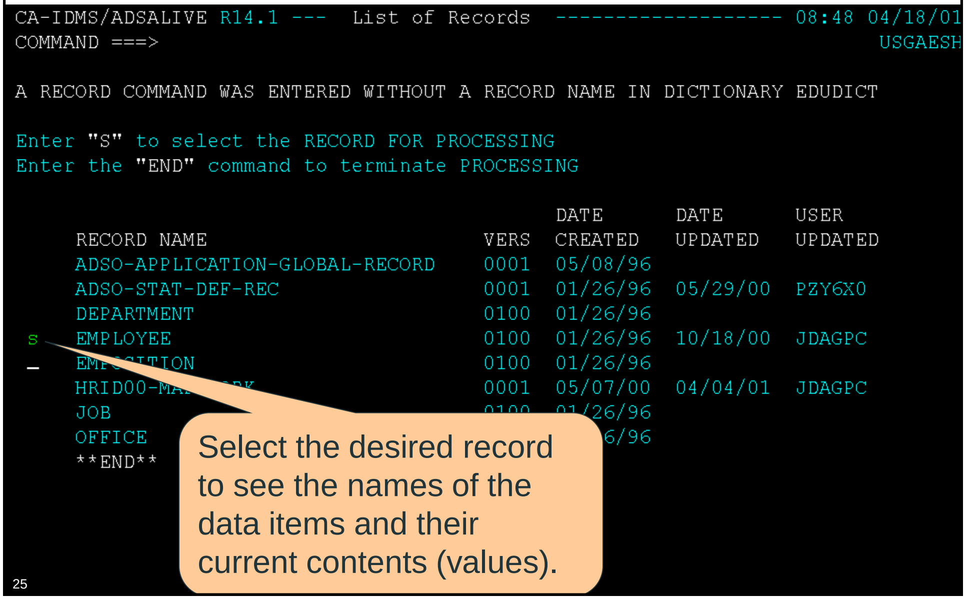


```
EDIT ---DIALOG HRIDOOD IS RUNNING...                              COLUMNS 001 072
COMMAND ===> record_                                              SCROLL ===> PAGE
==MSG> ..............NE   LINE TO EXECUTE FOLLOWS.............
=STOP>     if STATUS-0415 =
000043        PROTECT FIELD AGR-P
000044     if START-YEAR-0415 LT 50
000045        BRIGHTEN FIELD AGR-PASSED-TWO
000046     accept DB-KEY into WK-EMPL-DBKEY
000047     put detail key WK-EMPL-DBKEY.
000048       tain NEXT EMPLOYEE within DEPT-
000049        DEPT-ID-0410 = 3100
000050
000051          ite printer from EMPLOYEE class 61 .
000052            1 to WK-COUNTER.
000053
000054 e
000055
```

- "record" without naming a specific record will give you a list of records you can select from.

The "premap" of our dialog is now running – and we've encountered the specified "break point" – let's see where we are in the set -

Scenario 1: the Loop
3) Run the application 3 of 18

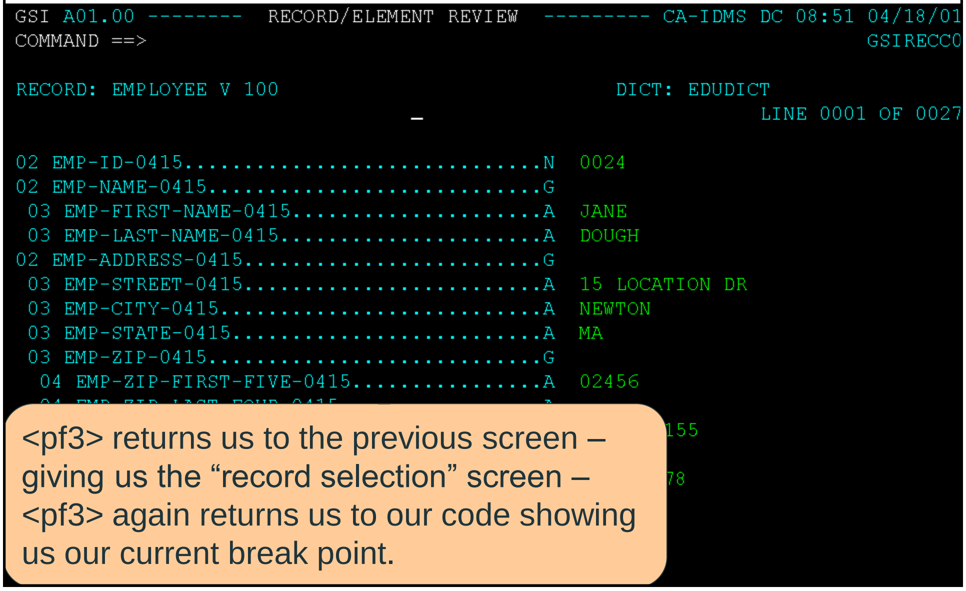


```
CA-IDMS/ADSALIVE R14.1 ---   List of Records  ------------------- 08:48 04/18/01
COMMAND ===>                                                              USGAESH

A RECORD COMMAND WAS ENTERED WITHOUT A RECORD NAME IN DICTIONARY EDUDICT

Enter "S" to select the RECORD FOR PROCESSING
Enter the "END" command to terminate PROCESSING

                                                    DATE      DATE      USER
     RECORD NAME                              VERS  CREATED   UPDATED   UPDATED
     ADSO-APPLICATION-GLOBAL-RECORD           0001  05/08/96
     ADSO-STAT-DEF-REC                        0001  01/26/96  05/29/00  PZY6X0
     DEPARTMENT                               0100  01/26/96
 s   EMPLOYEE                                 0100  01/26/96  10/18/00  JDAGPC
 _   EMP    ITION                             0100  01/26/96
     HRID00-MA     RK                         0001  05/07/00  04/04/01  JDAGPC
     JOB                                      0100  01/26/96
     OFFICE                                          6/96
     **END**
```

Select the desired record to see the names of the data items and their current contents (values).

25

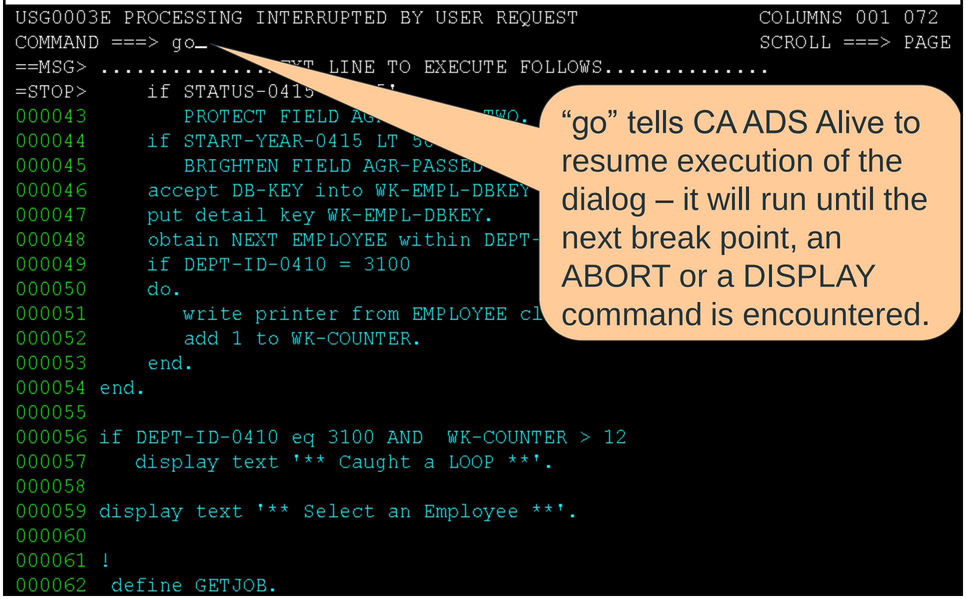
Scenario 1: the Loop
3) Run the application 4 of 18



```
GSI A01.00 --------   RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 08:51 04/18/01
COMMAND ==>                                                             GSIRECC0

RECORD: EMPLOYEE V 100                          DICT: EDUDICT
                                    _                     LINE 0001 OF 0027

02 EMP-ID-0415.............................N  0024
02 EMP-NAME-0415...........................G
 03 EMP-FIRST-NAME-0415....................A  JANE
 03 EMP-LAST-NAME-0415.....................A  DOUGH
02 EMP-ADDRESS-0415........................G
 03 EMP-STREET-0415........................A  15 LOCATION DR
 03 EMP-CITY-0415..........................A  NEWTON
 03 EMP-STATE-0415.........................A  MA
 03 EMP-ZIP-0415...........................G
  04 EMP-ZIP-FIRST-FIVE-0415...............A  02456
  04 EMP-ZIP-LAST-FOUR-0415................A
                                                 155

                                                78
```

<pf3> returns us to the previous screen – giving us the “record selection” screen – <pf3> again returns us to our code showing us our current break point.

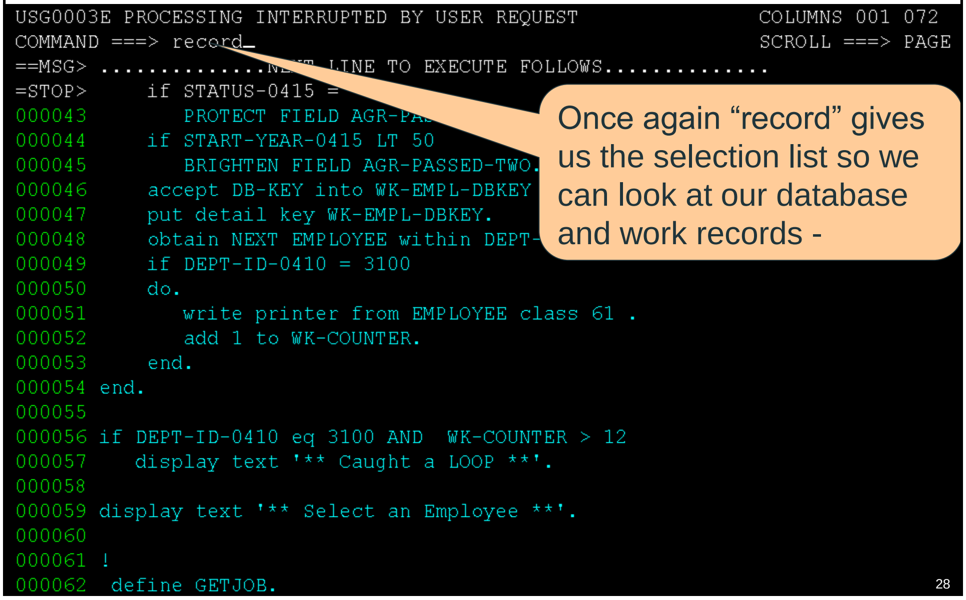
Scenario 1: the Loop
3) Run the application 5 of 18



```
USG0003E PROCESSING INTERRUPTED BY USER REQUEST                 COLUMNS 001 072
COMMAND ===> go_                                                 SCROLL ===> PAGE
==MSG> ..............EXT LINE TO EXECUTE FOLLOWS.............
=STOP>     if STATUS-0415
              PROTECT FIELD AG                  TWO.
           if START-YEAR-0415 LT 50
              BRIGHTEN FIELD AGR-PASSED
           accept DB-KEY into WK-EMPL-DBKEY
           put detail key WK-EMPL-DBKEY.
           obtain NEXT EMPLOYEE within DEPT-
           if DEPT-ID-0410 = 3100
           do.
              write printer from EMPLOYEE cl
              add 1 to WK-COUNTER.
           end.
       end.

       if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
          display text '** Caught a LOOP **'.

       display text '** Select an Employee **'.

       !
        define GETJOB.
```

"go" tells CA ADS Alive to resume execution of the dialog – it will run until the next break point, an ABORT or a DISPLAY command is encountered.

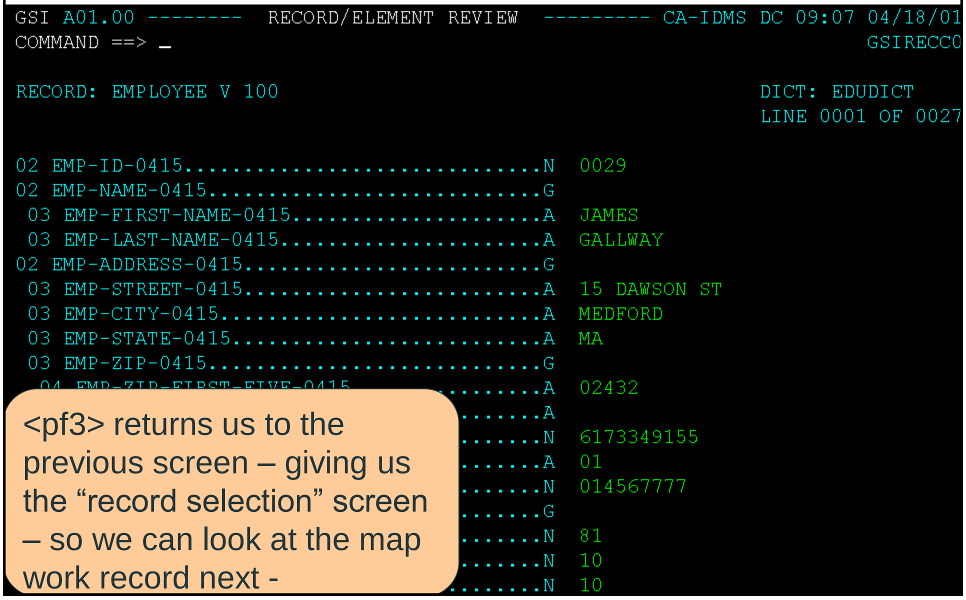
Scenario 1: the Loop
3) Run the application 6 of 18



```
USG0003E PROCESSING INTERRUPTED BY USER REQUEST                 COLUMNS 001 072
COMMAND ===> record_                                             SCROLL ===> PAGE
==MSG> ..............NE T LINE TO EXECUTE FOLLOWS.............
=STOP>     if STATUS-0415 =
              PROTECT FIELD AGR-PA
           if START-YEAR-0415 LT 50
              BRIGHTEN FIELD AGR-PASSED-TWO.
           accept DB-KEY into WK-EMPL-DBKEY
           put detail key WK-EMPL-DBKEY.
           obtain NEXT EMPLOYEE within DEPT-
           if DEPT-ID-0410 = 3100
           do.
              write printer from EMPLOYEE class 61 .
              add 1 to WK-COUNTER.
           end.
       end.

       if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
          display text '** Caught a LOOP **'.

       display text '** Select an Employee **'.

       !
        define GETJOB.
```

Once again "record" gives us the selection list so we can look at our database and work records -

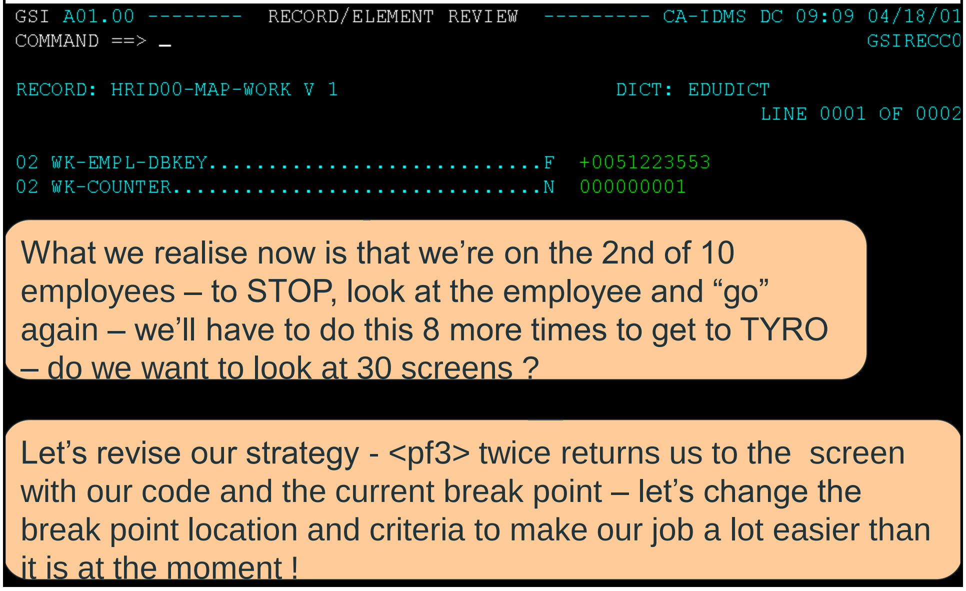
Scenario 1: the Loop
3) Run the application   7 of 18



```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 09:07 04/18/01
COMMAND ==> _                                                          GSIRECC0

RECORD: EMPLOYEE V 100                                          DICT: EDUDICT
                                                                LINE 0001 OF 0027

02 EMP-ID-0415.............................N  0029
02 EMP-NAME-0415...........................G
 03 EMP-FIRST-NAME-0415....................A  JAMES
 03 EMP-LAST-NAME-0415.....................A  GALLWAY
02 EMP-ADDRESS-0415........................G
 03 EMP-STREET-0415........................A  15 DAWSON ST
 03 EMP-CITY-0415..........................A  MEDFORD
 03 EMP-STATE-0415.........................A  MA
 03 EMP-ZIP-0415...........................G
  04 EMP-ZIP-FIRST-FIVE-0415...............A  02432
                                    .......A
                                    .......N  6173349155
                                    .......A  01
                                    .......N  014567777
                                    .......G
                                    .......N  81
                                    .......N  10
                                    .......N  10
```

<pf3> returns us to the previous screen – giving us the “record selection” screen – so we can look at the map work record next -

Scenario 1: the Loop
3) Run the application 8 of 18

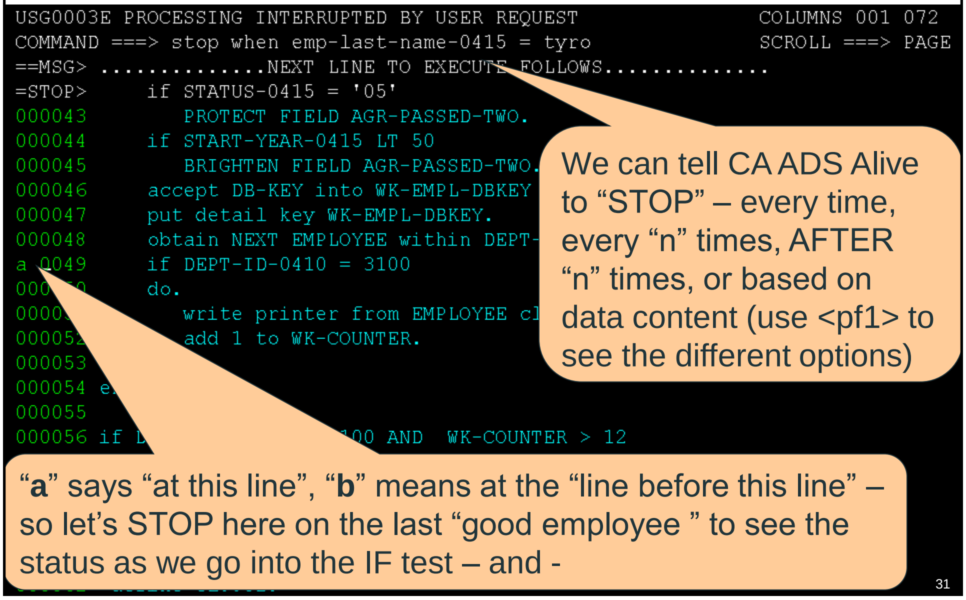


```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 09:09 04/18/01
COMMAND ==> _                                                          GSIRECC0

RECORD: HRID00-MAP-WORK V 1                              DICT: EDUDICT
                                                                LINE 0001 OF 0002

02 WK-EMPL-DBKEY...........................F  +0051223553
02 WK-COUNTER..............................N  000000001
```

What we realise now is that we’re on the 2nd of 10 employees – to STOP, look at the employee and “go” again – we’ll have to do this 8 more times to get to TYRO – do we want to look at 30 screens ?

Let’s revise our strategy - <pf3> twice returns us to the screen with our code and the current break point – let’s change the break point location and criteria to make our job a lot easier than it is at the moment !

Scenario 1: the Loop
3) Run the application 9 of 18



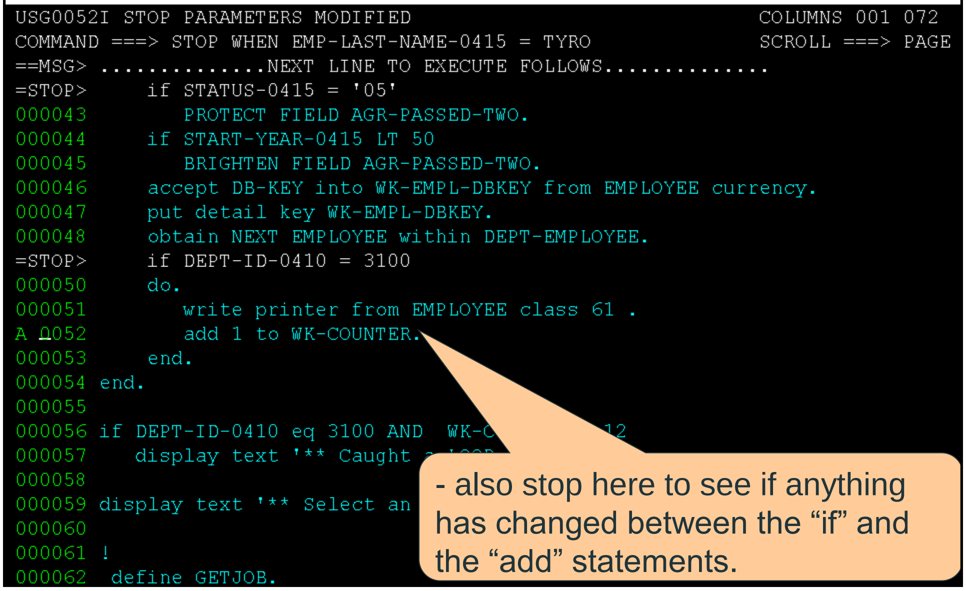
```
USG0003E PROCESSING INTERRUPTED BY USER REQUEST              COLUMNS 001 072
COMMAND ===> stop when emp-last-name-0415 = tyro             SCROLL ===> PAGE
==MSG> ..............NEXT LINE TO EXECUTE FOLLOWS.............
=STOP>     if STATUS-0415 = '05'
000043        PROTECT FIELD AGR-PASSED-TWO.
000044     if START-YEAR-0415 LT 50
000045        BRIGHTEN FIELD AGR-PASSED-TWO.
000046     accept DB-KEY into WK-EMPL-DBKEY
000047     put detail key WK-EMPL-DBKEY.
000048     obtain NEXT EMPLOYEE within DEPT-
a 0049     if DEPT-ID-0410 = 3100
000        do.
0000          write printer from EMPLOYEE cl
000052        add 1 to WK-COUNTER.
000053
000054 e
000055
000056 if D                       00 AND  WK-COUNTER > 12
```

We can tell CA ADS Alive to “STOP” – every time, every “n” times, AFTER “n” times, or based on data content (use <pf1> to see the different options)

“**a**” says “at this line”, “**b**” means at the “line before this line” – so let’s STOP here on the last “good employee ” to see the status as we go into the IF test – and -

31

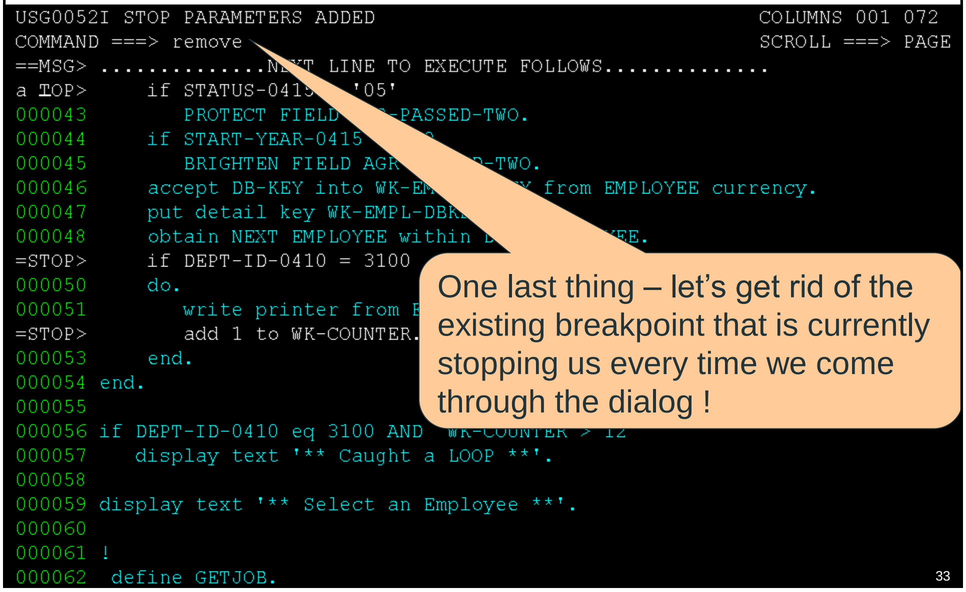
Scenario 1: the Loop
3) Run the application 10 of 18



```
USG0052I STOP PARAMETERS MODIFIED                            COLUMNS 001 072
COMMAND ===> STOP WHEN EMP-LAST-NAME-0415 = TYRO             SCROLL ===> PAGE
==MSG> ..............NEXT LINE TO EXECUTE FOLLOWS.............
=STOP>     if STATUS-0415 = '05'
000043        PROTECT FIELD AGR-PASSED-TWO.
000044     if START-YEAR-0415 LT 50
000045        BRIGHTEN FIELD AGR-PASSED-TWO.
000046     accept DB-KEY into WK-EMPL-DBKEY from EMPLOYEE currency.
000047     put detail key WK-EMPL-DBKEY.
000048     obtain NEXT EMPLOYEE within DEPT-EMPLOYEE.
=STOP>     if DEPT-ID-0410 = 3100
000050     do.
000051        write printer from EMPLOYEE class 61 .
A 0052        add 1 to WK-COUNTER.
000053     end.
000054 end.
000055
000056 if DEPT-ID-0410 eq 3100 AND  WK-C          12
000057    display text '** Caught
000058
000059 display text '** Select an
000060
000061 !
000062  define GETJOB.
```

- also stop here to see if anything has changed between the “if” and the “add” statements.

Scenario 1: the Loop
3) Run the application 11 of 18

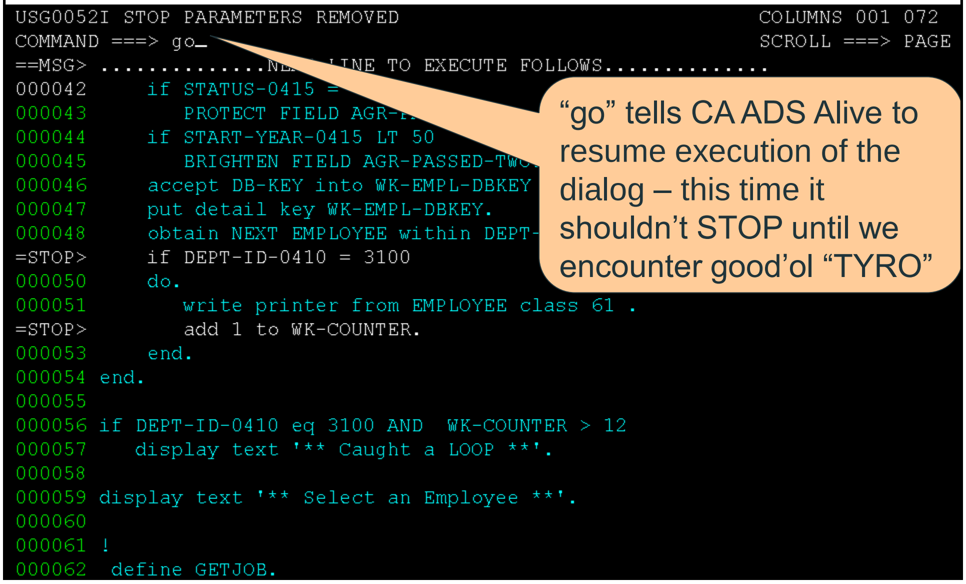


```
USG0052I STOP PARAMETERS ADDED                                  COLUMNS 001 072
COMMAND ===> remove                                             SCROLL ===> PAGE
==MSG> ..............NEXT LINE TO EXECUTE FOLLOWS..............
a TOP>     if STATUS-0415    '05'
000043        PROTECT FIELD    -PASSED-TWO.
000044     if START-YEAR-0415
000045        BRIGHTEN FIELD AGR      D-TWO.
000046     accept DB-KEY into WK-EM        from EMPLOYEE currency.
000047     put detail key WK-EMPL-DBK
000048     obtain NEXT EMPLOYEE within             EE.
=STOP>     if DEPT-ID-0410 = 3100
000050     do.
000051        write printer from E
=STOP>        add 1 to WK-COUNTER.
000053     end.
000054 end.
000055
000056 if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
000057    display text '** Caught a LOOP **'.
000058
000059 display text '** Select an Employee **'.
000060
000061 !
000062  define GETJOB.
```

One last thing – let's get rid of the existing breakpoint that is currently stopping us every time we come through the dialog !

33

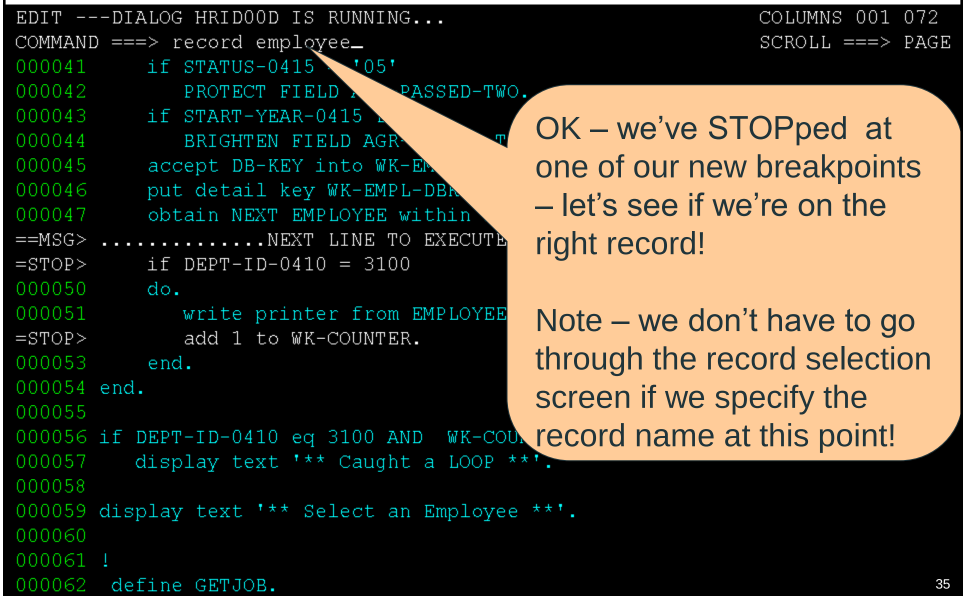
Scenario 1: the Loop
3) Run the application 12 of 18



```
USG0052I STOP PARAMETERS REMOVED                                COLUMNS 001 072
COMMAND ===> go_                                                SCROLL ===> PAGE
==MSG> ..............NE   LINE TO EXECUTE FOLLOWS..............
000042     if STATUS-0415 =
000043        PROTECT FIELD AGR-
000044     if START-YEAR-0415 LT 50
000045        BRIGHTEN FIELD AGR-PASSED-TWO.
000046     accept DB-KEY into WK-EMPL-DBKEY
000047     put detail key WK-EMPL-DBKEY.
000048     obtain NEXT EMPLOYEE within DEPT-
=STOP>     if DEPT-ID-0410 = 3100
000050     do.
000051        write printer from EMPLOYEE class 61 .
=STOP>        add 1 to WK-COUNTER.
000053     end.
000054 end.
000055
000056 if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
000057    display text '** Caught a LOOP **'.
000058
000059 display text '** Select an Employee **'.
000060
000061 !
000062  define GETJOB.
```

"go" tells CA ADS Alive to resume execution of the dialog – this time it shouldn't STOP until we encounter good'ol "TYRO"

Scenario 1: the Loop
3) Run the application 13 of 18



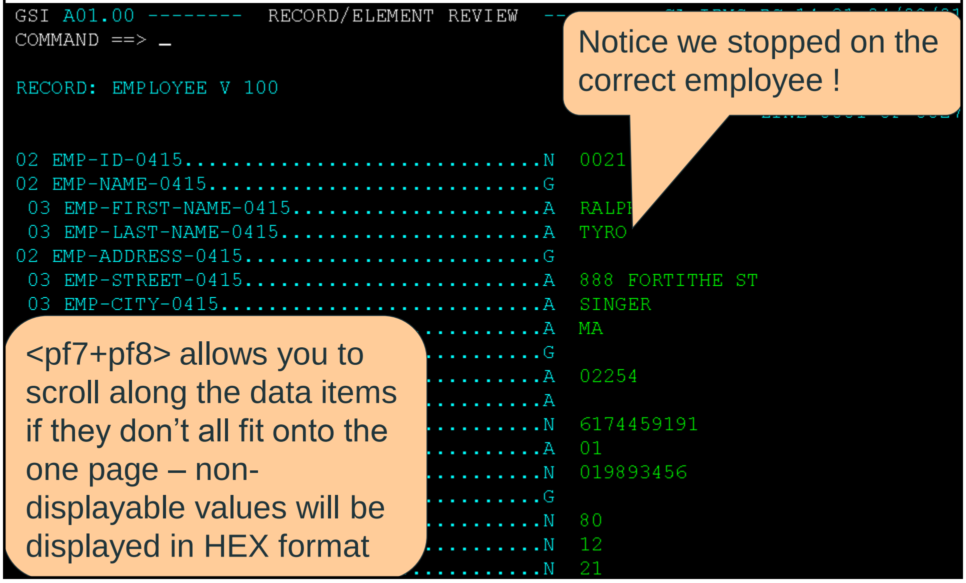
```
EDIT ---DIALOG HRIDOOD IS RUNNING...                             COLUMNS 001 072
COMMAND ===> record employee_                                    SCROLL ===> PAGE
000041     if STATUS-0415   '05'
000042        PROTECT FIELD    PASSED-TWO.
000043     if START-YEAR-0415
000044        BRIGHTEN FIELD AGR
000045     accept DB-KEY into WK-EM
000046     put detail key WK-EMPL-DBK
000047     obtain NEXT EMPLOYEE within
==MSG> ..............NEXT LINE TO EXECUTE
=STOP>     if DEPT-ID-0410 = 3100
000050     do.
000051        write printer from EMPLOYEE
=STOP>        add 1 to WK-COUNTER.
000053     end.
000054 end.
000055
000056 if DEPT-ID-0410 eq 3100 AND  WK-COU
000057    display text '** Caught a LOOP **'.
000058
000059 display text '** Select an Employee **'.
000060
000061 !
000062  define GETJOB.
```

35

OK – we've STOPped at one of our new breakpoints – let's see if we're on the right record!

Note – we don't have to go through the record selection screen if we specify the record name at this point!

Scenario 1: the Loop
3) Run the application 14 of 18

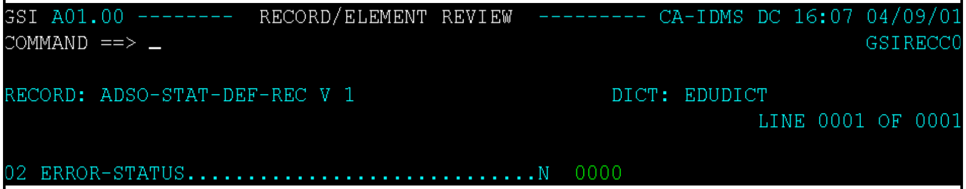


```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --
COMMAND ==> _

RECORD: EMPLOYEE V 100


02 EMP-ID-0415..............................N  0021
02 EMP-NAME-0415............................G
 03 EMP-FIRST-NAME-0415.....................A  RALPH
 03 EMP-LAST-NAME-0415......................A  TYRO
02 EMP-ADDRESS-0415.........................G
 03 EMP-STREET-0415.........................A  888 FORTITHE ST
 03 EMP-CITY-0415...........................A  SINGER
                                  ..........A  MA
                                  ..........G
                                  ..........A  02254
                                  ..........A
                                  ..........N  6174459191
                                  ..........A  01
                                  ..........N  019893456
                                  ..........G
                                  ..........N  80
                                  ..........N  12
                                  ..........N  21
```

Notice we stopped on the correct employee !

<pf7+pf8> allows you to scroll along the data items if they don't all fit onto the one page – non-displayable values will be displayed in HEX format

Scenario 1: the Loop
3) Run the application   15 of 18

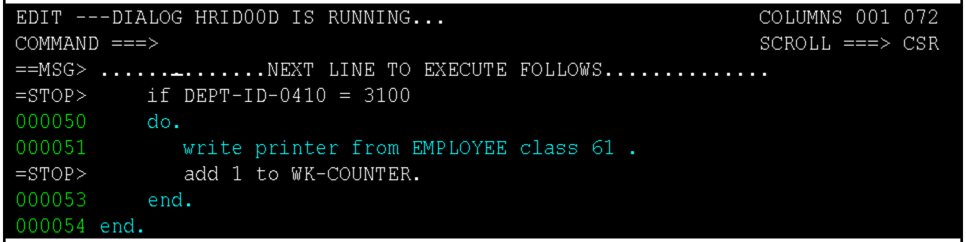


```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 16:07 04/09/01
COMMAND ==> _                                                   GSIRECC0

RECORD: ADSO-STAT-DEF-REC V 1                   DICT: EDUDICT
                                                   LINE 0001 OF 0001

02 ERROR-STATUS.............................N  0000
```

Our situation:

We've looked through the first 10 Employees in the set – we are currently sitting on the 10th employee and are about to do the WRITE PRINTER and add 1 to our counter. All is well so we return to the display of the lines of code (press <pf3> to "end" the record display and say "go" -

Scenario 1: the Loop
3) Run the application   16 of 18



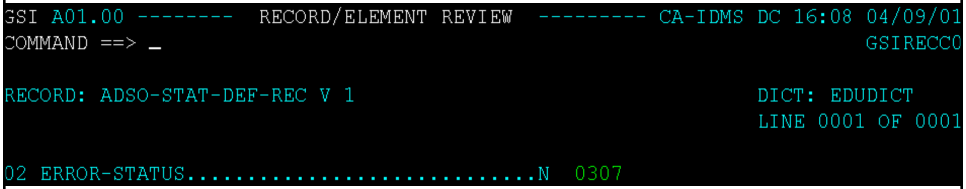
```
EDIT ---DIALOG HRID00D IS RUNNING...                        COLUMNS 001 072
COMMAND ===>                                                SCROLL ===> CSR
==MSG> .............NEXT LINE TO EXECUTE FOLLOWS.............
=STOP>     if DEPT-ID-0410 = 3100
000050     do.
000051        write printer from EMPLOYEE class 61 .
=STOP>        add 1 to WK-COUNTER.
000053     end.
000054  end.
```

We're on the "IF" test – so let's see what's happened to the ERROR-STATUS and EMPLOYEE record contents

Type "record" on the command line to get a list of records – then in turn select ADSO-STAT-DEF-REC and EMPLOYEE – here's what we see -

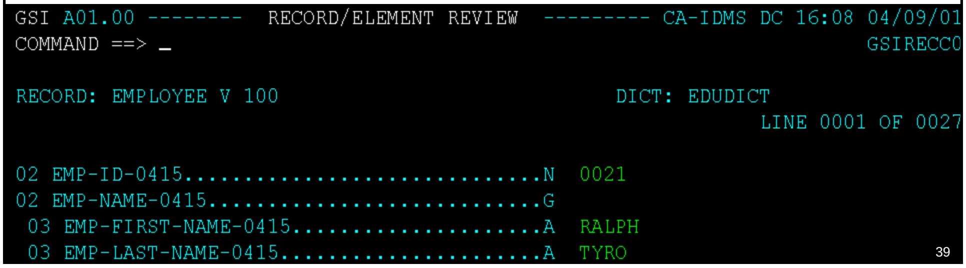
Scenario 1: the Loop

3) Run the application 17 of 18



```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 16:08 04/09/01
COMMAND ==> _                                                          GSIRECC0

RECORD: ADSO-STAT-DEF-REC V 1                                 DICT: EDUDICT
                                                              LINE 0001 OF 0001

02 ERROR-STATUS.............................N  0307
```

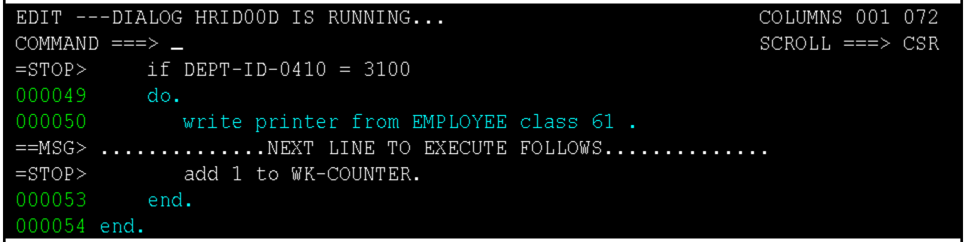
We've got our DB-END-OF-SET in ERROR-STATUS, and we still have the same EMPLOYEE record current (currency doesn't change on a non-zero error status) – so let's "GO" again -



```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 16:08 04/09/01
COMMAND ==> _                                                          GSIRECC0

RECORD: EMPLOYEE V 100                               DICT: EDUDICT
                                                              LINE 0001 OF 0027

02 EMP-ID-0415.............................N  0021
02 EMP-NAME-0415...........................G
 03 EMP-FIRST-NAME-0415....................A  RALPH
 03 EMP-LAST-NAME-0415.....................A  TYRO
```

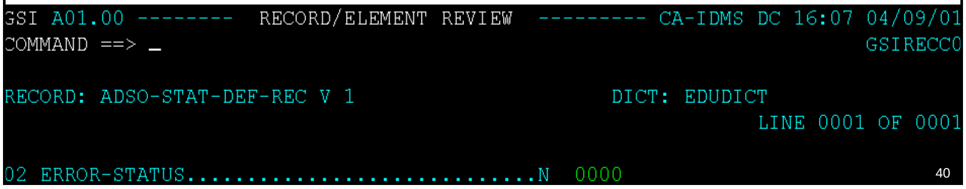
Scenario 1: the Loop

3) Run the application 18 of 18



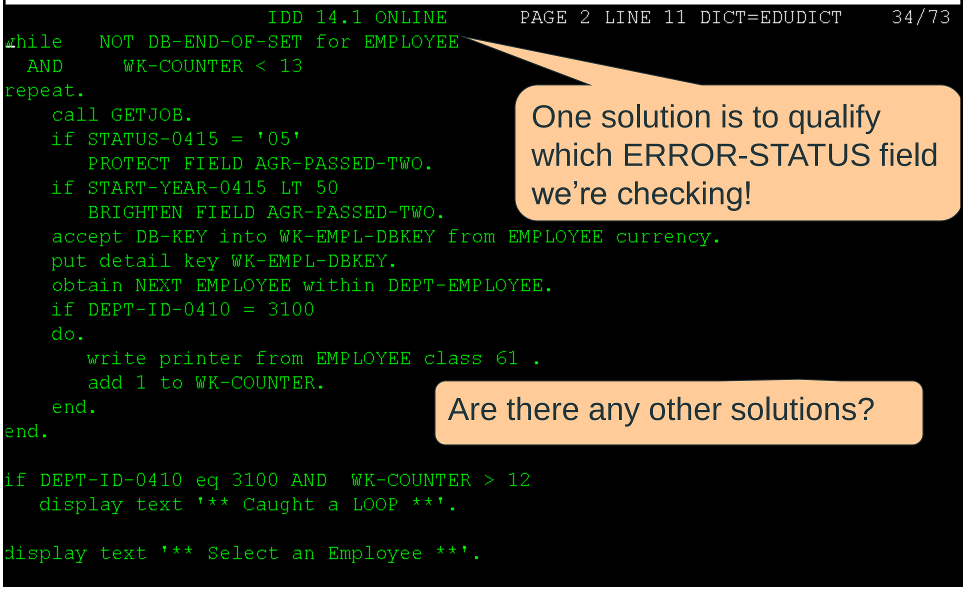
```
EDIT ---DIALOG HRID0OD IS RUNNING...                          COLUMNS 001 072
COMMAND ===> _                                                SCROLL ===> CSR
=STOP>     if DEPT-ID-0410 = 3100
000049     do.
000050        write printer from EMPLOYEE class 61 .
==MSG> ..............NEXT LINE TO EXECUTE FOLLOWS.............
=STOP>        add 1 to WK-COUNTER.
000053     end.
000054 end.
```

We're about to increment WK-COUNTER, but when we look at ERROR-STATUS we see that it's now DB-STATUS-OK ('0000' – it was '0307') – so we can see why our loop control on DB-END-OF-SET doesn't work!



```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 16:07 04/09/01
COMMAND ==> _                                                          GSIRECC0

RECORD: ADSO-STAT-DEF-REC V 1                        DICT: EDUDICT
                                                              LINE 0001 OF 0001

02 ERROR-STATUS.............................N  0000
```

Scenario 1: the Loop
4) Fix the problem 1 of 3 – one fix

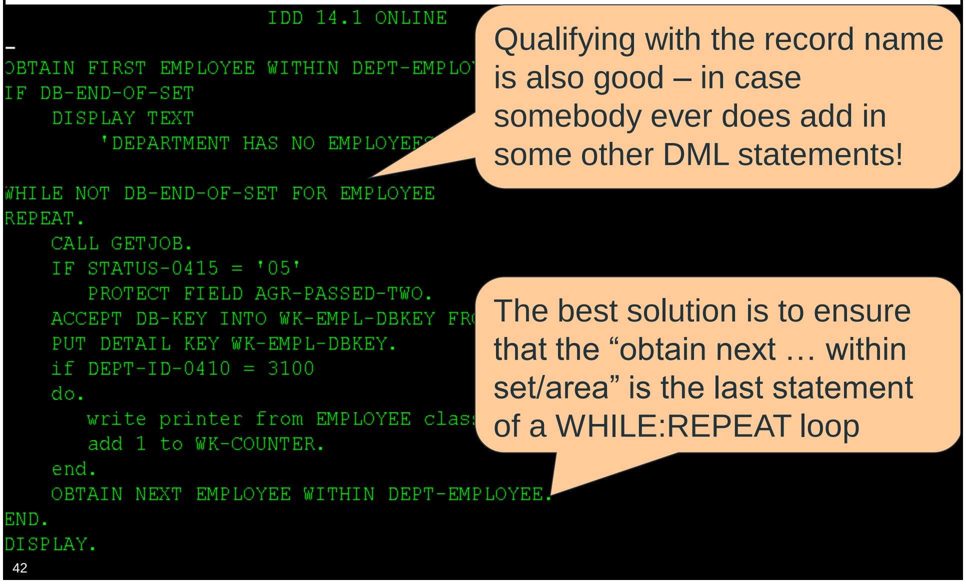


```
                      IDD 14.1 ONLINE      PAGE 2 LINE 11 DICT=EDUDICT    34/73
while   NOT DB-END-OF-SET for EMPLOYEE
  AND     WK-COUNTER < 13
repeat.
    call GETJOB.
    if STATUS-0415 = '05'
       PROTECT FIELD AGR-PASSED-TWO.
    if START-YEAR-0415 LT 50
       BRIGHTEN FIELD AGR-PASSED-TWO.
    accept DB-KEY into WK-EMPL-DBKEY from EMPLOYEE currency.
    put detail key WK-EMPL-DBKEY.
    obtain NEXT EMPLOYEE within DEPT-EMPLOYEE.
    if DEPT-ID-0410 = 3100
    do.
       write printer from EMPLOYEE class 61 .
       add 1 to WK-COUNTER.
    end.
end.

if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
   display text '** Caught a LOOP **'.

display text '** Select an Employee **'.
```

One solution is to qualify which ERROR-STATUS field we're checking!

Are there any other solutions?

Scenario 1: the Loop
4) Fix the problem 2 of 3 – another fix



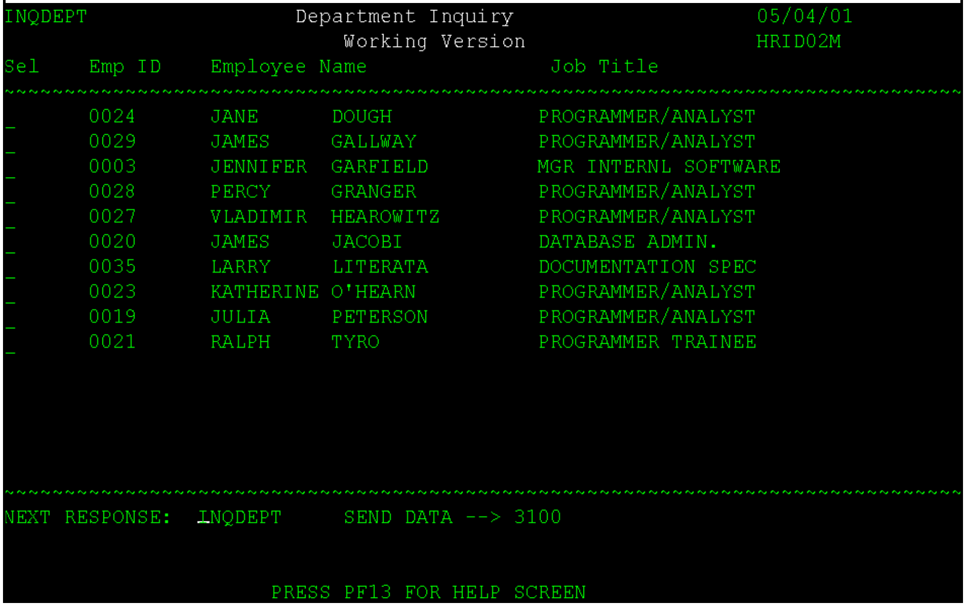
```
                      IDD 14.1 ONLINE
_
OBTAIN FIRST EMPLOYEE WITHIN DEPT-EMPLO
IF DB-END-OF-SET
    DISPLAY TEXT
        'DEPARTMENT HAS NO EMPLOYEE

WHILE NOT DB-END-OF-SET FOR EMPLOYEE
REPEAT.
    CALL GETJOB.
    IF STATUS-0415 = '05'
       PROTECT FIELD AGR-PASSED-TWO.
    ACCEPT DB-KEY INTO WK-EMPL-DBKEY FR
    PUT DETAIL KEY WK-EMPL-DBKEY.
    if DEPT-ID-0410 = 3100
    do.
       write printer from EMPLOYEE clas
       add 1 to WK-COUNTER.
    end.
    OBTAIN NEXT EMPLOYEE WITHIN DEPT-EMPLOYEE.
END.
DISPLAY.
```

Qualifying with the record name is also good – in case somebody ever does add in some other DML statements!

The best solution is to ensure that the "obtain next … within set/area" is the last statement of a WHILE:REPEAT loop

42

Scenario 1: the Loop
4) Fix the problem 3 of 3 – correct output



```
INQDEPT                 Department Inquiry                  05/04/01
                            Working Version                 HRID02M
Sel    Emp ID    Employee Name                Job Title
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
_      0024      JANE      DOUGH            PROGRAMMER/ANALYST
_      0029      JAMES     GALLWAY          PROGRAMMER/ANALYST
_      0003      JENNIFER  GARFIELD         MGR INTERNL SOFTWARE
_      0028      PERCY     GRANGER          PROGRAMMER/ANALYST
_      0027      VLADIMIR  HEAROWITZ        PROGRAMMER/ANALYST
_      0020      JAMES     JACOBI           DATABASE ADMIN.
_      0035      LARRY     LITERATA         DOCUMENTATION SPEC
_      0023      KATHERINE O'HEARN          PROGRAMMER/ANALYST
_      0019      JULIA     PETERSON         PROGRAMMER/ANALYST
_      0021      RALPH     TYRO             PROGRAMMER TRAINEE




~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NEXT RESPONSE:  INQDEPT     SEND DATA --> 3100


                      PRESS PF13 FOR HELP SCREEN
```

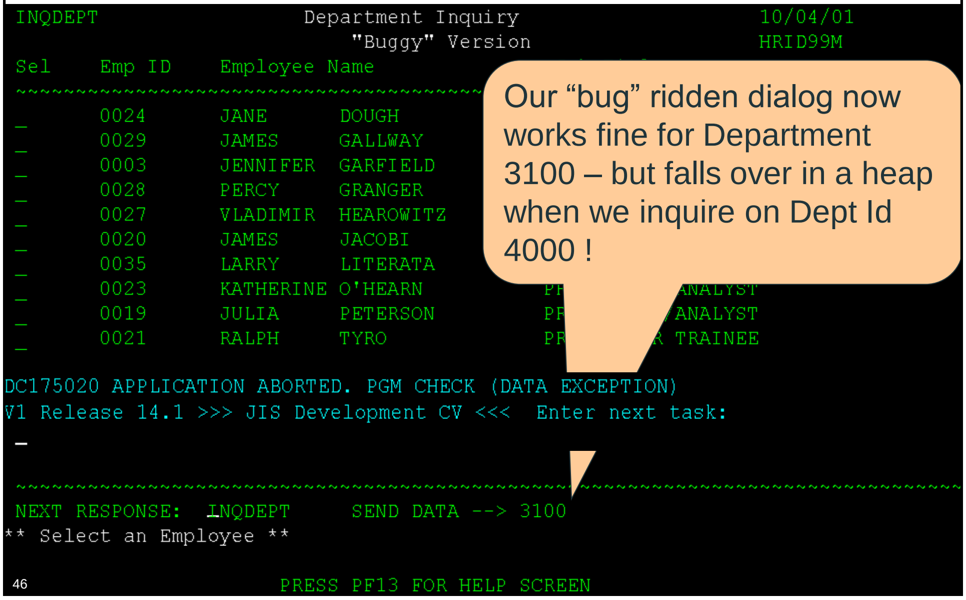
Session Outline

- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary

Scenario 2: “bad” data
Let’s review the steps!

- Ensure the dialog is generated with SYMBOL TABLES ENABLED (already done in this case)
- ANIMATE the dialog with CA ADS Alive
  - Set “breakpoints”
  - Remember to eXit NOT to Quit
- Run the application

Scenario 2: “bad” data
What we should get – and are getting

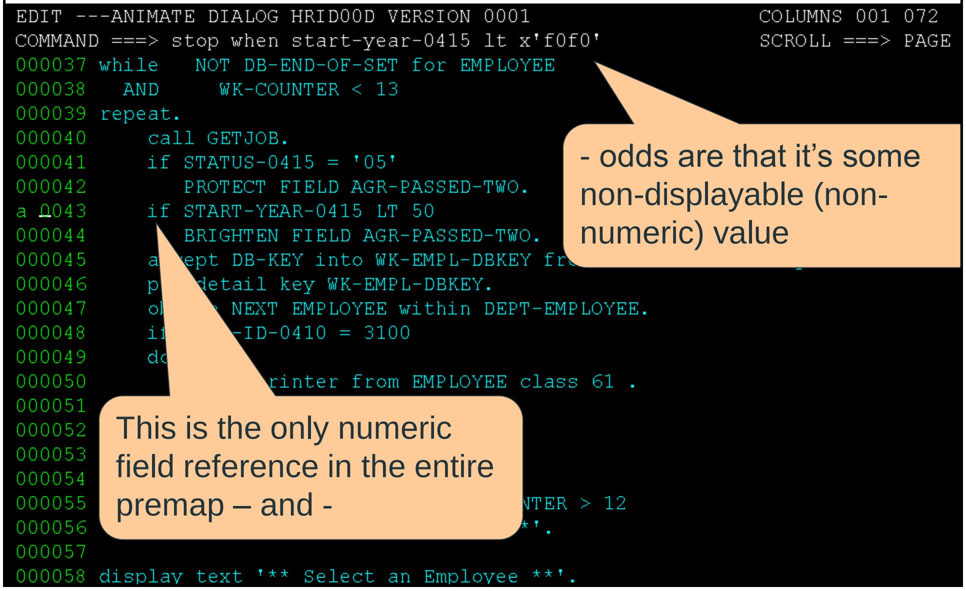


```
INQDEPT                     Department Inquiry                        10/04/01
                                "Buggy" Version                       HRID99M
Sel     Emp ID     Employee Name
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
_       0024       JANE       DOUGH
_       0029       JAMES      GALLWAY
_       0003       JENNIFER   GARFIELD
_       0028       PERCY      GRANGER
_       0027       VLADIMIR   HEAROWITZ
_       0020       JAMES      JACOBI
_       0035       LARRY      LITERATA
_       0023       KATHERINE O'HEARN            PR          ANALYST
_       0019       JULIA      PETERSON           PR         /ANALYST
_       0021       RALPH      TYRO               PR         R TRAINEE
DC175020 APPLICATION ABORTED. PGM CHECK (DATA EXCEPTION)
V1 Release 14.1 >>> JIS Development CV <<<  Enter next task:
 _

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 NEXT RESPONSE:  INQDEPT     SEND DATA --> 3100
** Select an Employee **

                         PRESS PF13 FOR HELP SCREEN
```

Our “bug” ridden dialog now works fine for Department 3100 – but falls over in a heap when we inquire on Dept Id 4000 !

Scenario 2: “bad data”

1) Animate = set the breakpoint(s)



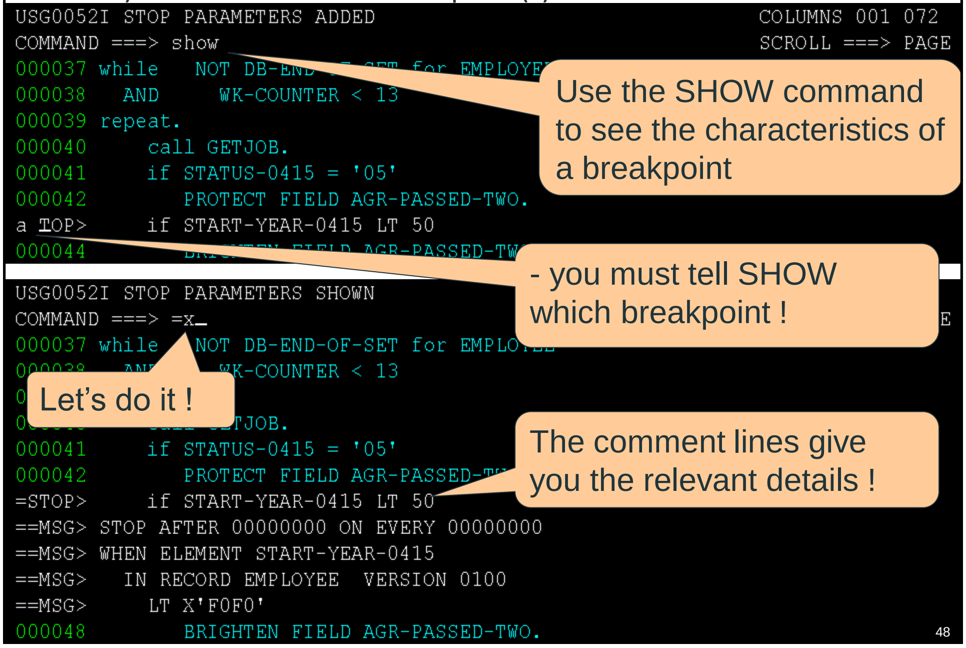
```
EDIT ---ANIMATE DIALOG HRID00D VERSION 0001                COLUMNS 001 072
COMMAND ===> stop when start-year-0415 lt x'f0f0'          SCROLL ===> PAGE
000037 while   NOT DB-END-OF-SET for EMPLOYEE
000038   AND     WK-COUNTER < 13
000039 repeat.
000040     call GETJOB.
000041     if STATUS-0415 = '05'
000042        PROTECT FIELD AGR-PASSED-TWO.
a 0043     if START-YEAR-0415 LT 50
000044        BRIGHTEN FIELD AGR-PASSED-TWO.
000045     a  ept DB-KEY into WK-EMPL-DBKEY fr
000046     p   detail key WK-EMPL-DBKEY.
000047     ob    NEXT EMPLOYEE within DEPT-EMPLOYEE.
000048     if    -ID-0410 = 3100
000049     do
000050              rinter from EMPLOYEE class 61 .
000051
000052
000053
000054
000055                                            NTER > 12
000056                                             *'.
000057
000058 display text '** Select an Employee **'.
```

- odds are that it’s some non-displayable (non-numeric) value

This is the only numeric field reference in the entire premap – and -

Scenario 2: “bad data”

1) Animate = set the breakpoint(s)



```
USG0052I STOP PARAMETERS ADDED                           COLUMNS 001 072
COMMAND ===> show                                        SCROLL ===> PAGE
000037 while   NOT DB-END      for EMPLOYE
000038   AND     WK-COUNTER < 13
000039 repeat.
000040     call GETJOB.
000041     if STATUS-0415 = '05'
000042        PROTECT FIELD AGR-PASSED-TWO.
a TOP>     if START-YEAR-0415 LT 50
000044        BRI    EN FIELD AGR-PASSED-TW
USG0052I STOP PARAMETERS SHOWN                                              E
COMMAND ===> =x_
000037 while   NOT DB-END-OF-SET for EMPLO
000038   AN      WK-COUNTER < 13
0
0            TJOB.
000041     if STATUS-0415 = '05'
000042        PROTECT FIELD AGR-PASSED-T
=STOP>     if START-YEAR-0415 LT 50
==MSG> STOP AFTER 00000000 ON EVERY 00000000
==MSG> WHEN ELEMENT START-YEAR-0415
==MSG>   IN RECORD EMPLOYEE  VERSION 0100
==MSG>     LT X'F0F0'
000048        BRIGHTEN FIELD AGR-PASSED-TWO.
```

Use the SHOW command to see the characteristics of a breakpoint

- you must tell SHOW which breakpoint !

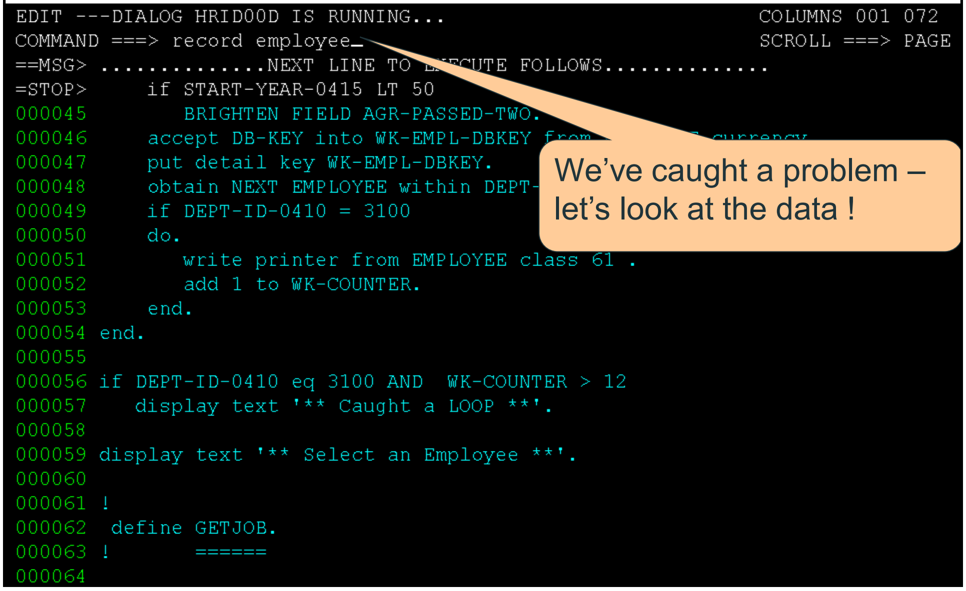
Let’s do it !

The comment lines give you the relevant details !

48

Scenario 2: “bad data”

2) Run the application – encounter the breakpoint

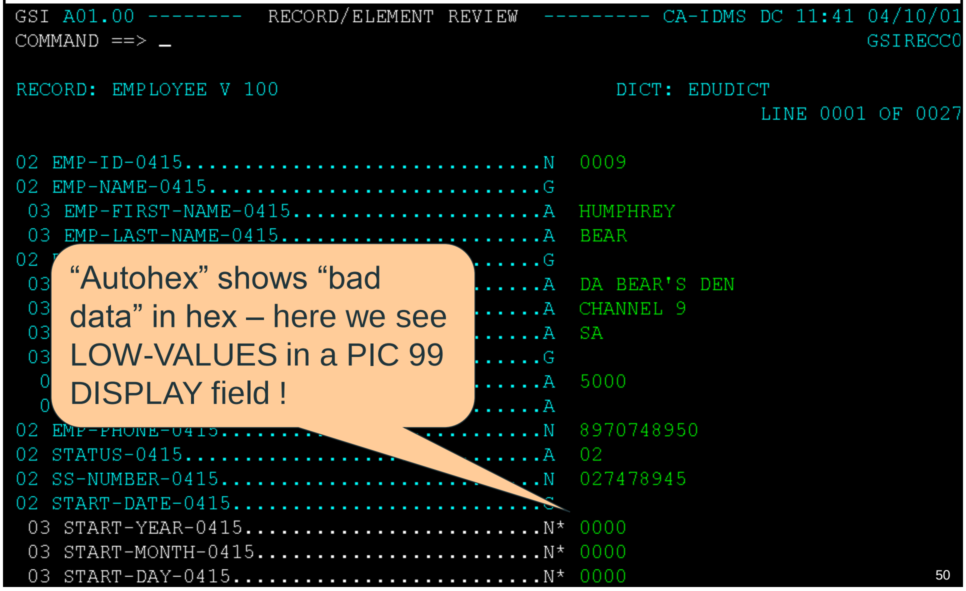


```
EDIT ---DIALOG HRIDOOD IS RUNNING...                                  COLUMNS 001 072
COMMAND ===> record employee_                                         SCROLL ===> PAGE
==MSG> ..............NEXT LINE TO EXECUTE FOLLOWS.............
=STOP>     if START-YEAR-0415 LT 50
              BRIGHTEN FIELD AGR-PASSED-TWO.
           accept DB-KEY into WK-EMPL-DBKEY from
           put detail key WK-EMPL-DBKEY.
           obtain NEXT EMPLOYEE within DEPT-
           if DEPT-ID-0410 = 3100
           do.
              write printer from EMPLOYEE class 61 .
              add 1 to WK-COUNTER.
           end.
       end.

       if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
          display text '** Caught a LOOP **'.

       display text '** Select an Employee **'.

       !
        define GETJOB.
       !       ======

```

We’ve caught a problem – let’s look at the data !

Scenario 2: “bad data”

2) Run the application – examine the data



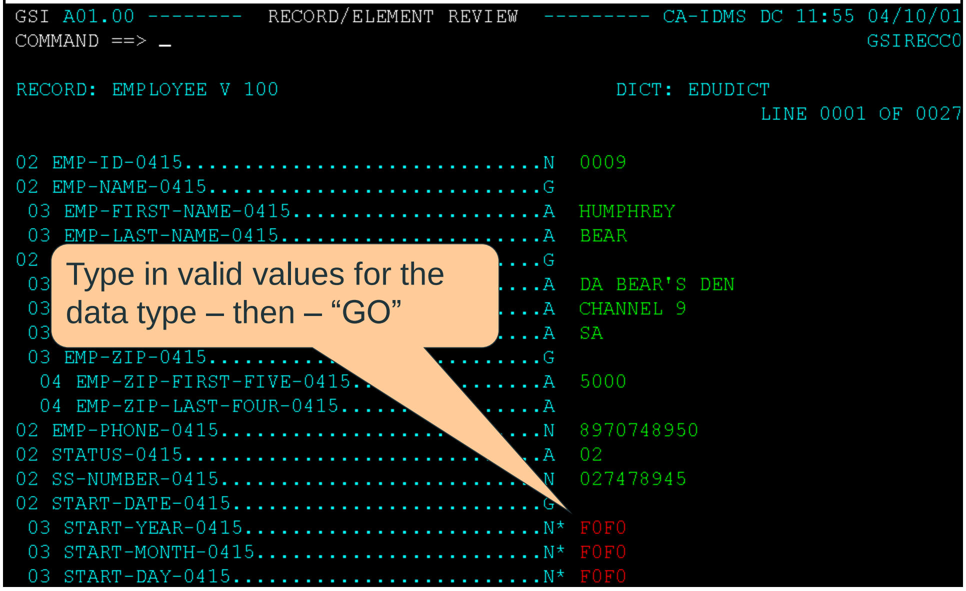
```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 11:41 04/10/01
COMMAND ==> _                                                          GSIRECC0

RECORD: EMPLOYEE V 100                            DICT: EDUDICT
                                                              LINE 0001 OF 0027

02 EMP-ID-0415.............................N  0009
02 EMP-NAME-0415...........................G
 03 EMP-FIRST-NAME-0415....................A  HUMPHREY
 03 EMP-LAST-NAME-0415.....................A  BEAR
02                                    .....G
 03                                   .....A  DA BEAR'S DEN
 03                                   .....A  CHANNEL 9
 03                                   .....A  SA
 03                                   .....G
  0                                   .....A  5000
  0                                   .....A
02 EMP-PHONE-0415..........................N  8970748950
02 STATUS-0415.............................A  02
02 SS-NUMBER-0415..........................N  027478945
02 START-DATE-0415.........................G
 03 START-YEAR-0415........................N* 0000
 03 START-MONTH-0415.......................N* 0000
 03 START-DAY-0415.........................N* 0000
```

“Autohex” shows “bad data” in hex – here we see LOW-VALUES in a PIC 99 DISPLAY field !

50

Scenario 2: “bad data”

2) Run the application – alter data “on the fly”

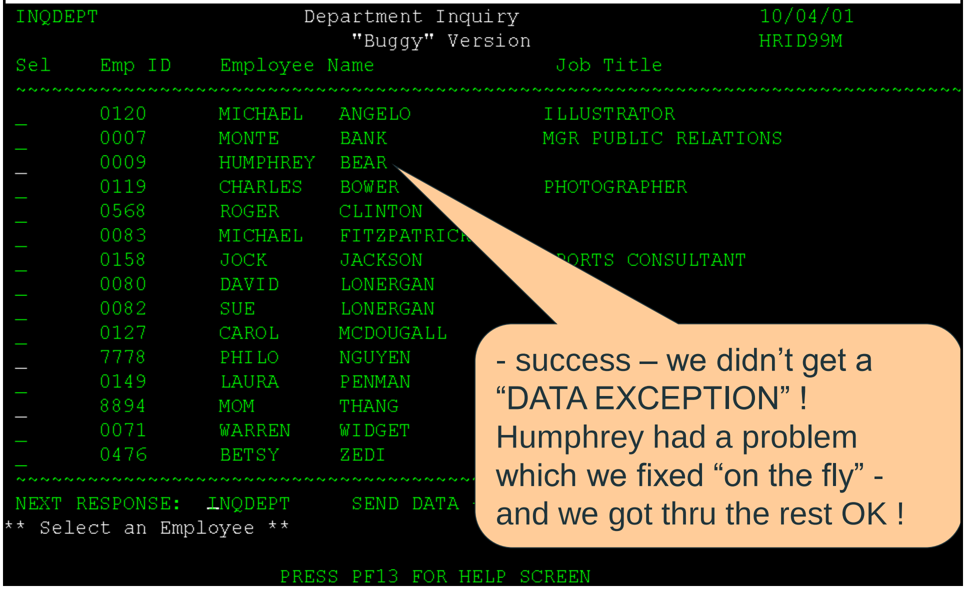


```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 11:55 04/10/01
COMMAND ==> _                                                         GSIRECC0

RECORD: EMPLOYEE V 100                         DICT: EDUDICT
                                                        LINE 0001 OF 0027

02 EMP-ID-0415.............................N  0009
02 EMP-NAME-0415...........................G
 03 EMP-FIRST-NAME-0415....................A  HUMPHREY
 03 EMP-LAST-NAME-0415.....................A  BEAR
02                                     ....G
 03                                    ....A  DA BEAR'S DEN
 03                                    ....A  CHANNEL 9
 03                                    ....A  SA
 03 EMP-ZIP-0415.........        .........G
  04 EMP-ZIP-FIRST-FIVE-0415.      .......A  5000
  04 EMP-ZIP-LAST-FOUR-0415.....     .....A
02 EMP-PHONE-0415.................      ....N  8970748950
02 STATUS-0415.........................     .A  02
02 SS-NUMBER-0415.......................     N  027478945
02 START-DATE-0415.........................G
 03 START-YEAR-0415........................N* F0F0
 03 START-MONTH-0415.......................N* F0F0
 03 START-DAY-0415.........................N* F0F0
```

Type in valid values for the data type – then – “GO”

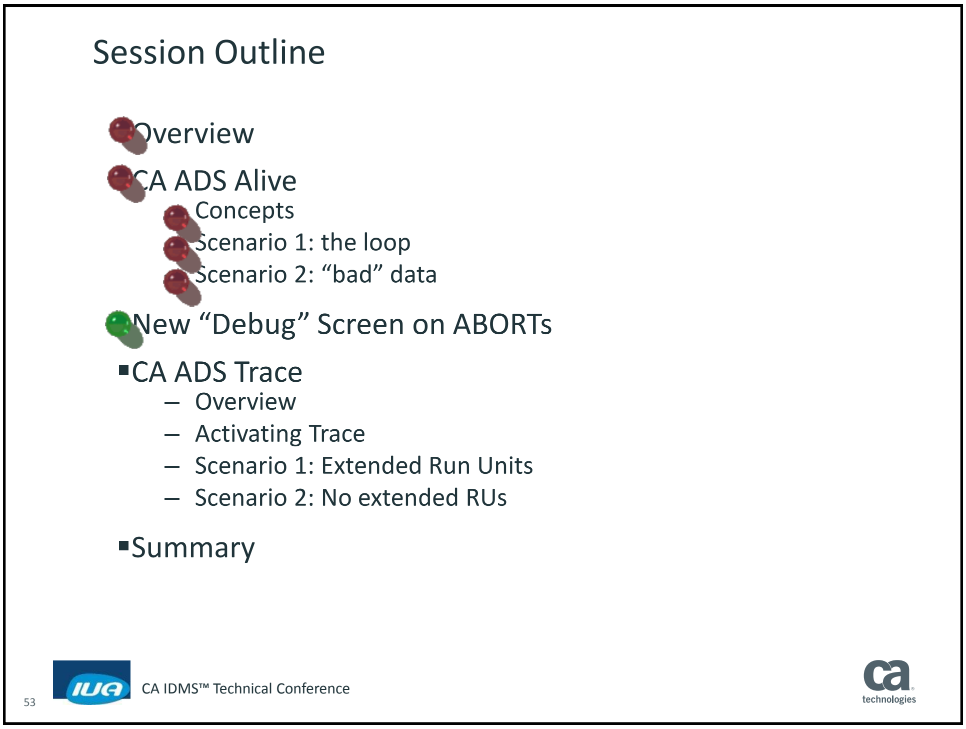
Scenario 2: “bad data”

2) Run the application – our “fix” worked



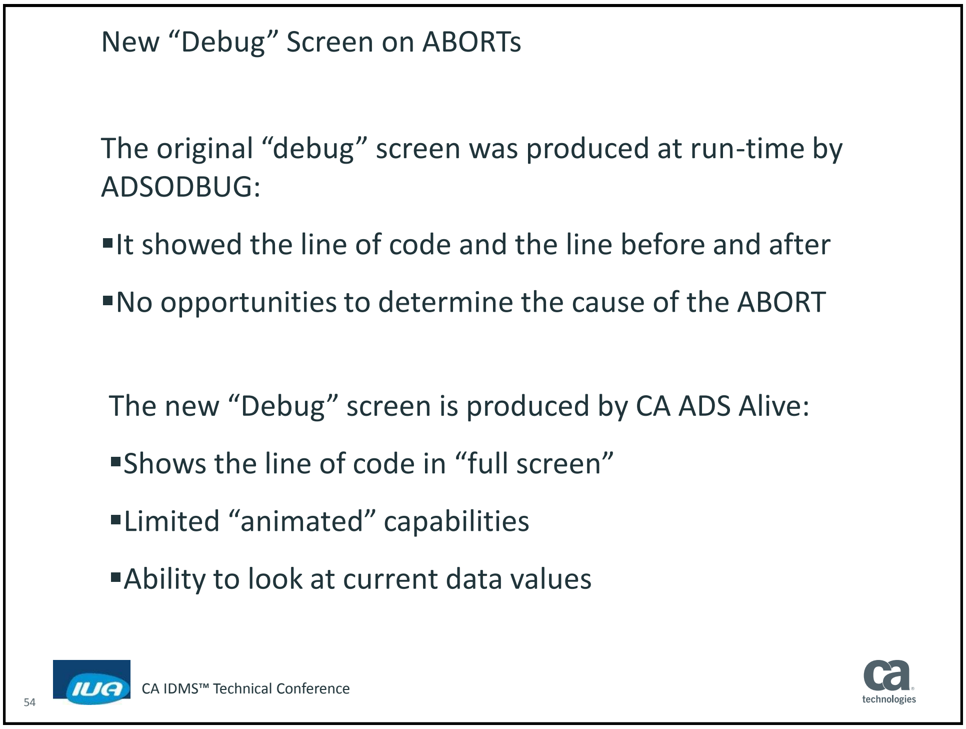
```
INQDEPT                 Department Inquiry                   10/04/01
                          "Buggy" Version                     HRID99M
Sel    Emp ID    Employee Name                Job Title
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
_      0120      MICHAEL   ANGELO             ILLUSTRATOR
_      0007      MONTE     BANK               MGR PUBLIC RELATIONS
_      0009      HUMPHREY  BEAR
_      0119      CHARLES   BOWER              PHOTOGRAPHER
_      0568      ROGER     CLINTON
_      0083      MICHAEL   FITZPATRICK
_      0158      JOCK      JACKSON             ORTS CONSULTANT
_      0080      DAVID     LONERGAN
_      0082      SUE       LONERGAN
_      0127      CAROL     MCDOUGALL
_      7778      PHILO     NGUYEN
_      0149      LAURA     PENMAN
_      8894      MOM       THANG
_      0071      WARREN    WIDGET
_      0476      BETSY     ZEDI
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NEXT RESPONSE:  INQDEPT     SEND DATA
** Select an Employee **

                      PRESS PF13 FOR HELP SCREEN
```

- success – we didn’t get a “DATA EXCEPTION” ! Humphrey had a problem which we fixed “on the fly” - and we got thru the rest OK !



Session Outline

- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary



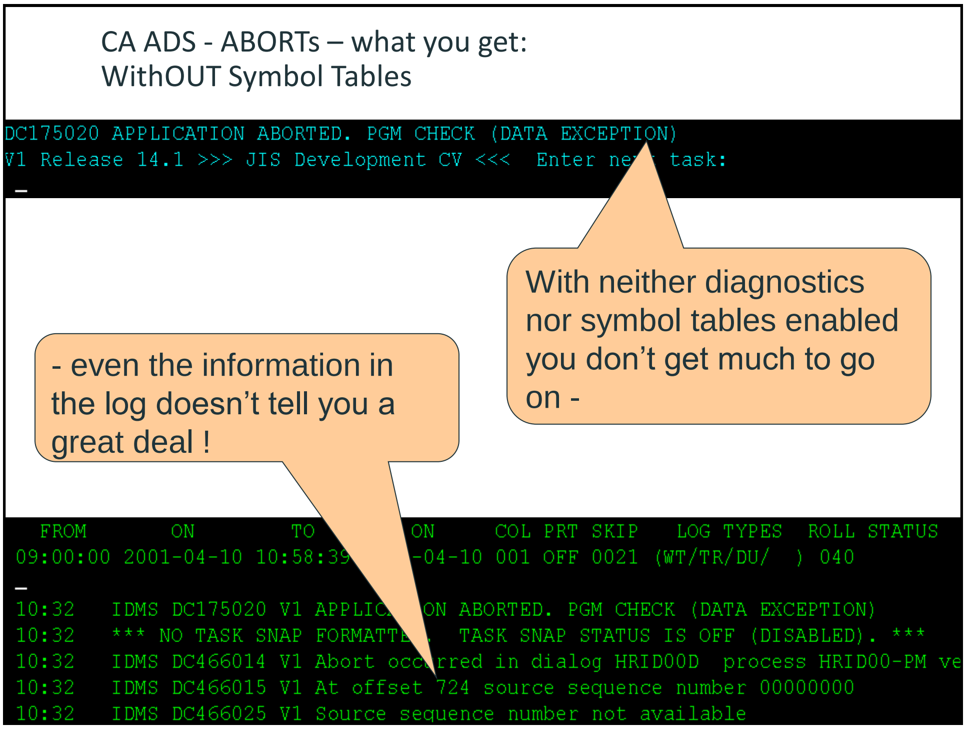
New “Debug” Screen on ABORTs

The original “debug” screen was produced at run-time by ADSODBUG:

- It showed the line of code and the line before and after
- No opportunities to determine the cause of the ABORT

The new “Debug” screen is produced by CA ADS Alive:

- Shows the line of code in “full screen”
- Limited “animated” capabilities
- Ability to look at current data values

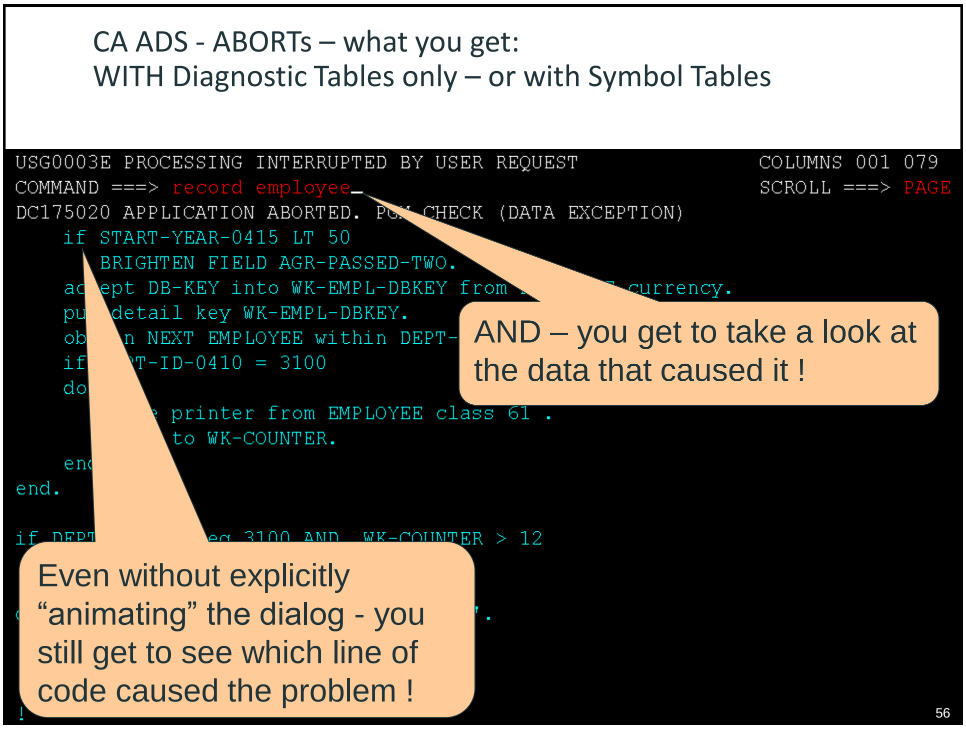


CA ADS - ABORTs – what you get:
WithOUT Symbol Tables

```
DC175020 APPLICATION ABORTED. PGM CHECK (DATA EXCEPTION)
V1 Release 14.1 >>> JIS Development CV <<<  Enter ne   task:
 _

   FROM        ON          TO            ON       COL PRT SKIP   LOG TYPES  ROLL STATUS
 09:00:00 2001-04-10 10:58:39     -04-10 001 OFF 0021 (WT/TR/DU/  ) 040
 _
 10:32   IDMS DC175020 V1 APPLIC    ON ABORTED. PGM CHECK (DATA EXCEPTION)
 10:32   *** NO TASK SNAP FORMATTE  .  TASK SNAP STATUS IS OFF (DISABLED). ***
 10:32   IDMS DC466014 V1 Abort occ rred in dialog HRID00D  process HRID00-PM ve
 10:32   IDMS DC466015 V1 At offset 724 source sequence number 00000000
 10:32   IDMS DC466025 V1 Source sequence number not available
```

With neither diagnostics nor symbol tables enabled you don't get much to go on -

- even the information in the log doesn't tell you a great deal !



CA ADS - ABORTs – what you get:
WITH Diagnostic Tables only – or with Symbol Tables

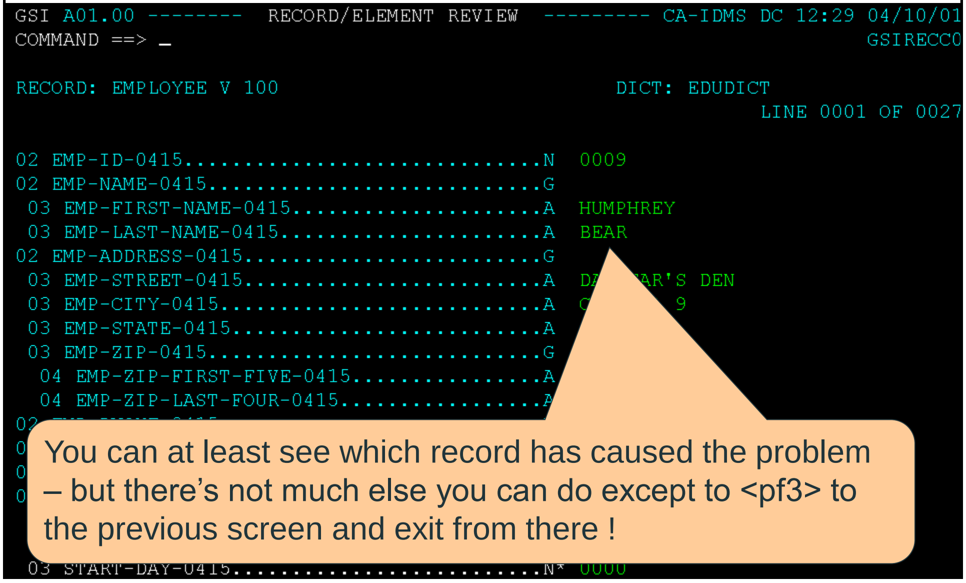
```
USG0003E PROCESSING INTERRUPTED BY USER REQUEST                 COLUMNS 001 079
COMMAND ===> record employee_                                   SCROLL ===> PAGE
DC175020 APPLICATION ABORTED. PG  CHECK (DATA EXCEPTION)
    if START-YEAR-0415 LT 50
       BRIGHTEN FIELD AGR-PASSED-TWO.
    ac ept DB-KEY into WK-EMPL-DBKEY from          currency.
    pu  detail key WK-EMPL-DBKEY.
    ob   n NEXT EMPLOYEE within DEPT-
    if    T-ID-0410 = 3100
    do
             printer from EMPLOYEE class 61 .
             to WK-COUNTER.

    en
end.

if DEPT         eq 3100 AND  WK-COUNTER > 12
```

AND – you get to take a look at the data that caused it !

Even without explicitly "animating" the dialog - you still get to see which line of code caused the problem !

56

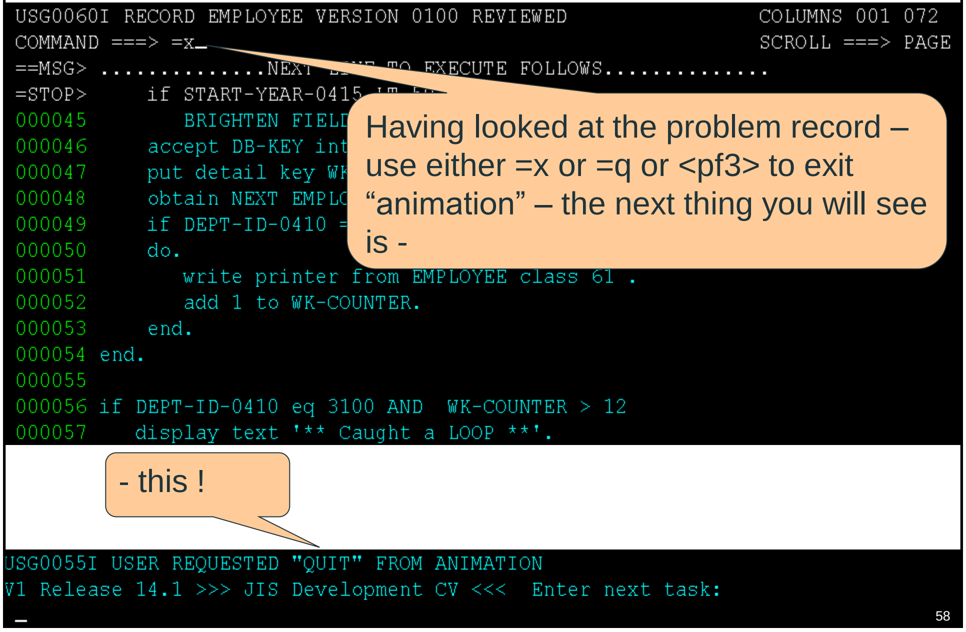
CA ADS - ABORTs – what you get:
Ability to look at the data



```
GSI A01.00 --------  RECORD/ELEMENT REVIEW  --------- CA-IDMS DC 12:29 04/10/01
COMMAND ==> _                                                           GSIRECC0

RECORD: EMPLOYEE V 100                            DICT: EDUDICT
                                                        LINE 0001 OF 0027

02 EMP-ID-0415.............................N  0009
02 EMP-NAME-0415...........................G
 03 EMP-FIRST-NAME-0415....................A  HUMPHREY
 03 EMP-LAST-NAME-0415.....................A  BEAR
02 EMP-ADDRESS-0415........................G
 03 EMP-STREET-0415........................A  DA    AR'S DEN
 03 EMP-CITY-0415..........................A  C       9
 03 EMP-STATE-0415.........................A
 03 EMP-ZIP-0415...........................G
  04 EMP-ZIP-FIRST-FIVE-0415...............A
  04 EMP-ZIP-LAST-FOUR-0415................A
02
0
0
0

 03 START-DAY-0415.........................N* 0000
```

You can at least see which record has caused the problem – but there’s not much else you can do except to <pf3> to the previous screen and exit from there !

CA ADS - ABORTs – what you get:
Ability to look at the data



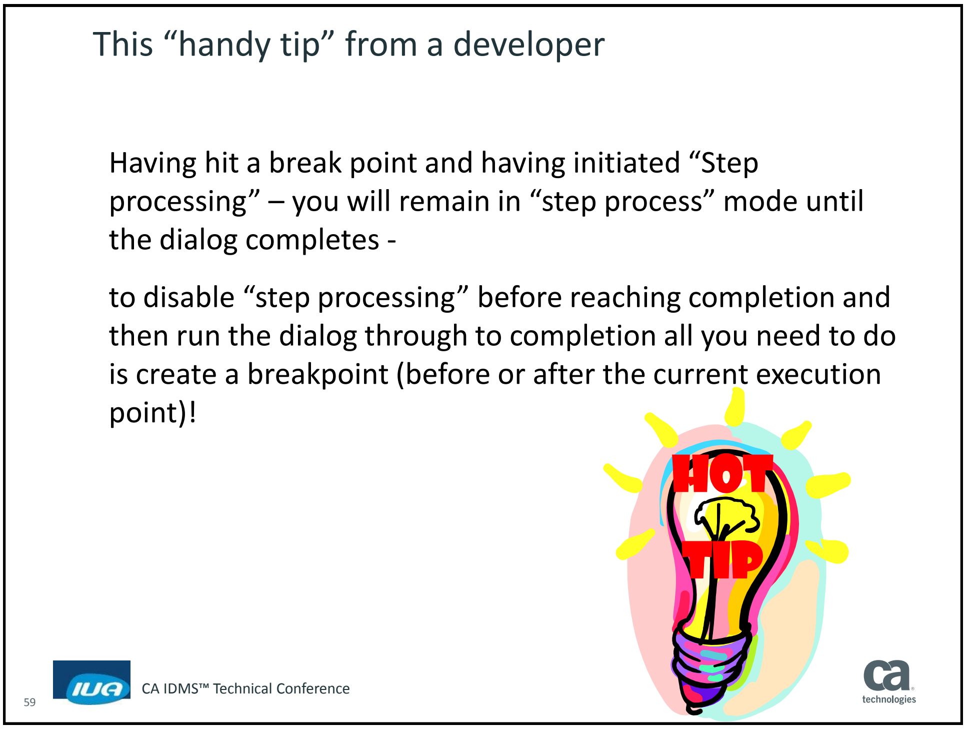
```
USG0060I RECORD EMPLOYEE VERSION 0100 REVIEWED                 COLUMNS 001 072
COMMAND ===> =x_                                               SCROLL ===> PAGE
==MSG> ...............NEXT        EXECUTE FOLLOWS.............
=STOP>     if START-YEAR-0415
000045         BRIGHTEN FIELD
000046     accept DB-KEY int
000047     put detail key WK
000048     obtain NEXT EMPLO
000049     if DEPT-ID-0410 =
000050     do.
000051        write printer from EMPLOYEE class 61 .
000052        add 1 to WK-COUNTER.
000053     end.
000054 end.
000055
000056 if DEPT-ID-0410 eq 3100 AND  WK-COUNTER > 12
000057    display text '** Caught a LOOP **'.
```

Having looked at the problem record – use either =x or =q or <pf3> to exit “animation” – the next thing you will see is -

- this !

```
USG0055I USER REQUESTED "QUIT" FROM ANIMATION
V1 Release 14.1 >>> JIS Development CV <<<  Enter next task:
_
```

58

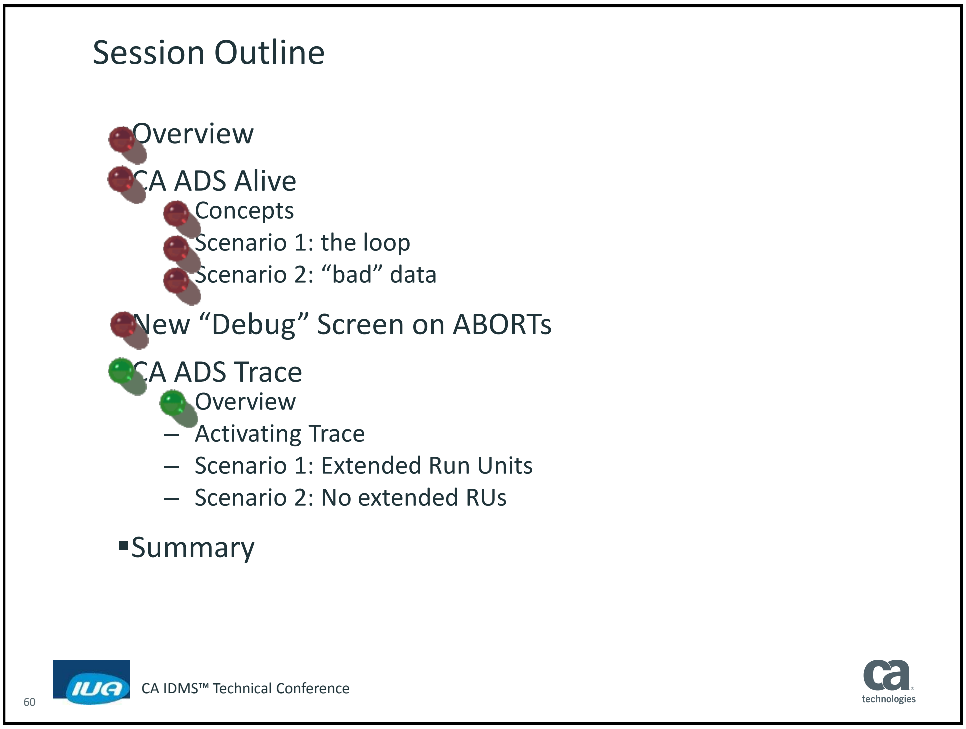


This “handy tip” from a developer

Having hit a break point and having initiated “Step processing” – you will remain in “step process” mode until the dialog completes -

to disable “step processing” before reaching completion and then run the dialog through to completion all you need to do is create a breakpoint (before or after the current execution point)!

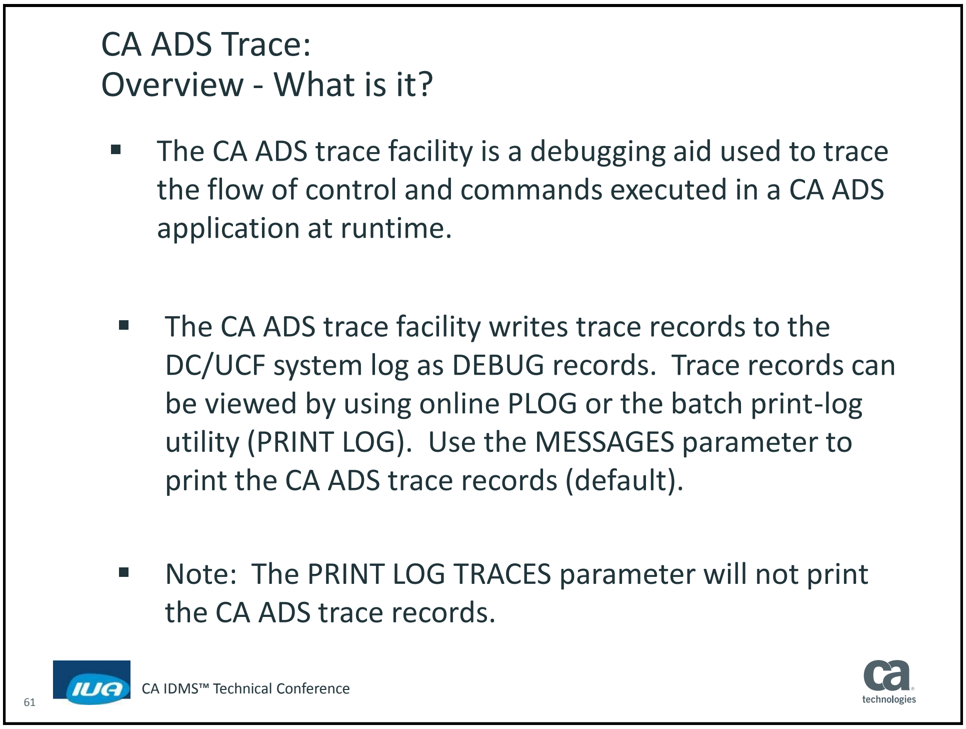
59 CA IDMS™ Technical Conference



Session Outline

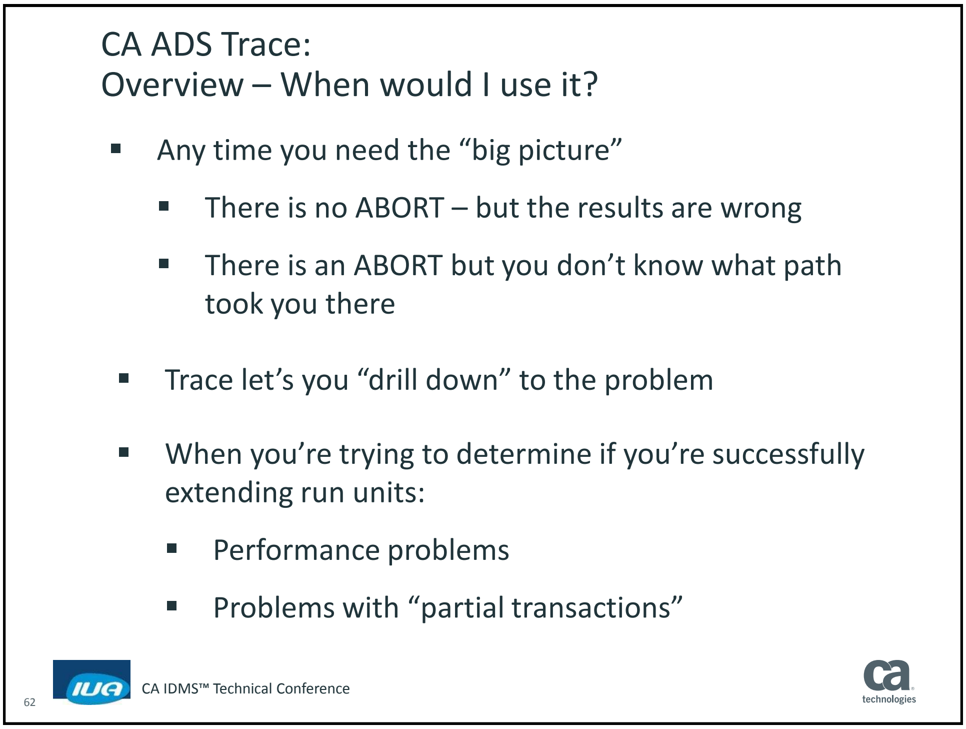
- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary

60 CA IDMS™ Technical Conference



CA ADS Trace: Overview - What is it?

- The CA ADS trace facility is a debugging aid used to trace the flow of control and commands executed in a CA ADS application at runtime.
- The CA ADS trace facility writes trace records to the DC/UCF system log as DEBUG records. Trace records can be viewed by using online PLOG or the batch print-log utility (PRINT LOG). Use the MESSAGES parameter to print the CA ADS trace records (default).
- Note: The PRINT LOG TRACES parameter will not print the CA ADS trace records.



CA ADS Trace: Overview – When would I use it?

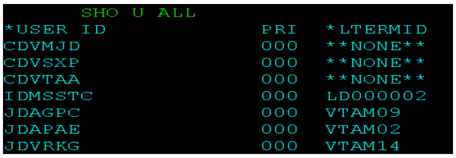
- Any time you need the “big picture”
  - There is no ABORT – but the results are wrong
  - There is an ABORT but you don’t know what path took you there
- Trace let’s you “drill down” to the problem
- When you’re trying to determine if you’re successfully extending run units:
  - Performance problems
  - Problems with “partial transactions”

CA ADS Trace:
Looking at the trace Output 1 of 3

- Use the “online print log” facility – invoked with the OLP task code
- When invoking OLP you can specify date/time limits:
  - Enter next task code: OLP FROM 1235
- When invoking OLP you can specify that only a particular session should be displayed:
  - Enter next task code: OLP LTE ‘VTAM03’
- Of course you can always combine commands:
  - Enter next task: OLP LTE ‘VTAM03’ from 1359
  - Enter next task: OLP from 0741 roll 31

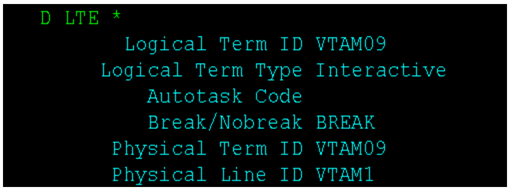
CA ADS Trace:
Looking at the trace Output 2 of 3

- How do I know which terminal session I want to display?
- For another user at e-n-t type: DCUF SHO U ALL



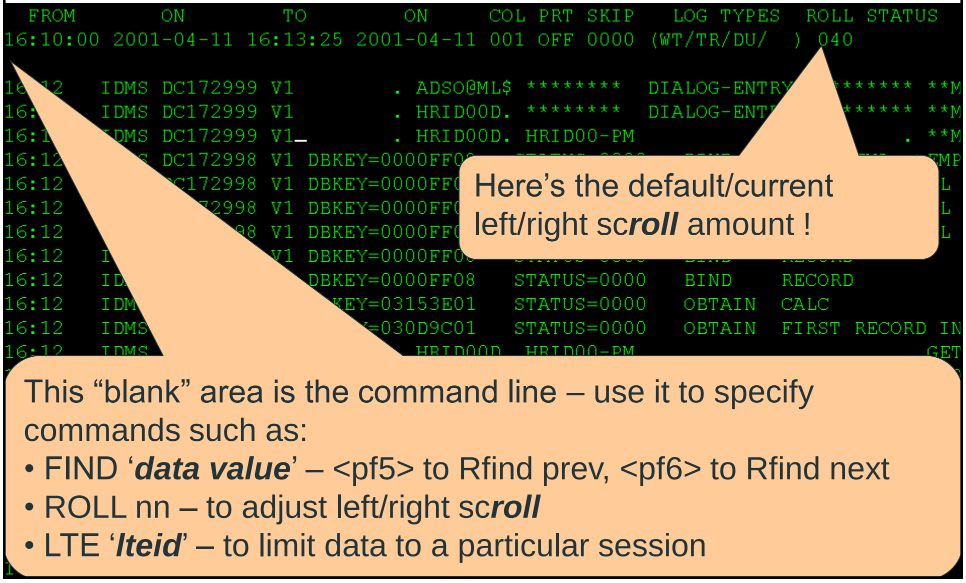
```
      SHO U ALL
*USER ID                 PRI   *LTERMID
CDVMJD                   000   **NONE**
CDVSXP                   000   **NONE**
CDVTAA                   000   **NONE**
IDMSSTC                  000   LD000002
JDAGPC                   000   VTAM09
JDAPAE                   000   VTAM02
JDVRKG                   000   VTAM14
```

- For yourself at e-n-t type: DCMT D LTE *



```
  D LTE *
          Logical Term ID VTAM09
        Logical Term Type Interactive
            Autotask Code
          Break/Nobreak BREAK
         Physical Term ID VTAM09
         Physical Line ID VTAM1
```

CA ADS Trace:
Looking at the trace Output   3 of 3



```
 FROM        ON         TO         ON      COL PRT SKIP   LOG TYPES  ROLL STATUS
16:10:00 2001-04-11 16:13:25 2001-04-11 001 OFF 0000 (WT/TR/DU/  ) 040

16 12   IDMS DC172999 V1         . ADSO@ML$ ********  DIALOG-ENTRY  ******* **M
16:     IDMS DC172999 V1         . HRID00D. ********  DIALOG-ENT    ****** **M
16:1     DMS DC172999 V1_        . HRID00D. HRID00-PM                     . **M
16:12      S DC172998 V1 DBKEY=0000FF0                                      EMP
16:12        C172998 V1 DBKEY=0000FF                                        L
16:12          72998 V1 DBKEY=0000FF                                        L
16:12             98 V1 DBKEY=0000FF                                        L
16:12   I            V1 DBKEY=0000FF0
16:12   ID              DBKEY=0000FF08    STATUS=0000   BIND    RECORD
16:12   IDM               KEY=03153E01    STATUS=0000   OBTAIN  CALC
16:12   IDMS                =030D9C01    STATUS=0000   OBTAIN  FIRST RECORD IN
16:12   IDMS                 HRID00D. HRID00-PM                            GET
```

Here's the default/current left/right sc***roll*** amount !

This "blank" area is the command line – use it to specify commands such as:

- FIND '***data value***' – <pf5> to Rfind prev, <pf6> to Rfind next
- ROLL nn – to adjust left/right sc***roll***
- LTE '***lteid***' – to limit data to a particular session

CA ADS Trace:
Activating Trace

- Use the TRACE keyword in the runtime system initiating statement when requesting execution of the application – for example:
  - Enter next task: ADS TRACE=ALL *HRTASK*
  - Enter next task: ADS TRACE=CTL *DEBTORS*
- By default you will get Trace output for every dialog and menu from the start of the application to the bit you're really interested in
- You can control this and reduce the amount of output going to the log by -

coding the TRACE command in the dialog process!

CA ADS Trace:

Why do I sometimes get different output?

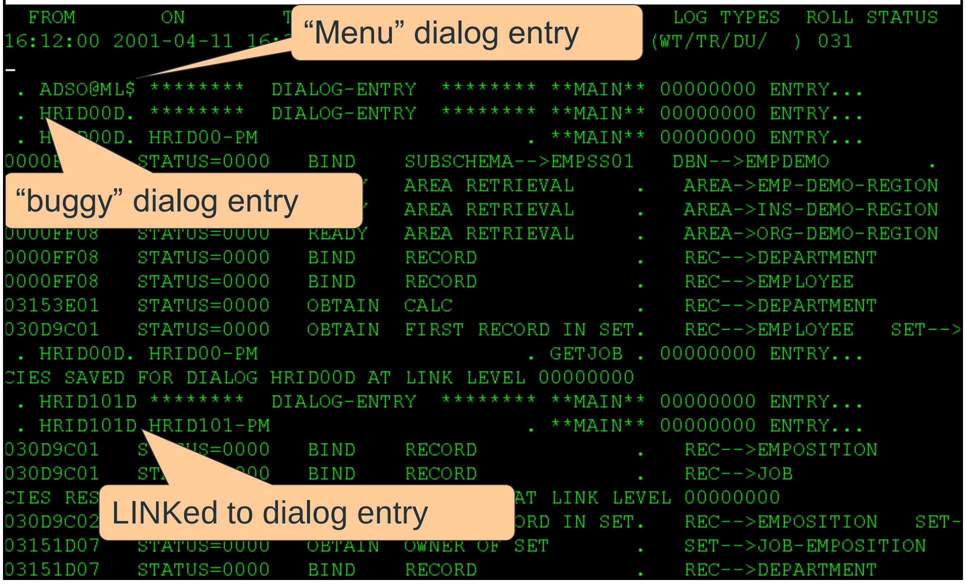
- 2 variables control what output you get:
  - TRACE=ALL/CTL
  - DIAGNOSTIC/SYMBOL Table options
- Your “**base**” information is dialog “**entry**” information and all explicit/implicit **DML** verbs

Trace	No Tables	Diagnostic	Symbol Tables
=CTL	base	base	base + more “entry” info - easier to follow
=ALL	base	base	Every command

- In the following examples we will show you what to expect for each of the three variations of output

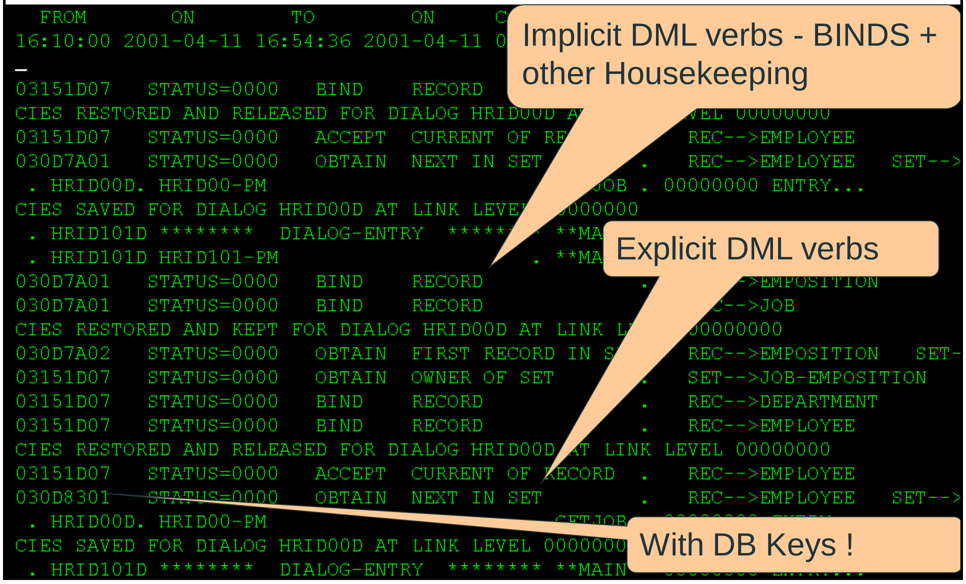
CA ADS Trace:

TRACE=ALL + No Tables 1 of 2



```
  FROM        ON        T                                LOG TYPES   ROLL STATUS
16:12:00 2001-04-11 16:                                (WT/TR/DU/   ) 031
_
 . ADSO@ML$ ********   DIALOG-ENTRY   ******** **MAIN** 00000000 ENTRY...
 . HRID00D. ********   DIALOG-ENTRY   ******** **MAIN** 00000000 ENTRY...
 . H   00D. HRID00-PM                        . **MAIN** 00000000 ENTRY...
0000F      STATUS=0000    BIND     SUBSCHEMA-->EMPSS01    DBN-->EMPDEMO          .
                                   AREA RETRIEVAL      .    AREA->EMP-DEMO-REGION
                                   AREA RETRIEVAL      .    AREA->INS-DEMO-REGION
0000FF08   STATUS=0000    READY    AREA RETRIEVAL      .    AREA->ORG-DEMO-REGION
0000FF08   STATUS=0000    BIND     RECORD              .    REC-->DEPARTMENT
0000FF08   STATUS=0000    BIND     RECORD              .    REC-->EMPLOYEE
03153E01   STATUS=0000    OBTAIN   CALC                .    REC-->DEPARTMENT
030D9C01   STATUS=0000    OBTAIN   FIRST RECORD IN SET.    REC-->EMPLOYEE    SET-->
 . HRID00D. HRID00-PM                        . GETJOB . 00000000 ENTRY...
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
 . HRID101D ********   DIALOG-ENTRY   ******** **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                       . **MAIN** 00000000 ENTRY...
030D9C01   S    US=0000    BIND     RECORD              .    REC-->EMPOSITION
030D9C01   ST      000    BIND     RECORD              .    REC-->JOB
CIES RES                             AT LINK LEVEL 00000000
030D9C02                             ORD IN SET.    REC-->EMPOSITION    SET-
03151D07   STATUS=0000    OBTAIN   OWNER OF SET        .    SET-->JOB-EMPOSITION
03151D07   STATUS=0000    BIND     RECORD              .    REC-->DEPARTMENT
```

CA ADS Trace:
TRACE=ALL + No Tables 2 of 2



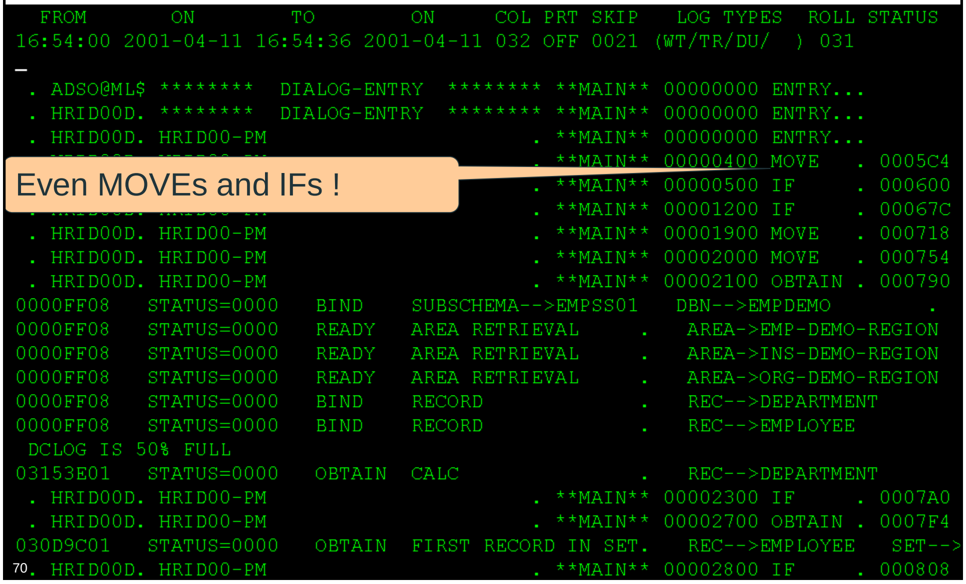
```
  FROM        ON         TO         ON      C
16:10:00 2001-04-11 16:54:36 2001-04-11 0
_
03151D07   STATUS=0000   BIND    RECORD
CIES RESTORED AND RELEASED FOR DIALOG HRID00D A              VEL 00000000
03151D07   STATUS=0000   ACCEPT  CURRENT OF RE                 REC-->EMPLOYEE
030D7A01   STATUS=0000   OBTAIN  NEXT IN SET               .   REC-->EMPLOYEE   SET-->
 . HRID00D. HRID00-PM                                  JOB . 00000000 ENTRY...
CIES SAVED FOR DIALOG HRID00D AT LINK LEVE    000000
 . HRID101D ********  DIALOG-ENTRY  ******    **MA
 . HRID101D HRID101-PM                      . **MA
030D7A01   STATUS=0000   BIND    RECORD                    .      >EMPOSITION
030D7A01   STATUS=0000   BIND    RECORD                         -->JOB
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK L         00000000
030D7A02   STATUS=0000   OBTAIN  FIRST RECORD IN S             REC-->EMPOSITION   SET-
03151D07   STATUS=0000   OBTAIN  OWNER OF SET              .   SET-->JOB-EMPOSITION
03151D07   STATUS=0000   BIND    RECORD                    .   REC-->DEPARTMENT
03151D07   STATUS=0000   BIND    RECORD                    .   REC-->EMPLOYEE
CIES RESTORED AND RELEASED FOR DIALOG HRID00D AT LINK LEVEL 00000000
03151D07   STATUS=0000   ACCEPT  CURRENT OF RECORD  .   REC-->EMPLOYEE
030D8301   STATUS=0000   OBTAIN  NEXT IN SET               .   REC-->EMPLOYEE   SET-->
 . HRID00D. HRID00-PM
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 0000000
 . HRID101D ********  DIALOG-ENTRY  ******** **MAIN
```

Implicit DML verbs - BINDS + other Housekeeping

Explicit DML verbs

With DB Keys !

CA ADS Trace:
TRACE=ALL + Symbol Tables 1 of 4

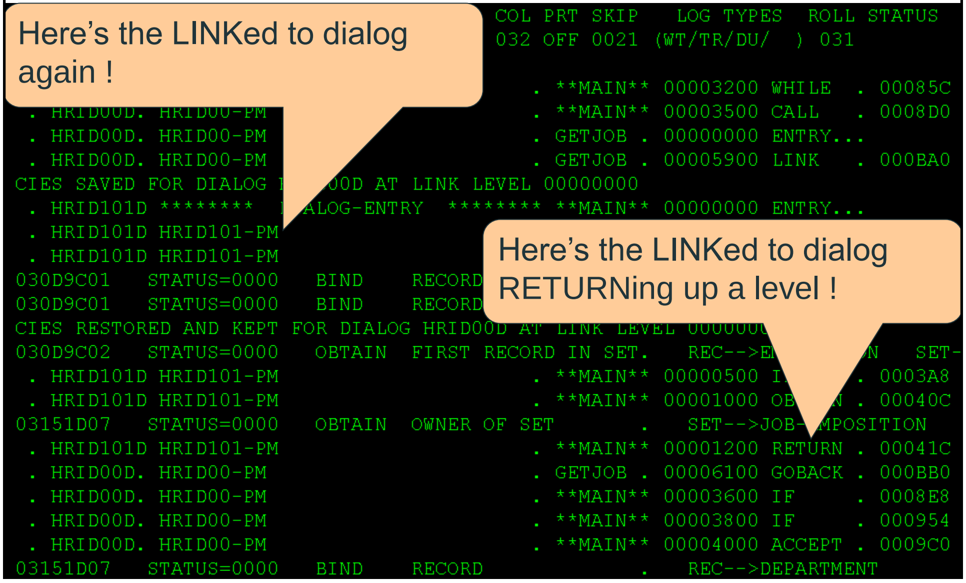


```
  FROM        ON         TO         ON      COL PRT SKIP   LOG TYPES   ROLL STATUS
16:54:00 2001-04-11 16:54:36 2001-04-11 032 OFF 0021 (WT/TR/DU/   ) 031
_
 . ADSO@ML$ ********  DIALOG-ENTRY  ******** **MAIN** 00000000 ENTRY...
 . HRID00D. ********  DIALOG-ENTRY  ******** **MAIN** 00000000 ENTRY...
 . HRID00D. HRID00-PM                      . **MAIN** 00000000 ENTRY...
                                           . **MAIN** 00000400 MOVE   . 0005C4
                                           . **MAIN** 00000500 IF     . 000600
                                           . **MAIN** 00001200 IF     . 00067C
 . HRID00D. HRID00-PM                      . **MAIN** 00001900 MOVE   . 000718
 . HRID00D. HRID00-PM                      . **MAIN** 00002000 MOVE   . 000754
 . HRID00D. HRID00-PM                      . **MAIN** 00002100 OBTAIN . 000790
0000FF08   STATUS=0000   BIND    SUBSCHEMA-->EMPSS01   DBN-->EMPDEMO         .
0000FF08   STATUS=0000   READY   AREA RETRIEVAL     .   AREA->EMP-DEMO-REGION
0000FF08   STATUS=0000   READY   AREA RETRIEVAL     .   AREA->INS-DEMO-REGION
0000FF08   STATUS=0000   READY   AREA RETRIEVAL     .   AREA->ORG-DEMO-REGION
0000FF08   STATUS=0000   BIND    RECORD             .   REC-->DEPARTMENT
0000FF08   STATUS=0000   BIND    RECORD             .   REC-->EMPLOYEE
 DCLOG IS 50% FULL
03153E01   STATUS=0000   OBTAIN  CALC               .   REC-->DEPARTMENT
 . HRID00D. HRID00-PM                      . **MAIN** 00002300 IF     . 0007A0
 . HRID00D. HRID00-PM                      . **MAIN** 00002700 OBTAIN . 0007F4
030D9C01   STATUS=0000   OBTAIN  FIRST RECORD IN SET.   REC-->EMPLOYEE   SET-->
 . HRID00D. HRID00-PM                      . **MAIN** 00002800 IF     . 000808
```

70

Even MOVEs and IFs !

CA ADS Trace:
TRACE=ALL + Symbol Tables 2 of 4

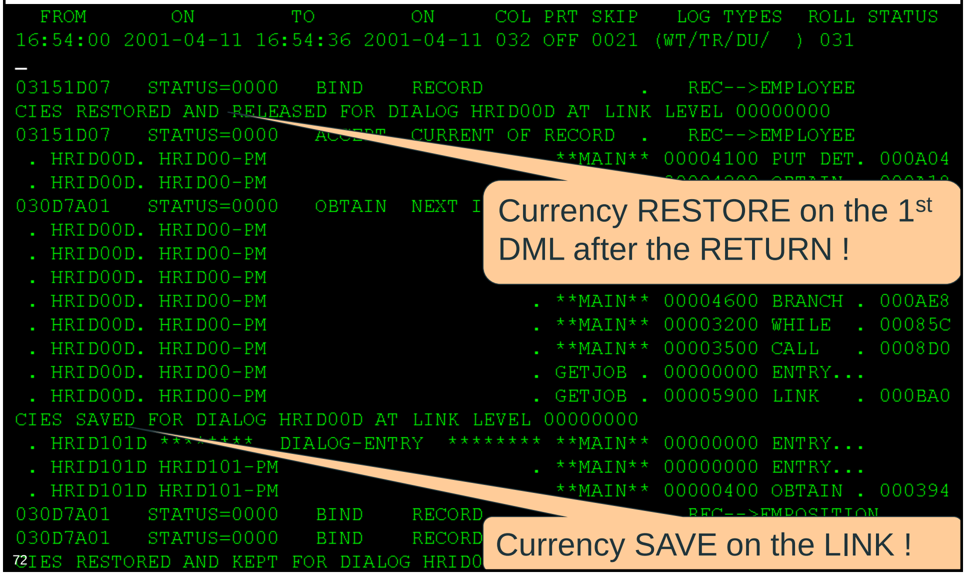


```
                                        COL PRT SKIP   LOG TYPES  ROLL STATUS
                                        032 OFF 0021 (WT/TR/DU/  ) 031

                                           . **MAIN** 00003200 WHILE  . 00085C
 . HRID00D. HRID00-PM                      . **MAIN** 00003500 CALL   . 0008D0
 . HRID00D. HRID00-PM                      . GETJOB . 00000000 ENTRY...
 . HRID00D. HRID00-PM                      . GETJOB . 00005900 LINK   . 000BA0
CIES SAVED FOR DIALOG I   00D AT LINK LEVEL 00000000
 . HRID101D ********  I ALOG-ENTRY  ******** **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM
 . HRID101D HRID101-PM
030D9C01   STATUS=0000   BIND    RECORD
030D9C01   STATUS=0000   BIND    RECORD
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 000000
030D9C02   STATUS=0000   OBTAIN  FIRST RECORD IN SET.   REC-->E        N   SET-
 . HRID101D HRID101-PM                     . **MAIN** 00000500 I      . 0003A8
 . HRID101D HRID101-PM                     . **MAIN** 00001000 OB   N . 00040C
03151D07   STATUS=0000   OBTAIN  OWNER OF SET       .   SET-->JOB- MPOSITION
 . HRID101D HRID101-PM                     . **MAIN** 00001200 RETURN . 00041C
 . HRID00D. HRID00-PM                      . GETJOB . 00006100 GOBACK . 000BB0
 . HRID00D. HRID00-PM                      . **MAIN** 00003600 IF     . 0008E8
 . HRID00D. HRID00-PM                      . **MAIN** 00003800 IF     . 000954
 . HRID00D. HRID00-PM                      . **MAIN** 00004000 ACCEPT . 0009C0
03151D07   STATUS=0000   BIND    RECORD             .   REC-->DEPARTMENT
```

Here's the LINKed to dialog again !

Here's the LINKed to dialog RETURNing up a level !

CA ADS Trace:
TRACE=ALL + Symbol Tables 3 of 4

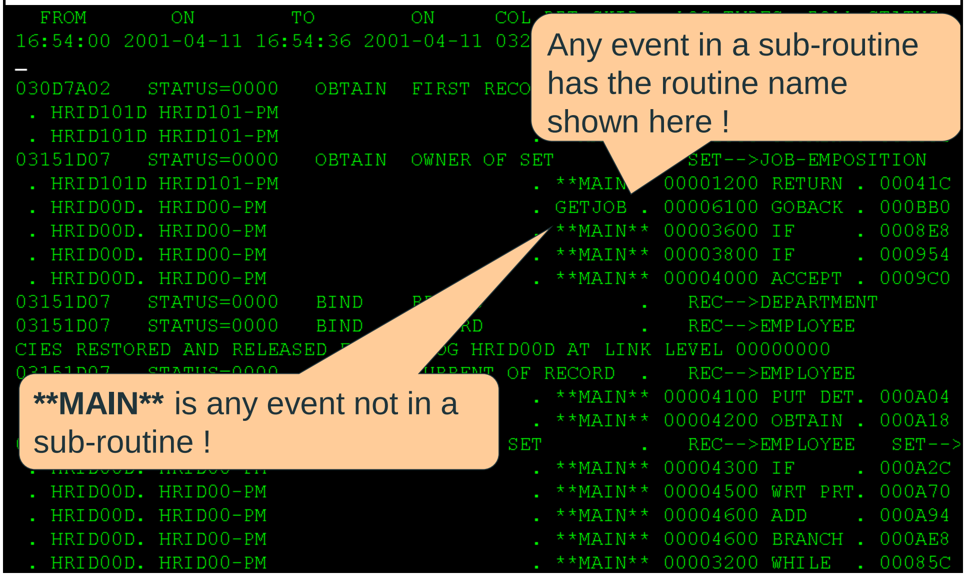


```
  FROM       ON        TO        ON       COL PRT SKIP   LOG TYPES  ROLL STATUS
16:54:00 2001-04-11 16:54:36 2001-04-11 032 OFF 0021 (WT/TR/DU/  ) 031
_
03151D07   STATUS=0000   BIND    RECORD             .   REC-->EMPLOYEE
CIES RESTORED AND RELEASED FOR DIALOG HRID00D AT LINK LEVEL 00000000
03151D07   STATUS=0000   ACCEPT  CURRENT OF RECORD  .   REC-->EMPLOYEE
 . HRID00D. HRID00-PM                        **MAIN** 00004100 PUT DET. 000A04
 . HRID00D. HRID00-PM
030D7A01   STATUS=0000   OBTAIN  NEXT  I
 . HRID00D. HRID00-PM
 . HRID00D. HRID00-PM
 . HRID00D. HRID00-PM
 . HRID00D. HRID00-PM                      . **MAIN** 00004600 BRANCH . 000AE8
 . HRID00D. HRID00-PM                      . **MAIN** 00003200 WHILE  . 00085C
 . HRID00D. HRID00-PM                      . **MAIN** 00003500 CALL   . 0008D0
 . HRID00D. HRID00-PM                      . GETJOB . 00000000 ENTRY...
 . HRID00D. HRID00-PM                      . GETJOB . 00005900 LINK   . 000BA0
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
 . HRID101D ********  DIALOG-ENTRY  ******** **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                     . **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                       **MAIN** 00000400 OBTAIN . 000394
030D7A01   STATUS=0000   BIND    RECORD                 REC-->EMPOSITION
030D7A01   STATUS=0000   BIND    RECORD
CIES RESTORED AND KEPT FOR DIALOG HRID0
```

Currency RESTORE on the 1st DML after the RETURN !

Currency SAVE on the LINK !

CA ADS Trace: TRACE=ALL + Symbol Tables 4 of 4

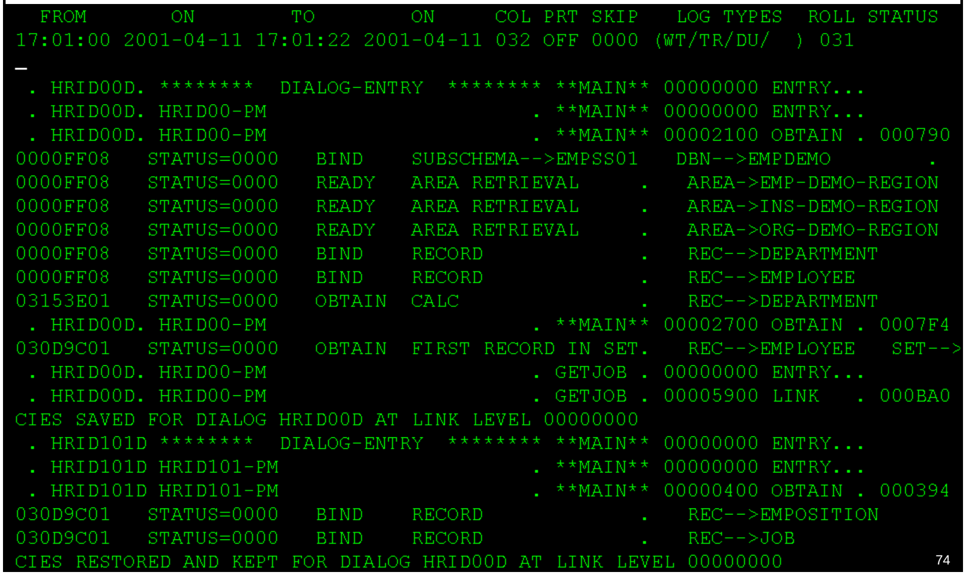


```
  FROM       ON        TO        ON      COL
16:54:00 2001-04-11 16:54:36 2001-04-11 032
_
030D7A02   STATUS=0000   OBTAIN  FIRST RECO
 . HRID101D HRID101-PM
 . HRID101D HRID101-PM                       .
03151D07   STATUS=0000   OBTAIN  OWNER OF SET           SET-->JOB-EMPOSITION
 . HRID101D HRID101-PM                       . **MAIN   00001200 RETURN . 00041C
 . HRID00D. HRID00-PM                        . GETJOB . 00006100 GOBACK . 000BB0
 . HRID00D. HRID00-PM                        . **MAIN** 00003600 IF     . 0008E8
 . HRID00D. HRID00-PM                        . **MAIN** 00003800 IF     . 000954
 . HRID00D. HRID00-PM                        . **MAIN** 00004000 ACCEPT . 0009C0
03151D07   STATUS=0000   BIND    RE                   .   REC-->DEPARTMENT
03151D07   STATUS=0000   BIND        RD               .   REC-->EMPLOYEE
CIES RESTORED AND RELEASED F        OG HRID00D AT LINK LEVEL 00000000
03151D07   STATUS=0000         URRENT OF RECORD .   REC-->EMPLOYEE
                                             . **MAIN** 00004100 PUT DET. 000A04
                                             . **MAIN** 00004200 OBTAIN . 000A18
                                          SET         .   REC-->EMPLOYEE   SET-->
                                             . **MAIN** 00004300 IF     . 000A2C
 . HRID00D. HRID00-PM                        . **MAIN** 00004500 WRT PRT. 000A70
 . HRID00D. HRID00-PM                        . **MAIN** 00004600 ADD    . 000A94
 . HRID00D. HRID00-PM                        . **MAIN** 00004600 BRANCH . 000AE8
 . HRID00D. HRID00-PM                        . **MAIN** 00003200 WHILE  . 00085C
```

Any event in a sub-routine has the routine name shown here !

MAIN is any event not in a sub-routine !

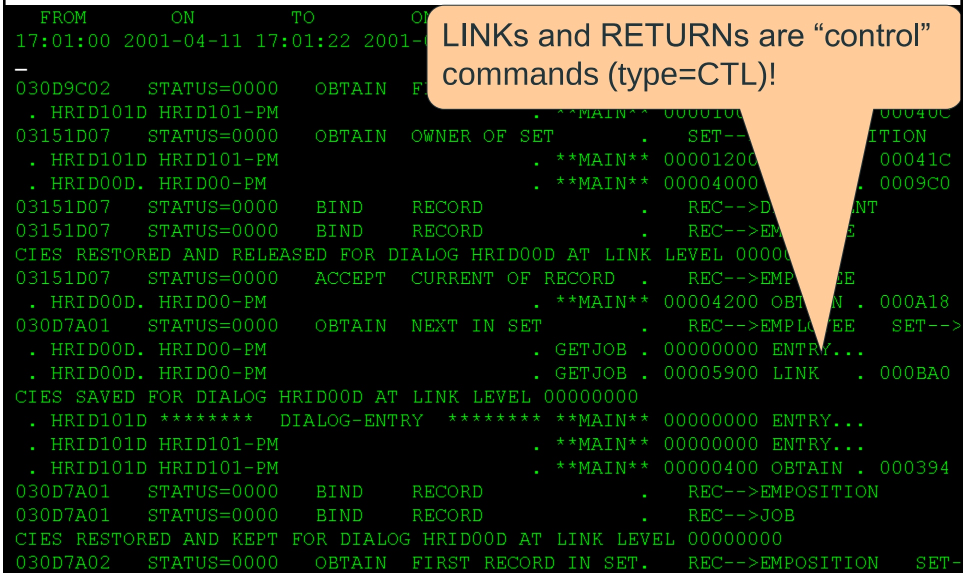
CA ADS Trace: TRACE=CTL + Symbol Tables 1 of 3



```
  FROM       ON        TO        ON      COL PRT SKIP   LOG TYPES  ROLL STATUS
17:01:00 2001-04-11 17:01:22 2001-04-11 032 OFF 0000 (WT/TR/DU/  ) 031
_
 . HRID00D. ********  DIALOG-ENTRY   ******** **MAIN** 00000000 ENTRY...
 . HRID00D. HRID00-PM                        . **MAIN** 00000000 ENTRY...
 . HRID00D. HRID00-PM                        . **MAIN** 00002100 OBTAIN . 000790
0000FF08   STATUS=0000   BIND    SUBSCHEMA-->EMPSS01    DBN-->EMPDEMO        .
0000FF08   STATUS=0000   READY   AREA RETRIEVAL       .   AREA->EMP-DEMO-REGION
0000FF08   STATUS=0000   READY   AREA RETRIEVAL       .   AREA->INS-DEMO-REGION
0000FF08   STATUS=0000   READY   AREA RETRIEVAL       .   AREA->ORG-DEMO-REGION
0000FF08   STATUS=0000   BIND    RECORD               .   REC-->DEPARTMENT
0000FF08   STATUS=0000   BIND    RECORD               .   REC-->EMPLOYEE
03153E01   STATUS=0000   OBTAIN  CALC                 .   REC-->DEPARTMENT
 . HRID00D. HRID00-PM                        . **MAIN** 00002700 OBTAIN . 0007F4
030D9C01   STATUS=0000   OBTAIN  FIRST RECORD IN SET.   REC-->EMPLOYEE   SET-->
 . HRID00D. HRID00-PM                        . GETJOB . 00000000 ENTRY...
 . HRID00D. HRID00-PM                        . GETJOB . 00005900 LINK   . 000BA0
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
 . HRID101D ********  DIALOG-ENTRY   ******** **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                       . **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                       . **MAIN** 00000400 OBTAIN . 000394
030D9C01   STATUS=0000   BIND    RECORD               .   REC-->EMPOSITION
030D9C01   STATUS=0000   BIND    RECORD               .   REC-->JOB
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000
```

74

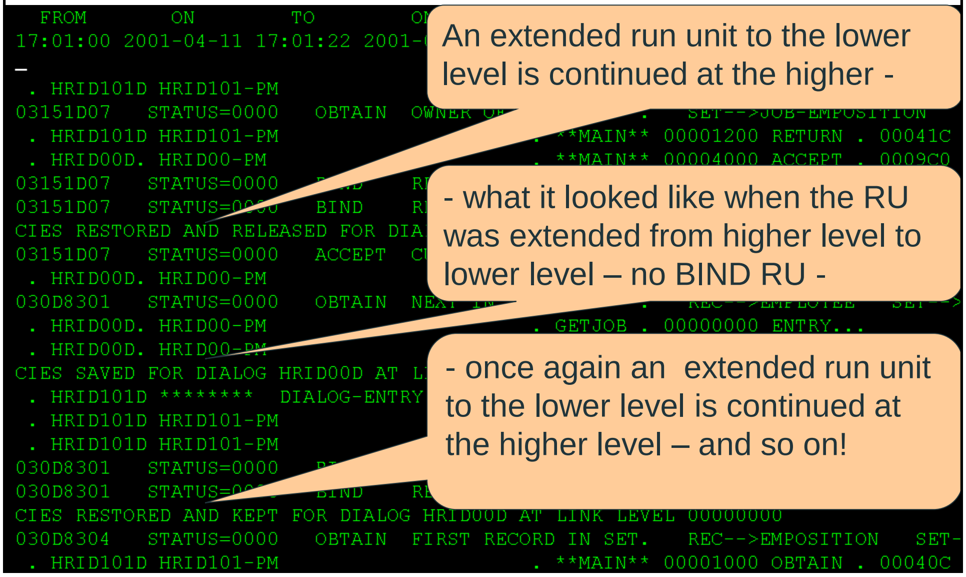
CA ADS Trace:
TRACE=CTL + Symbol Tables 2 of 3



```
  FROM       ON         TO         ON
17:01:00 2001-04-11 17:01:22 2001-(
_
030D9C02   STATUS=0000   OBTAIN   FI
 . HRID101D HRID101-PM                        . **MAIN** 000010                    00040C
03151D07   STATUS=0000   OBTAIN   OWNER OF SET          .   SET--            ITION
 . HRID101D HRID101-PM                        . **MAIN** 00001200              00041C
 . HRID00D. HRID00-PM                         . **MAIN** 00004000            . 0009C0
03151D07   STATUS=0000   BIND     RECORD                .   REC-->D        NT
03151D07   STATUS=0000   BIND     RECORD                .   REC-->EM        E
CIES RESTORED AND RELEASED FOR DIALOG HRID00D AT LINK LEVEL 0000
03151D07   STATUS=0000   ACCEPT   CURRENT OF RECORD     .   REC-->EMP      EE
 . HRID00D. HRID00-PM                         . **MAIN** 00004200 OBT    N . 000A18
030D7A01   STATUS=0000   OBTAIN   NEXT IN SET           .   REC-->EMPL  EE    SET-->
 . HRID00D. HRID00-PM                         . GETJOB . 00000000 ENTRY...
 . HRID00D. HRID00-PM                         . GETJOB . 00005900 LINK     . 000BA0
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
 . HRID101D ********  DIALOG-ENTRY  ******** **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                        . **MAIN** 00000000 ENTRY...
 . HRID101D HRID101-PM                        . **MAIN** 00000400 OBTAIN . 000394
030D7A01   STATUS=0000   BIND     RECORD                .   REC-->EMPOSITION
030D7A01   STATUS=0000   BIND     RECORD                .   REC-->JOB
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000
030D7A02   STATUS=0000   OBTAIN   FIRST RECORD IN SET.  REC-->EMPOSITION   SET-
```

LINKs and RETURNs are “control” commands (type=CTL)!

CA ADS Trace:
TRACE=CTL + Symbol Tables 3 of 3



```
  FROM       ON         TO         ON
17:01:00 2001-04-11 17:01:22 2001-(
_
 . HRID101D HRID101-PM
03151D07   STATUS=0000   OBTAIN   OWNER OF                  SET-->JOB-EMPOSITION
 . HRID101D HRID101-PM                        . **MAIN** 00001200 RETURN . 00041C
 . HRID00D. HRID00-PM                         . **MAIN** 00004000 ACCEPT . 0009C0
03151D07   STATUS=0000            RI
03151D07   STATUS=0     BIND     RI
CIES RESTORED AND RELEASED FOR DIA
03151D07   STATUS=0000   ACCEPT   CU
 . HRID00D. HRID00-PM
030D8301   STATUS=0000   OBTAIN   NE                        REC-->EMPLOYEE    SET-->
 . HRID00D. HRID00-PM                         . GETJOB . 00000000 ENTRY...
 . HRID00D. HRID00-PM
CIES SAVED FOR DIALOG HRID00D AT LI
 . HRID101D ********  DIALOG-ENTRY
 . HRID101D HRID101-PM
 . HRID101D HRID101-PM
030D8301   STATUS=0000
030D8301   STATUS=0     BIND     RE
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000
030D8304   STATUS=0000   OBTAIN   FIRST RECORD IN SET.  REC-->EMPOSITION   SET-
 . HRID101D HRID101-PM                        . **MAIN** 00001000 OBTAIN . 00040C
```

An extended run unit to the lower level is continued at the higher -

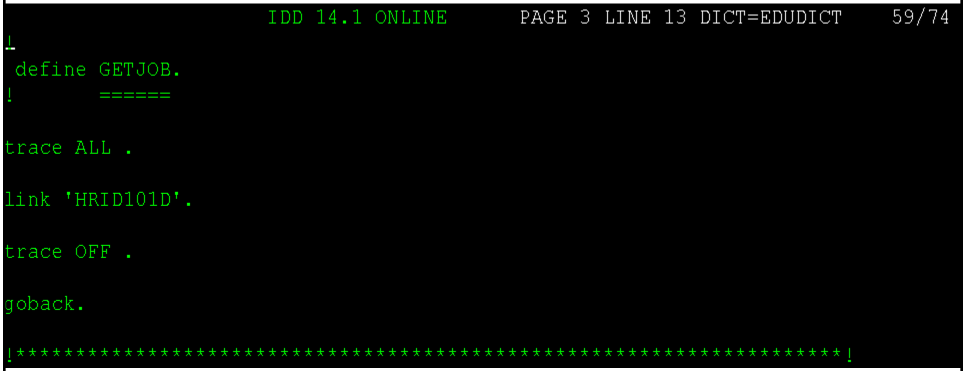
- what it looked like when the RU was extended from higher level to lower level – no BIND RU -

- once again an extended run unit to the lower level is continued at the higher level – and so on!

CA ADS Trace:

Limiting TRACE to Specific Processes

- Specifying TRACE when initiating CA ADS results in tracing the execution of all dialogs in the application



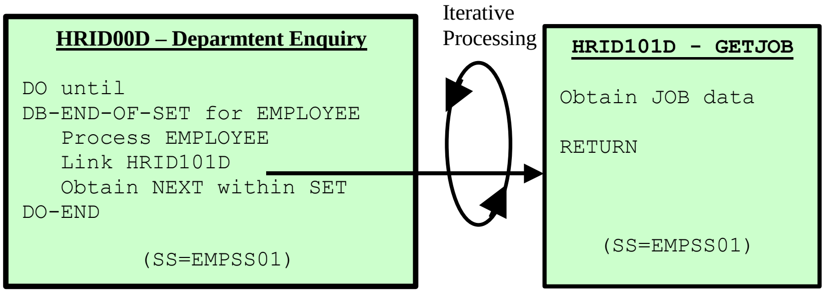
```
                      IDD 14.1 ONLINE      PAGE 3 LINE 13 DICT=EDUDICT    59/74
!
 define GETJOB.
!       ======

trace ALL .

link 'HRID101D'.

trace OFF .

goback.

!*********************************************************************!
```

- You use **TRACE (ALL / CTL)** and **TRACE OFF** in process logic to limit the trace output

Session Outline

- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary

CA ADS Trace:
Scenario 1: Extended Run Unit 1 of 2



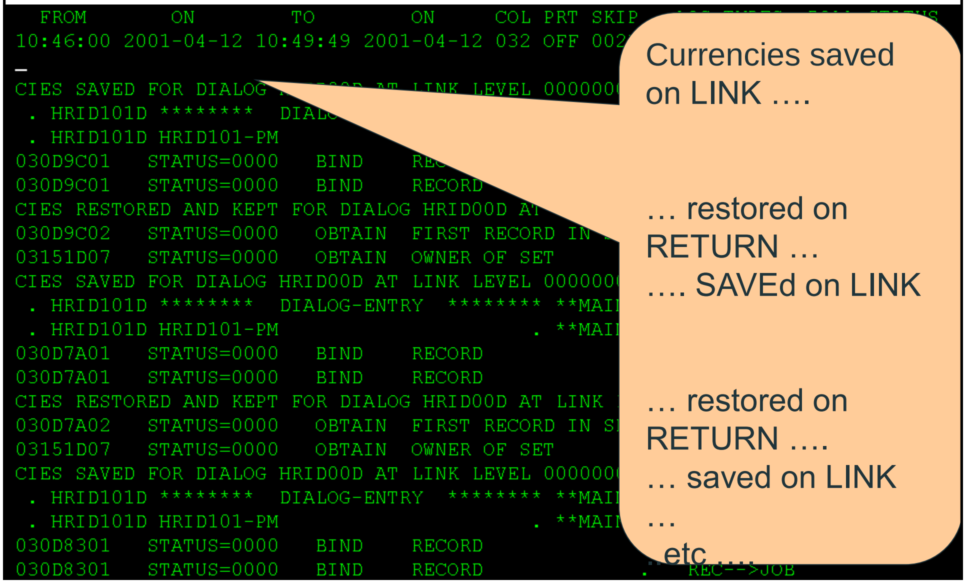
HRID00D – Deparmtent Enquiry

```
DO until
DB-END-OF-SET for EMPLOYEE
    Process EMPLOYEE
    Link HRID101D
    Obtain NEXT within SET
DO-END

            (SS=EMPSS01)
```

HRID101D - GETJOB

```
Obtain JOB data

RETURN


     (SS=EMPSS01)
```

CA ADS Trace:
Scenario 1: Extended Run Unit 2 of 2

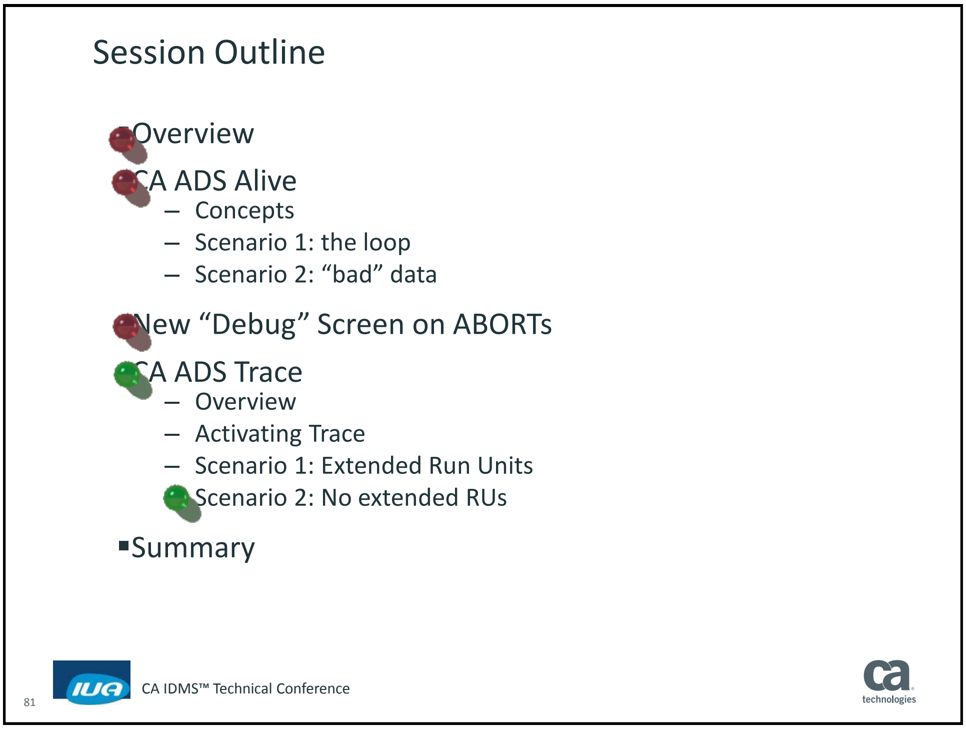


```
  FROM       ON         TO         ON      COL PRT SKIP
10:46:00 2001-04-12 10:49:49 2001-04-12 032 OFF 002
_
CIES SAVED FOR DIALOG
 . HRID101D ********   DIALO
 . HRID101D HRID101-PM
030D9C01   STATUS=0000   BIND    REC
030D9C01   STATUS=0000   BIND    RECORD
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT
030D9C02   STATUS=0000   OBTAIN  FIRST RECORD IN
03151D07   STATUS=0000   OBTAIN  OWNER OF SET
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 000000
 . HRID101D ********   DIALOG-ENTRY  ******** **MAI
 . HRID101D HRID101-PM                        . **MAI
030D7A01   STATUS=0000   BIND    RECORD
030D7A01   STATUS=0000   BIND    RECORD
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK
030D7A02   STATUS=0000   OBTAIN  FIRST RECORD IN S
03151D07   STATUS=0000   OBTAIN  OWNER OF SET
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 000000
 . HRID101D ********   DIALOG-ENTRY  ******** **MAI
 . HRID101D HRID101-PM                        . **MAI
030D8301   STATUS=0000   BIND    RECORD
030D8301   STATUS=0000   BIND    RECORD                 .    REC-->JOB
```

Currencies saved on LINK ….

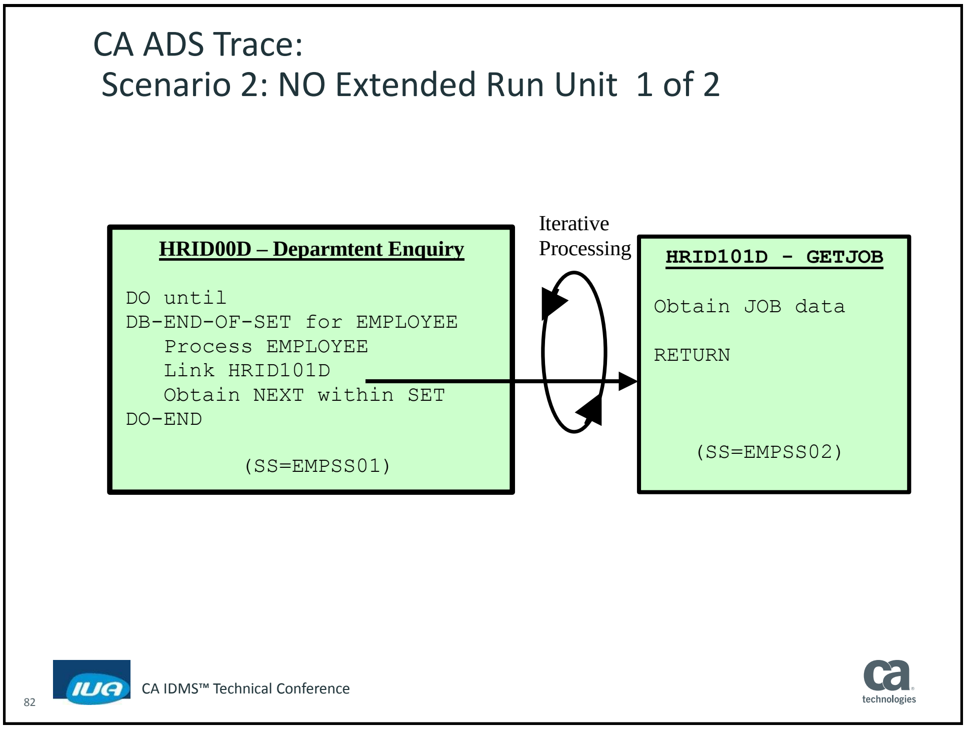
… restored on RETURN …
…. SAVEd on LINK

… restored on RETURN ….
… saved on LINK
…
..etc ….



Session Outline

- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary



CA ADS Trace: Scenario 2: NO Extended Run Unit 1 of 2

HRID00D – Deparmtent Enquiry

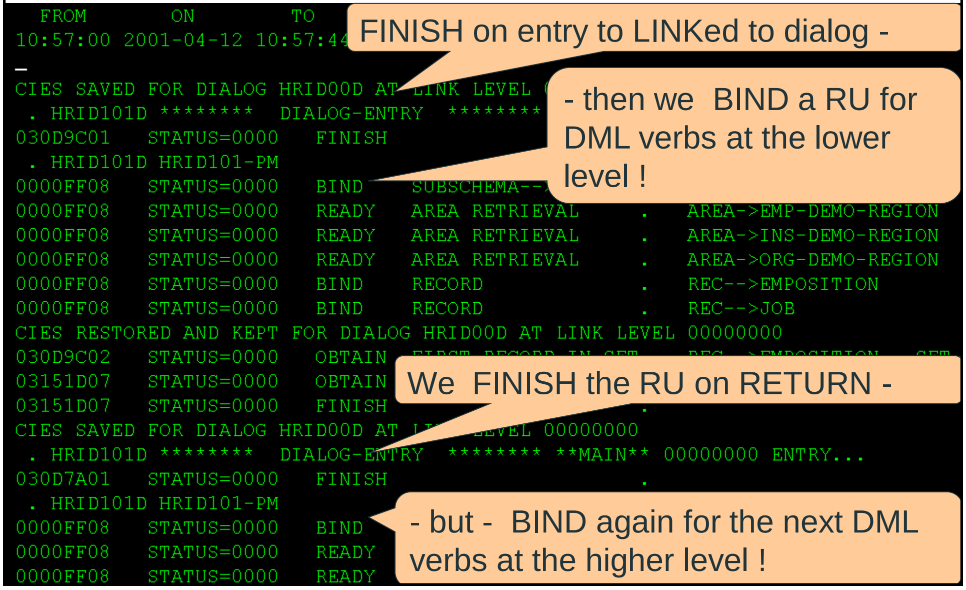
```
DO until
DB-END-OF-SET for EMPLOYEE
   Process EMPLOYEE
   Link HRID101D
   Obtain NEXT within SET
DO-END

         (SS=EMPSS01)
```

Iterative Processing

HRID101D - GETJOB

```
Obtain JOB data

RETURN


   (SS=EMPSS02)
```

CA ADS Trace:
Scenario 2: NO Extended Run Unit 2 of 2



```
  FROM        ON         TO
10:57:00 2001-04-12 10:57:44
_
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL (
 . HRID101D ********  DIALOG-ENTRY  ********
030D9C01   STATUS=0000   FINISH
 . HRID101D HRID101-PM
0000FF08   STATUS=0000   BIND     SUBSCHEMA--
0000FF08   STATUS=0000   READY    AREA RETRIEVAL      .    AREA->EMP-DEMO-REGION
0000FF08   STATUS=0000   READY    AREA RETRIEVAL      .    AREA->INS-DEMO-REGION
0000FF08   STATUS=0000   READY    AREA RETRIEVAL      .    AREA->ORG-DEMO-REGION
0000FF08   STATUS=0000   BIND     RECORD              .    REC-->EMPOSITION
0000FF08   STATUS=0000   BIND     RECORD              .    REC-->JOB
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000
030D9C02   STATUS=0000   OBTAIN   FIRST RECORD IN SET      REC-->EMPOSITION    SET
03151D07   STATUS=0000   OBTAIN
03151D07   STATUS=0000   FINISH                       .
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
 . HRID101D ********  DIALOG-ENTRY  ******** **MAIN** 00000000 ENTRY...
030D7A01   STATUS=0000   FINISH                       .
 . HRID101D HRID101-PM
0000FF08   STATUS=0000   BIND
0000FF08   STATUS=0000   READY
0000FF08   STATUS=0000   READY
```

FINISH on entry to LINKed to dialog -

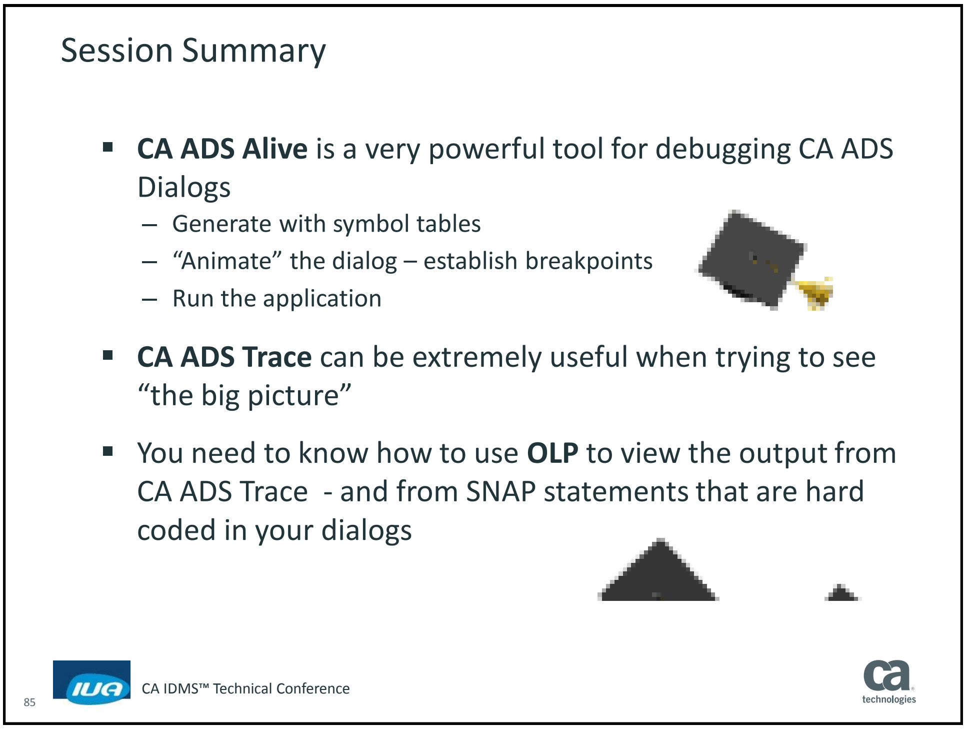
- then we BIND a RU for DML verbs at the lower level !

We FINISH the RU on RETURN -

- but - BIND again for the next DML verbs at the higher level !

Session Outline

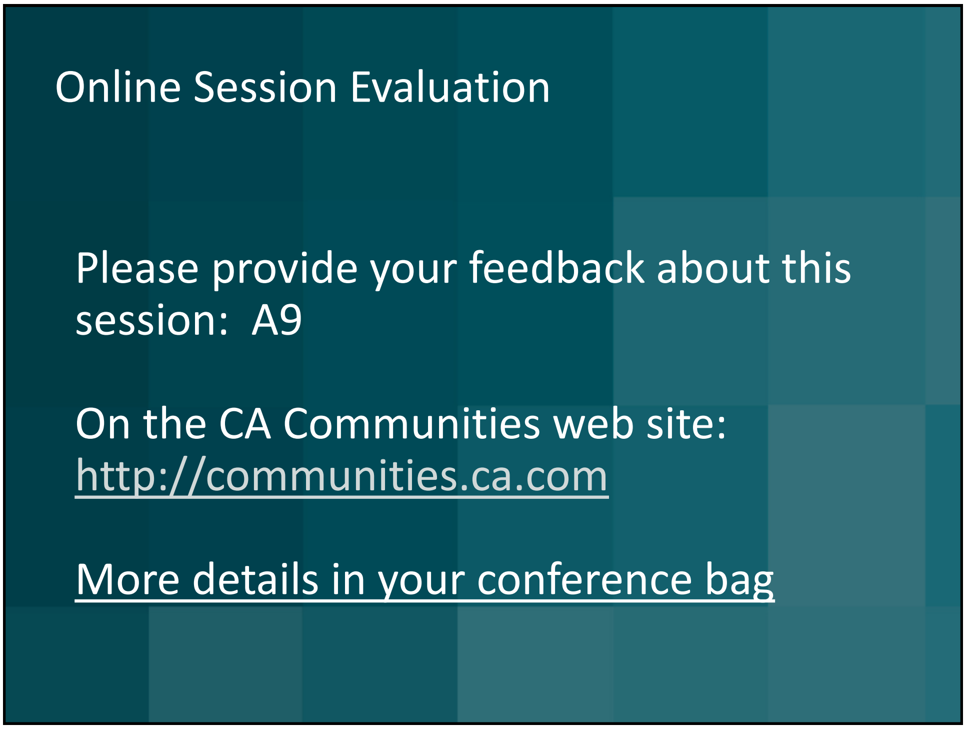
- Overview
- CA ADS Alive
  - Concepts
  - Scenario 1: the loop
  - Scenario 2: “bad” data
- New “Debug” Screen on ABORTs
- CA ADS Trace
  - Overview
  - Activating Trace
  - Scenario 1: Extended Run Units
  - Scenario 2: No extended RUs
- Summary



Session Summary

- **CA ADS Alive** is a very powerful tool for debugging CA ADS Dialogs
  - Generate with symbol tables
  - “Animate” the dialog – establish breakpoints
  - Run the application
- **CA ADS Trace** can be extremely useful when trying to see “the big picture”
- You need to know how to use **OLP** to view the output from CA ADS Trace - and from SNAP statements that are hard coded in your dialogs

Thanks !

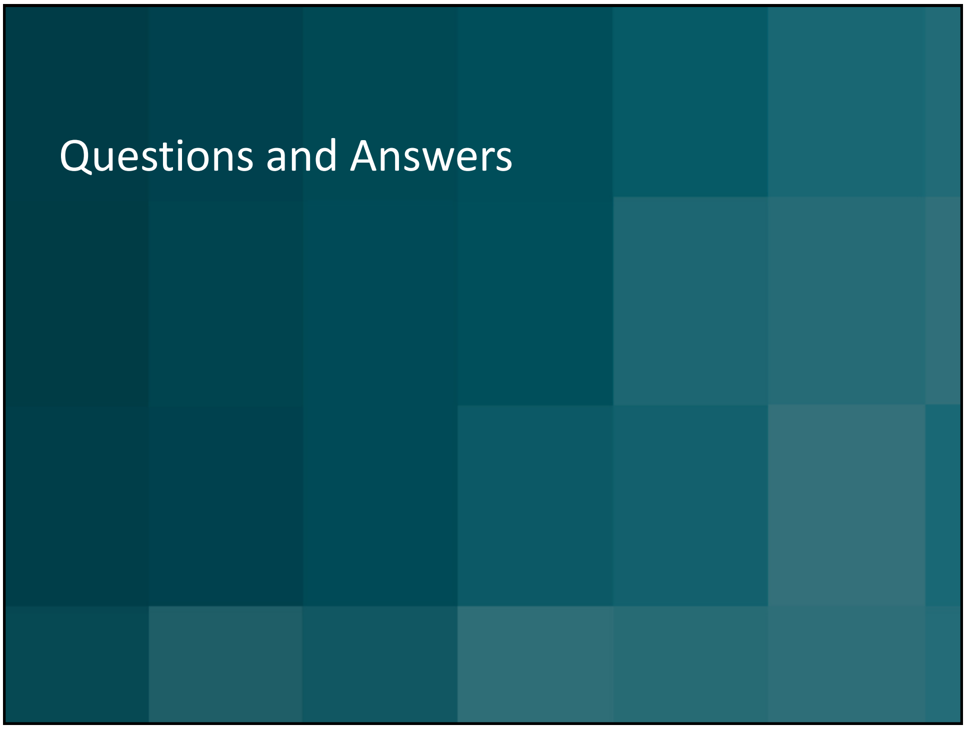


Online Session Evaluation

Please provide your feedback about this session: A9

On the CA Communities web site:
http://communities.ca.com

More details in your conference bag



Questions and Answers