Discovering OMEGAMON

Volume 2

OMEGAMON DB2 V5.3.0 –

Enhanced 3270 User Interface and Classic Interface Lab Exercises

IBM

Contents

Overview

OMEGAMON XE for DB2 provides the ability to monitor DB2 subsystems, both in a datasharing and non-datasharing configuration. OMEGAMON XE for DB2 v5.3.0 includes a choice of user interfaces: Classic, CUA, enhanced 3270 and Tivoli Enterprise Portal Server (TEPS). This choice of user interfaces provides the user with option of using the optimal interface for a given technical requirement. This series of exercises will illustrate several of the features and functions available in the enhanced 3270 user interface and the Classic 3270 interface specific to monitoring DB2.

Introduction

This lab will demonstrate how to utilize OMEGAMON XE for DB2 V5.3.0. In this lab's exercises, the user will perform a series of exercises focused on the following:

- Monitoring critical DB2 resources
- Analysis of DB2 lock conflicts
- DB2 application thread analysis
- Analysis of DB2 History
- Analysis of DB2 application traces

Icons

The following symbols appear in this document at places where additional guidance is available.

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step, but is helpful or good to know.
	Trouble-shooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

Lab #1 Enhanced 3270 User Interface OMEGAMON DB2 Scenario Walkthrough

Introduction

This lab exercise will demonstrate how to logon, navigate, and use some of the primary features of the OMEGAMON DB2 V5.30 enhanced 3270 user interface. In this lab the user will perform a series of scenarios focused on the following:

- DB2 Sysplex level resource monitoring
- DB2 Group Object I/O analysis
- DB2 subsystem level monitoring
- DB2 thread activity analysis
- DB2-CICS thread activity analysis
- Navigate to another OMEGAMON z/OS product menu

1.1 Logon to e3270ui

a) Enter the User ID and Password provided by the instructor and Press ENTER

Logon information will be provided by the lab instructors.

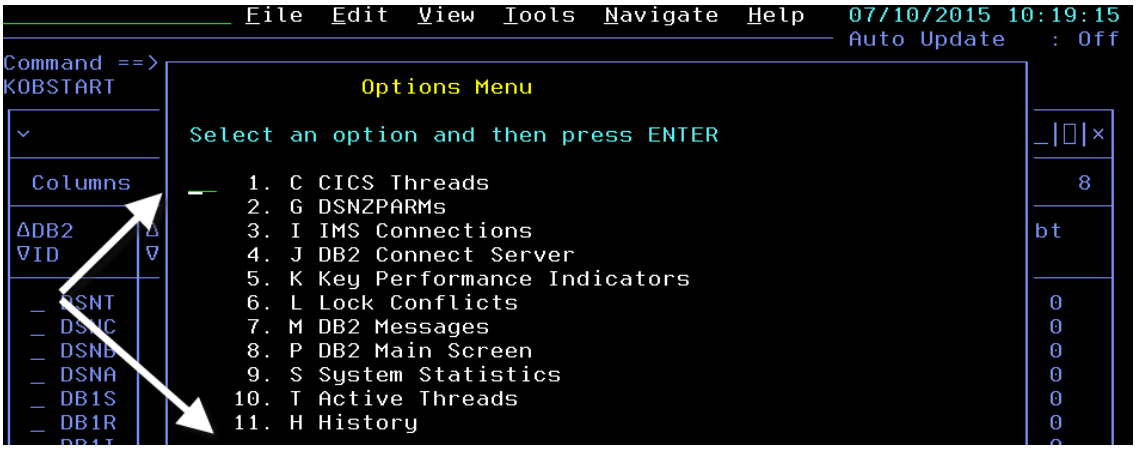
1.2 Overview from the DB2 Plex Perspective

The following is the e3270 ui (panelid KOBSTART). This panel contains overview information for whatever OMEGAMON monitoring agents are connected to the enhanced 3270 user interface (in this example you see z/OS, CICS, and DB2). From this panel you may scroll, filter, or drill down for additional information from any of the various monitoring agents.

First we will perform a brief overview of DB2 resource monitoring at the Sysplex/data sharing level now available in the new enhanced 3270 interface. Beginning from the KOBSTART panel we see a Sysplex level view of the enterprise.

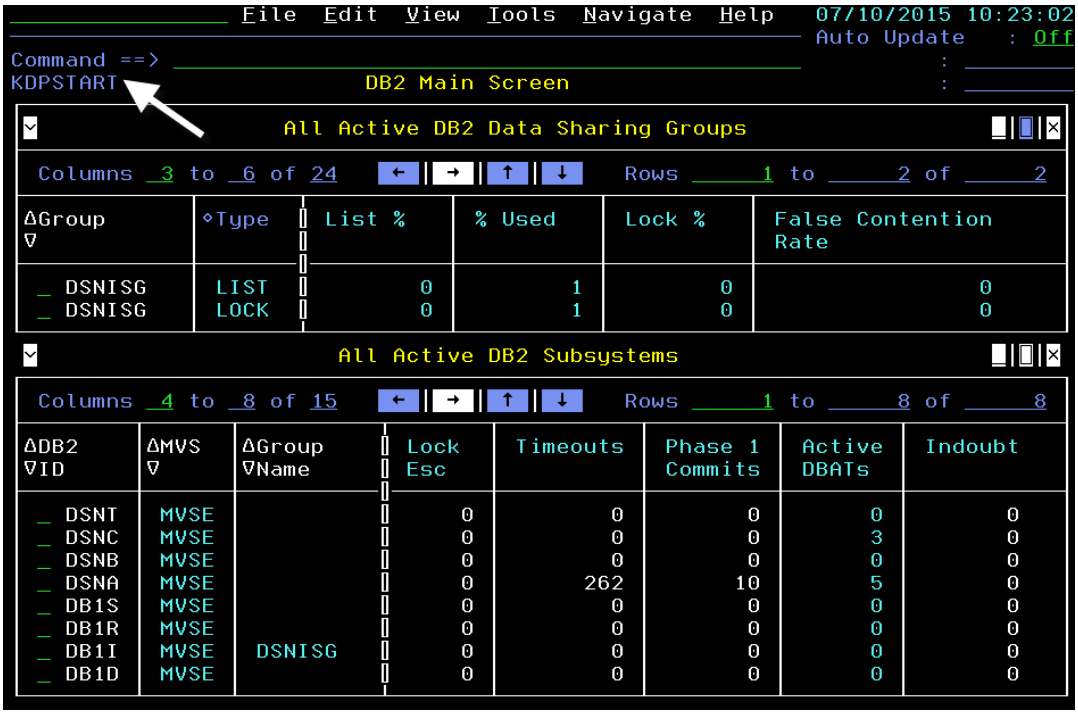
We will look at a few of the drill down options from this panel.

a) **Position the cursor** by the one of the DB2s (DB2 ID), enter / and **Press Enter**. You will then see a popup with several navigation options.



From here you can look at several categories of DB2 level resources (Threads, Locks, Subsystem, Messages, History, and more). You also have option P to navigate to the OMEGAMON DB2 Main Screen.

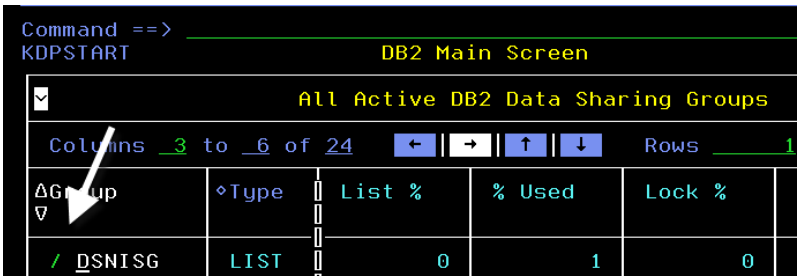
b) Enter **P** for DB2 Main Screen in the option popup and **Press Enter**.



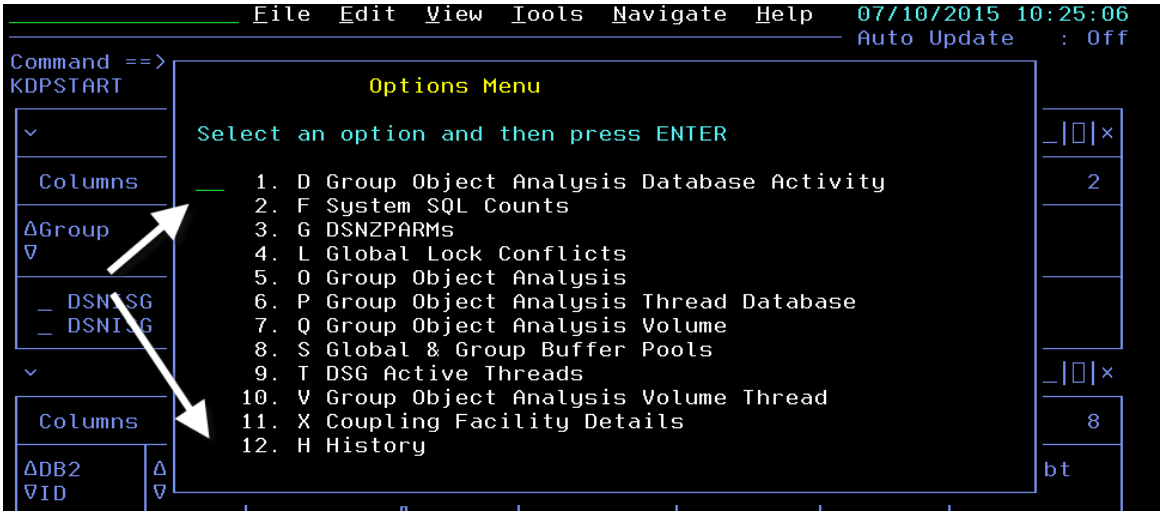
You are now looking at the DB2 Main Screen (KDPSTART). The DB2 Main Panel consists of two portions. The top portion shows the active data sharing groups in the environment (in this example DSNISG). The bottom portion of the screen shows the active DB2 subsystems.

First, let's explore the DB2plex level information available in the enhanced 3270 ui.

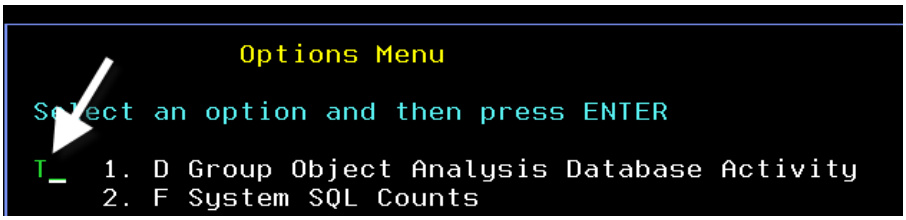
c) **Position the cursor** by the Data Sharing Group, enter / and **Press Enter**.



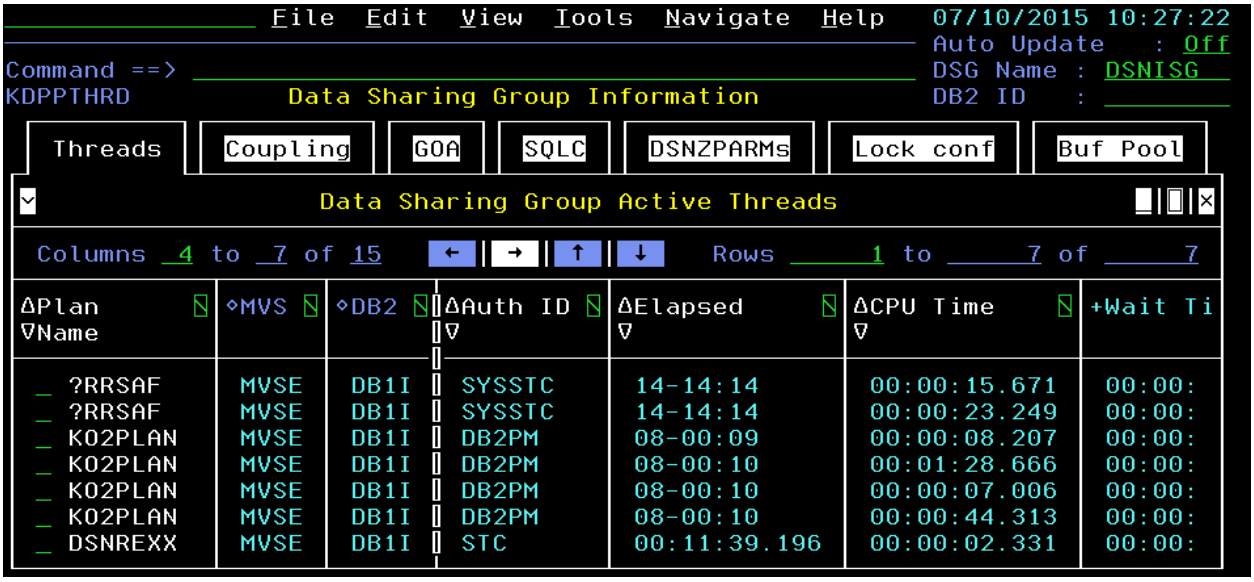
You are now looking at the Options Menu for DB2plex information. From here you can look at DB2plex level resources, such as Global Locks, Coupling Facility, Group Object Analysis, and a data sharing group level view of DB2 threads.



d) Enter **T** for DSG Active Threads in the option and **Press Enter**.

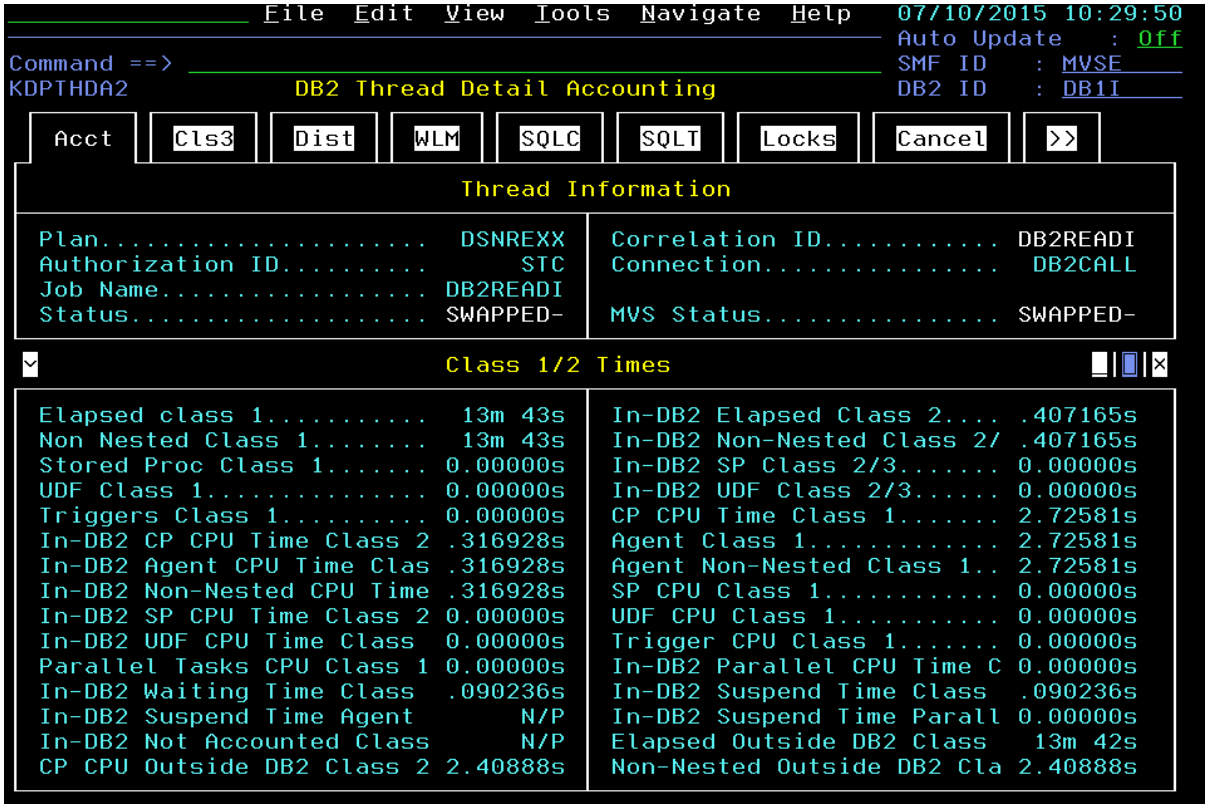


You are now looking at the active threads in the data sharing group. You may scroll the panel to the right to see additional information for the various threads active in the group. You may also drill in on one of the threads for additional detail. You may also click on the various tabs to see other data sharing level information.



e) **Position the cursor** by one of threads (suggestion – look at DSNREXX), enter **S** and **Press Enter**.

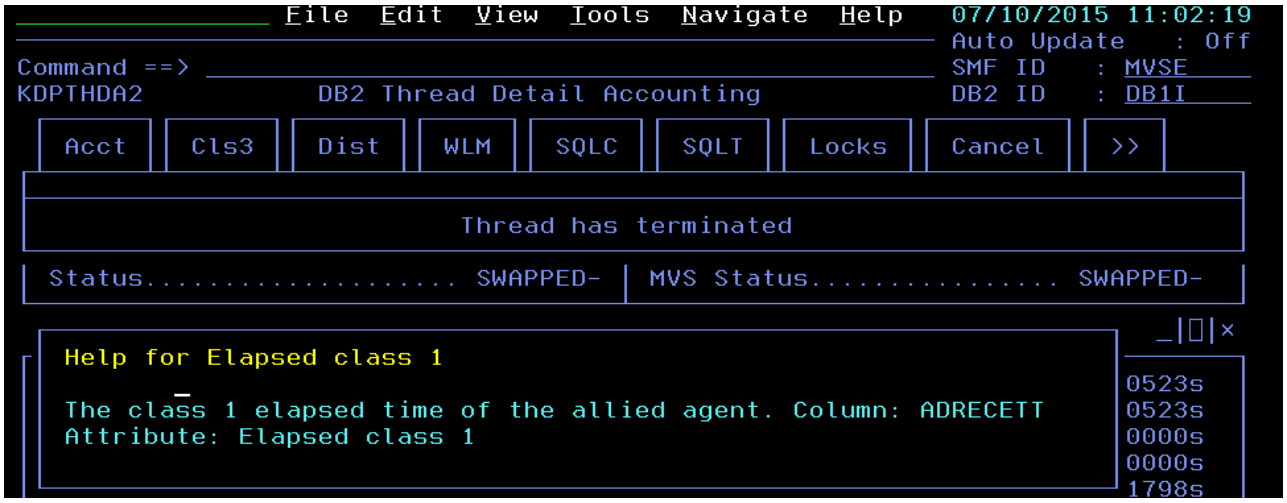
You are now looking at additional detail for one of the active threads in the data sharing group.



This panel shows many of the standard DB2 Accounting Class 1 and Accounting Class 2 counters for the thread. This includes such information as thread elapsed time, thread IN-DB2 time, and thread CPU time counters.

If you are unclear as to the meaning of a particular number, you may get help for the field.

f) To see this help information, **position the cursor** on one of the fields (such as Elapsed Class 1), and **Press F1**.



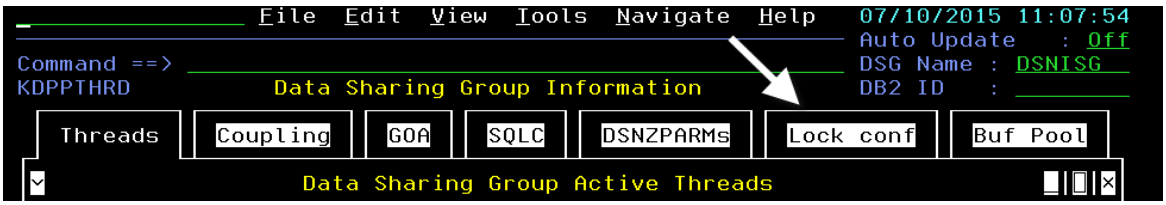
You now see help information, with an explanation of the meaning of the number.

Press F3 to make the help popup go away.

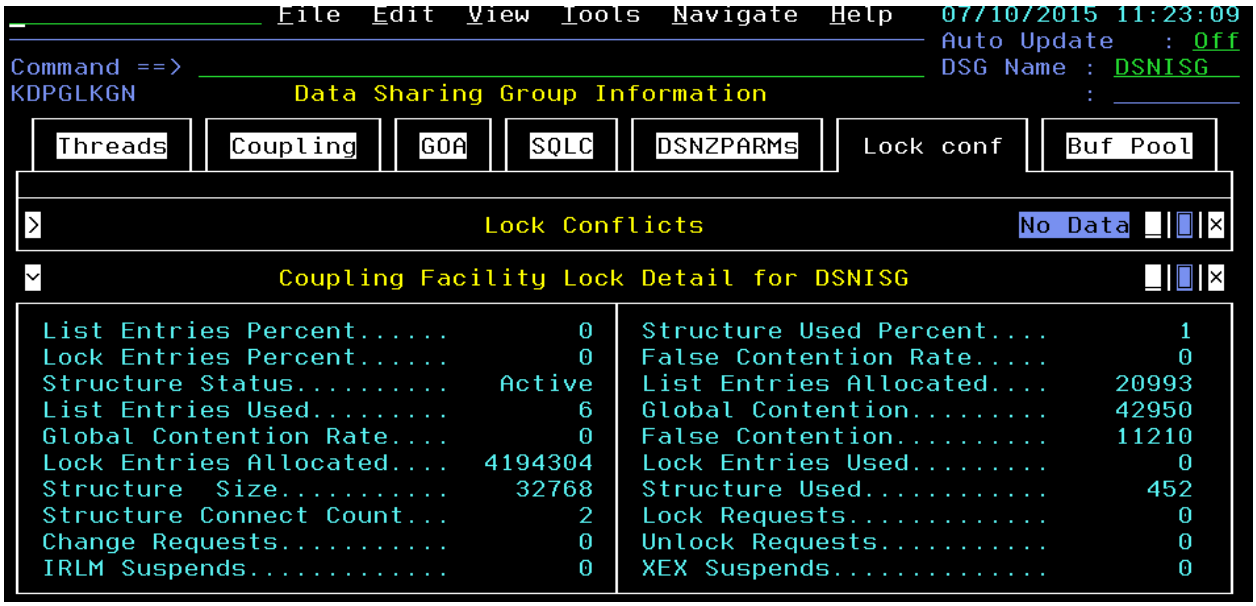
Now that you have seen the threads active in the data sharing group, you may want to more about the structures in the data sharing group.

g) **Press F3**, to return to the KDPPTHRD Data Sharing Group Active Threads panel. Note that from this panel you may use the tab navigation to look at other aspects of the data sharing group.

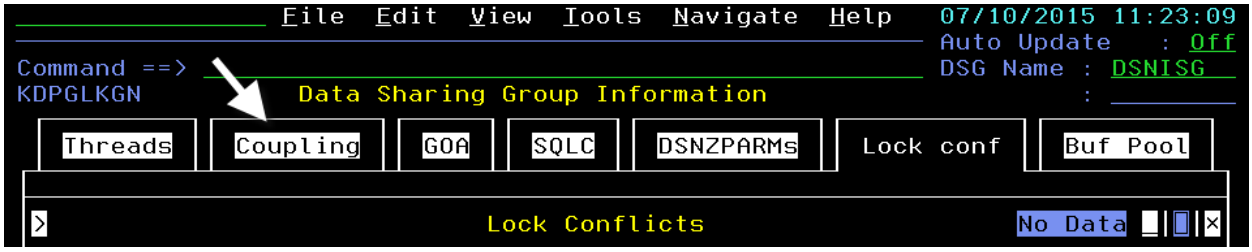
h) To look at lock and coupling facility information. **Position the cursor on the Lock conf tab** and **Press Enter**.



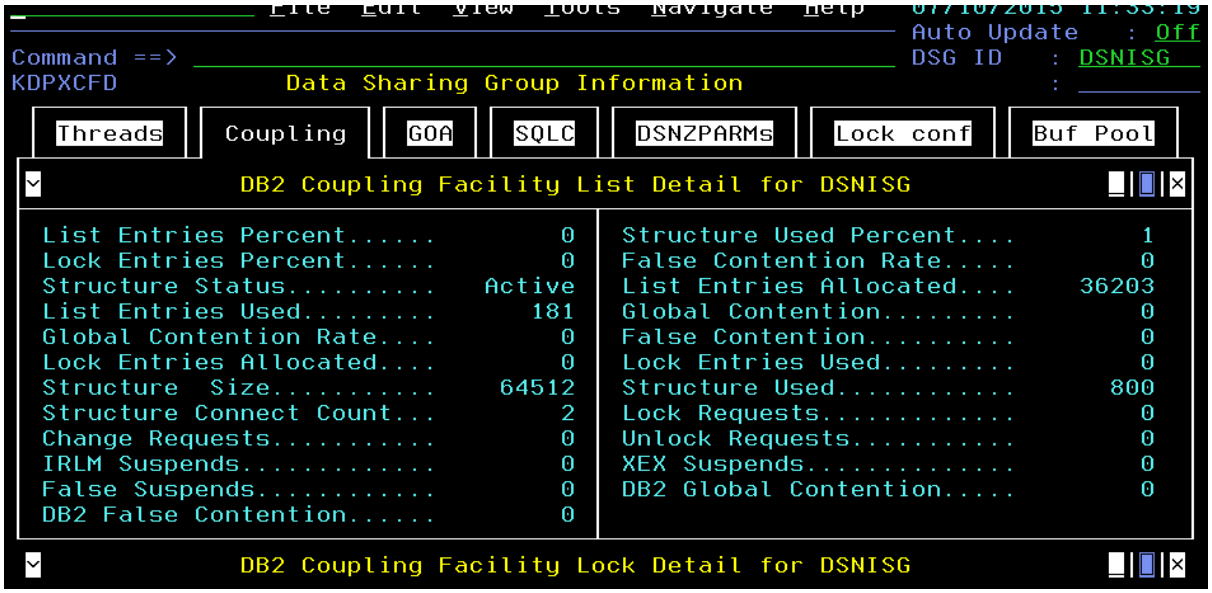
You are now looking at locking information for the data sharing group, including lock and unlock requests, and global and false lock contention.

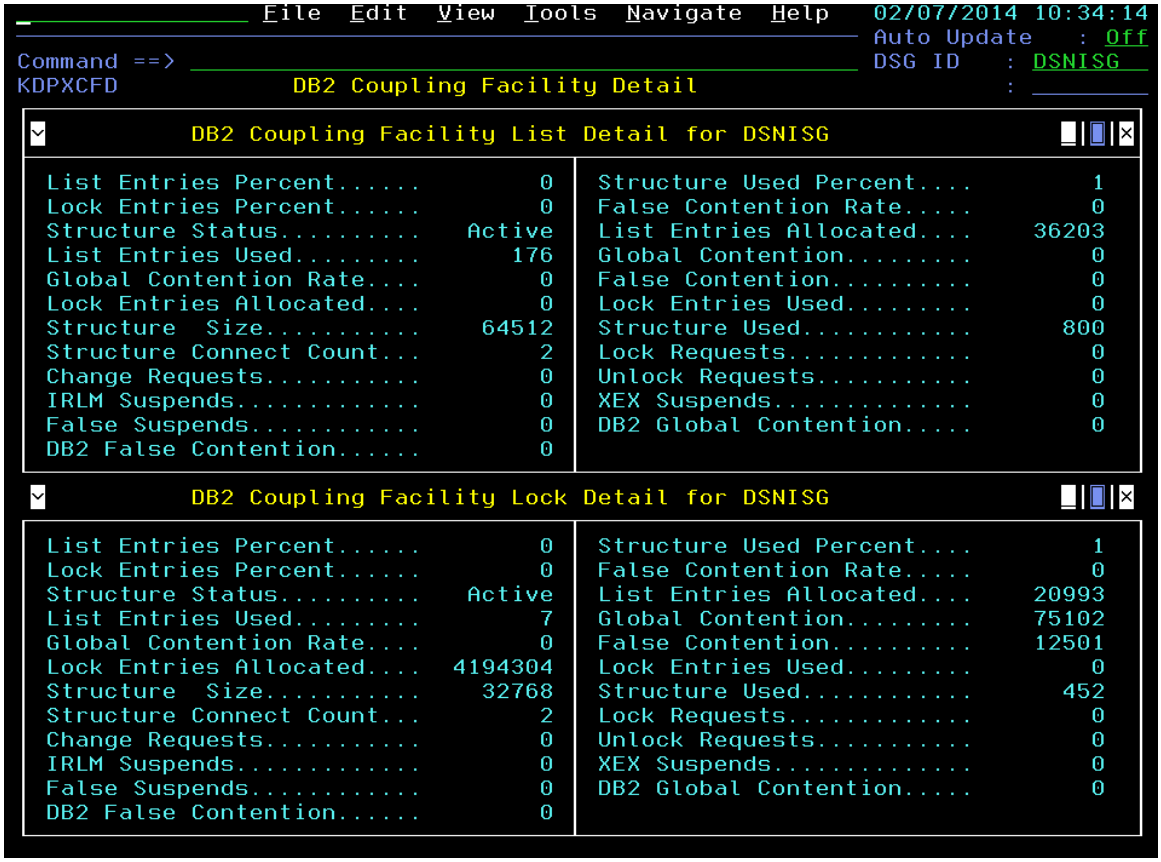


i) To look at lock and coupling facility information. **Position the cursor on the Coupling tab** and **Press Enter**.



You are now looking at the DB2 Coupling Facility lock and list structure details for the data sharing group. This includes information such as structure size, utilization, global contention, false contention, and number of requests to the structures.

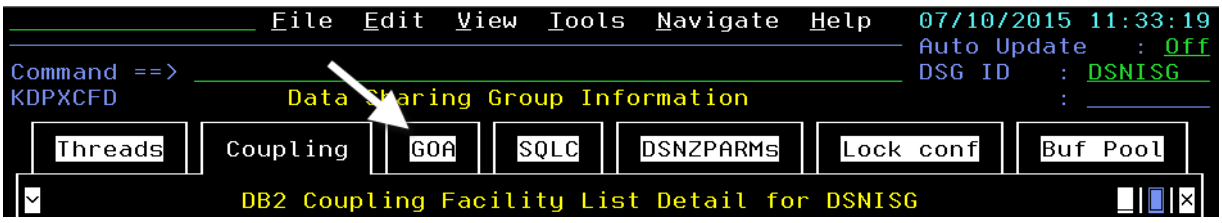




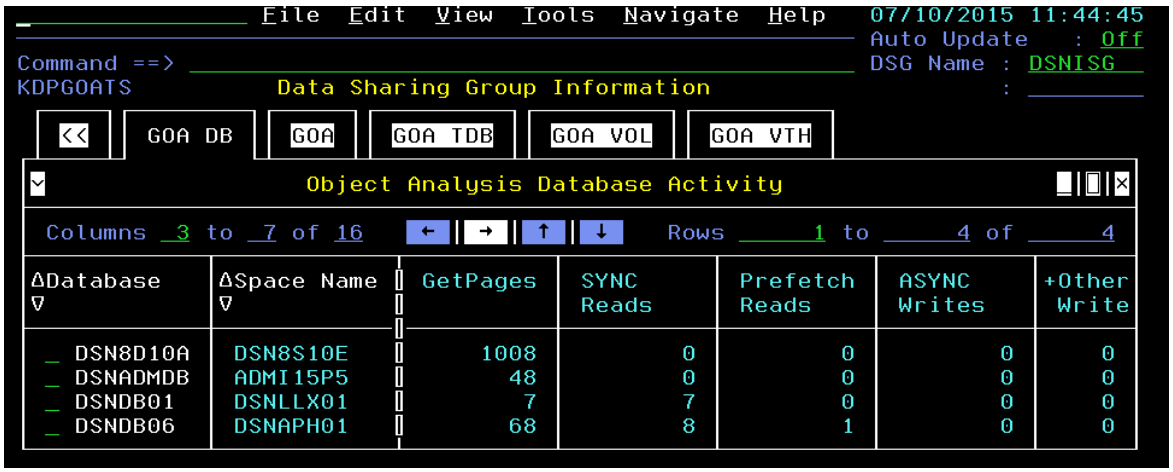
1.3 DB2 Group Object Analysis – I/O Analysis

Group Object Analysis is a facility of OMEGAMON DB2 that allows you to see DB2 Getpage and I/O activity in the DB2 data sharing group, and analyze that activity. This portion of the lab will look at how to use Group Object Analysis in the enhanced 3270 user interface.

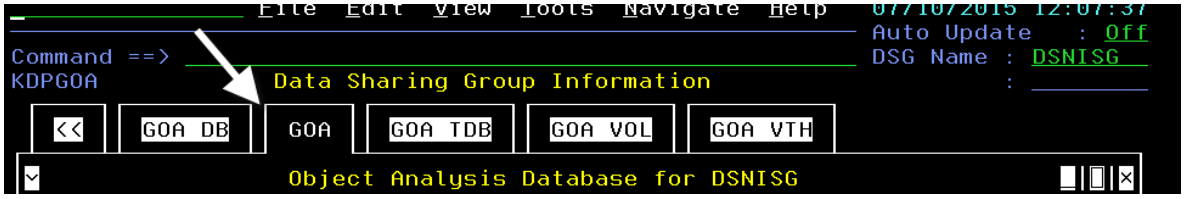
a) To see Group Object Analysis data **Position the cursor on the GOA tab** and **Press Enter**.



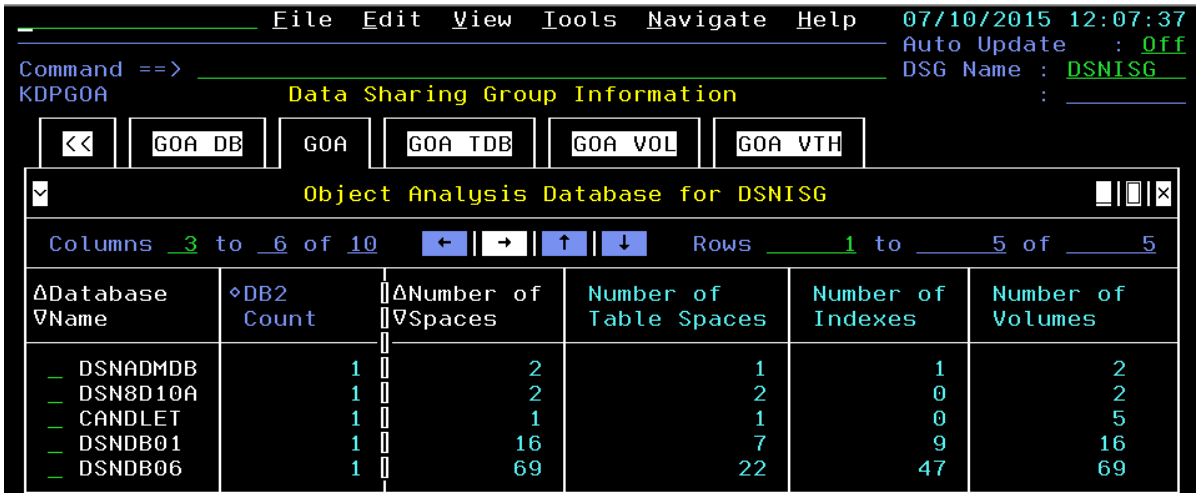
You are now looking at the Group Object Analysis database activity overview panel. This panel shows such information as Getpages and I/O counts by database and tablespace across the data sharing group.



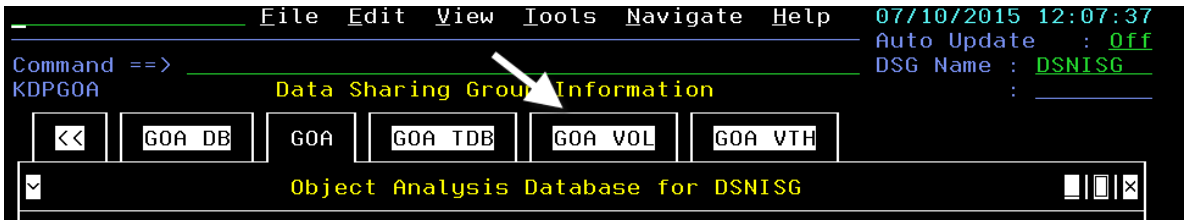
b) To see database overview information, such as number of objects by type, Position **the cursor on the GOA tab** and **Press Enter**.



You are now looking at the database overview for the data sharing group.



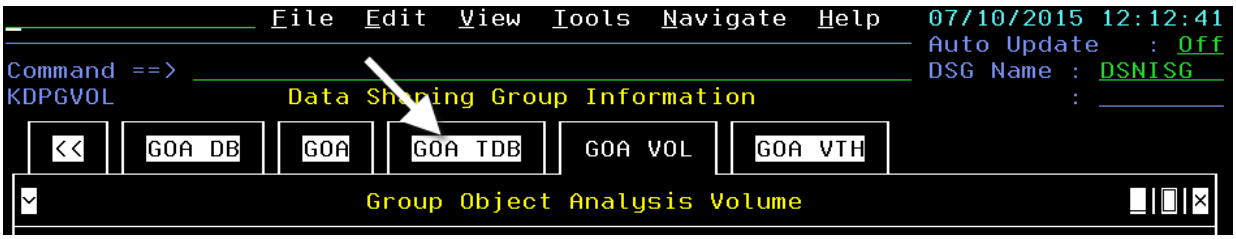
c) To see DASD volume activity for the data sharing group, Position **the cursor on the GOA VOL tab** and **Press Enter**.



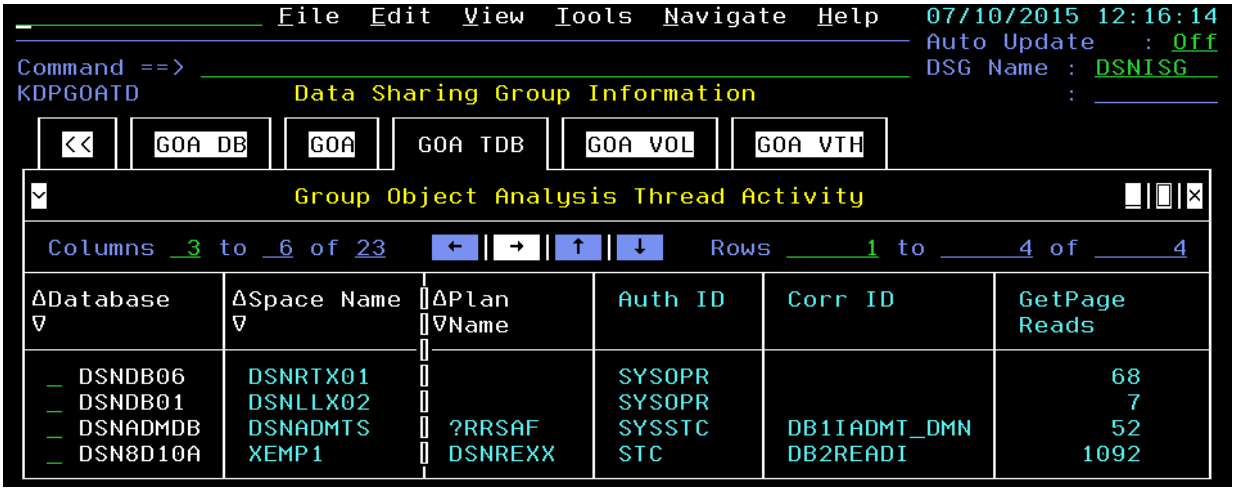
You are now looking at the DASD volume utilization for the data sharing group. This information includes utilization percent and millisecond response time (MSR time).

You may also see getpage and I/O activity correlated to the application threads running in the DB2 data sharing group.

d) To see DB2 thread activity for the data sharing group, Position **the cursor on the GOA TDB tab** and **Press Enter**.



You are now looking at the listing of database and thread activity for the data sharing group. This shows the object name and the thread plan name and Corrid for the thread.

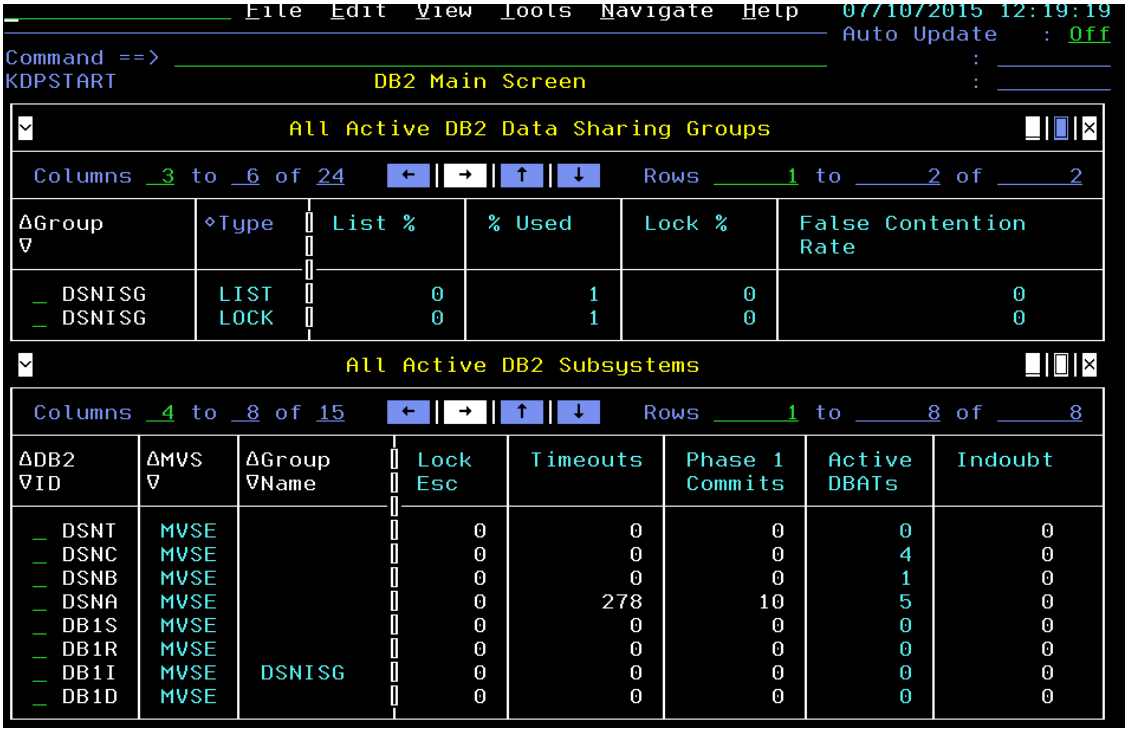


We have looked at the core elements of Group Object Analysis. You have seen how Group Object Analysis shows DB2 object information, analyzes getpage and I/O activity, and correlates that information by object and by thread. Feel free to look at some of the other panel options.

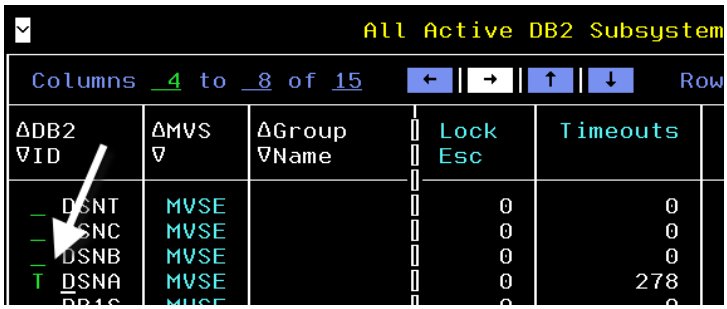
e) When you are finished, **Press F3** until you **return to the KDPSTART** DB2 Main Screen.

1.4 DB2 Thread Analysis – Long Running Threads

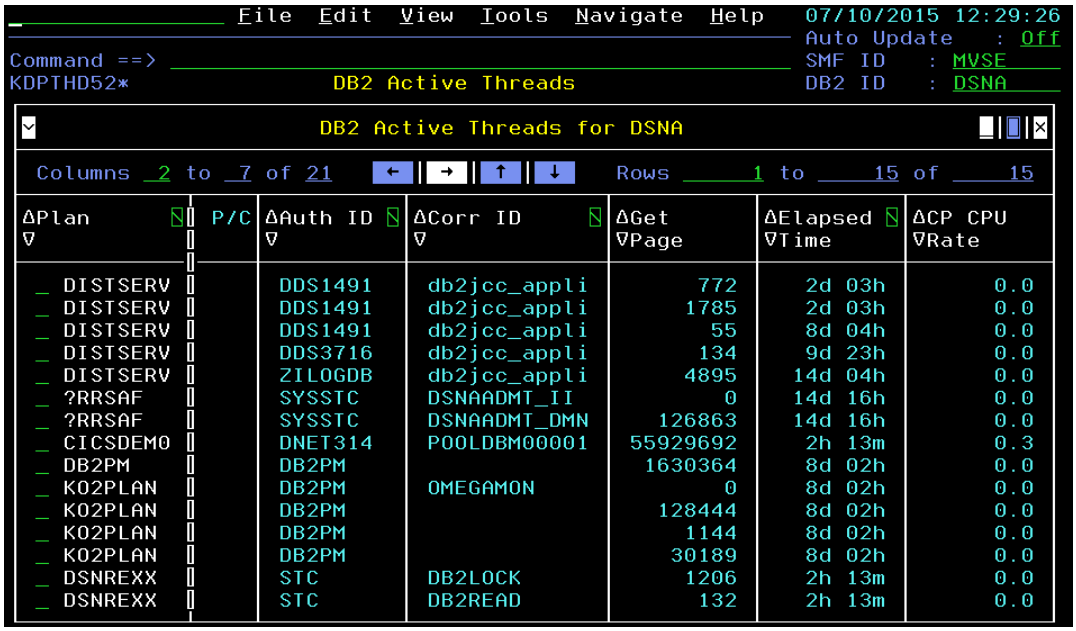
You have looked at how you can display DB2 threads in a data sharing group. Now let's look at some more detailed thread analysis scenarios. Let's begin at the DB2 Main Screen (KDPSTART). The bottom portion of the DB2 Main Screen shows a line of information for each monitored DB2 subsystem. To get information on a particular DB2 subsystem you use the standard navigation process we've demonstrated earlier in this lab exercise.



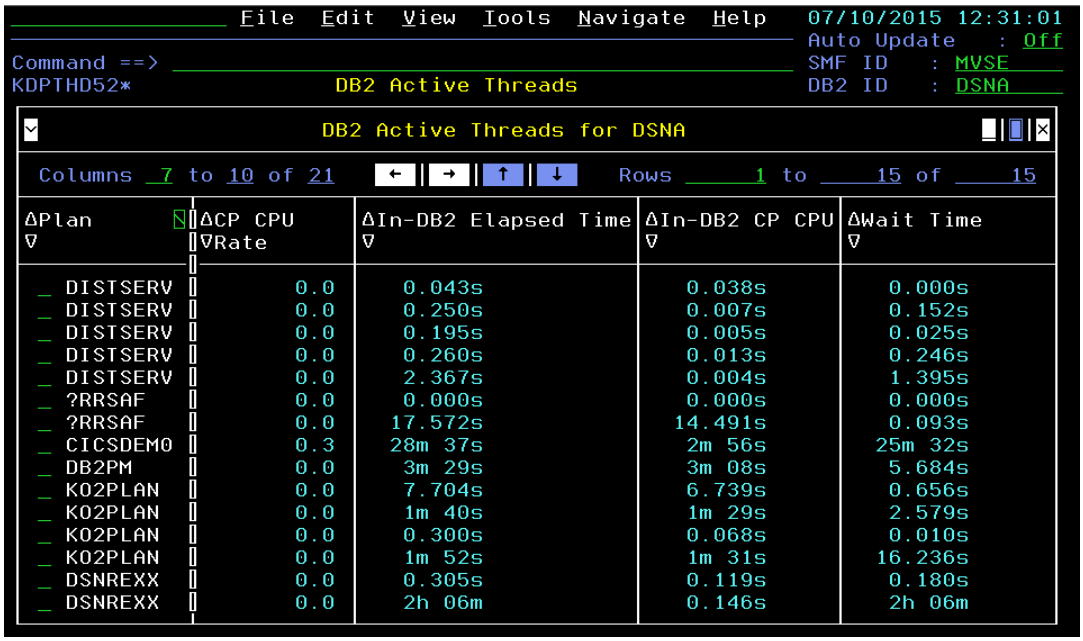
a) P**osition the cursor** next to the DB2ID (in this example DSNA), enter **T** for active threads and **Press Enter**.



You are now looking at the Active Threads display for DB2 subsystem DSNA. Try shifting the screen to see additional thread information in the active thread overview (**Press F11** or use the arrow navigation option). Note that you will see many of the more relevant DB2 counters (Authid, CP rate, various IN-DB2 times, Getpage count, reads and prefetch counts, workstation name, and more). How may times you shift the screen will depend upon the resolution of your particular terminal screen.

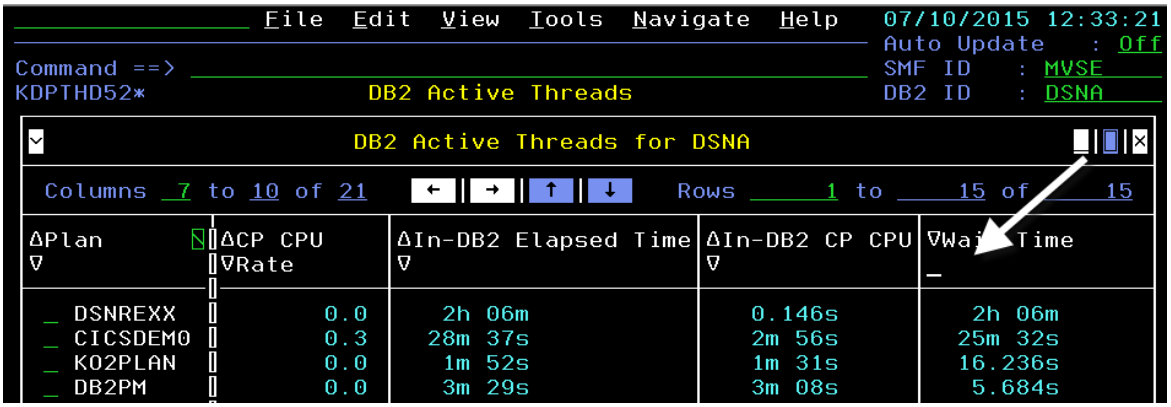


b) Once you have looked at the available columns, **shift the screen** to where you can see In-DB2 Elapsed Time, Wait Time, and DB2 Status. The screen should appear similar to below.



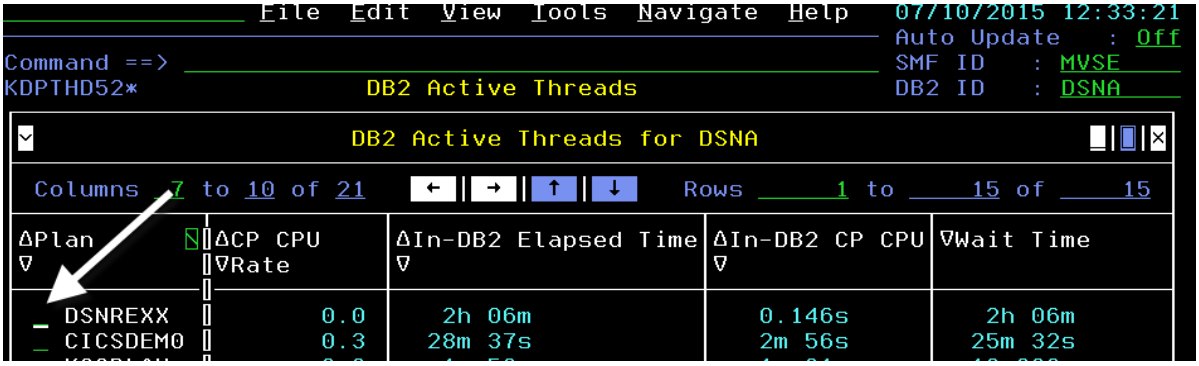
The Wait Time column is a summary of the various DB2 wait times for the DB2 thread (including I/O, Lock/Latch and various other waits).

c) **Position the cursor** on the sort arrows for Wait Time and **Press Enter** to sort by Wait Time.



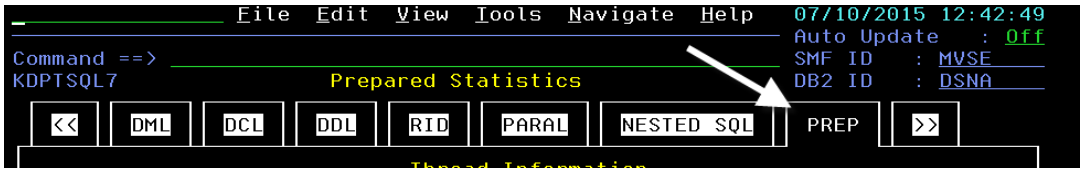
You are now looking at Active Threads display, sorted by Wait Time.

d) To see more detail about the thread, **position the cursor** by the thread (DSNREXX) and **Press Enter**.

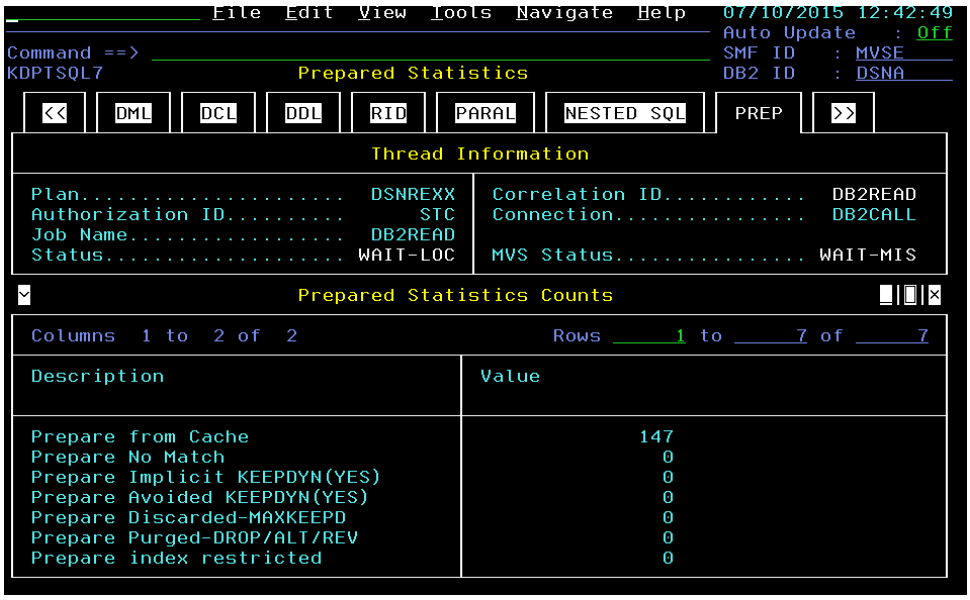


You are now looking at the Thread SQL information. This panel shows the various counts and indicators showing what SQL has been executed by the thread. Note that there are multiple tabs highlighted at the top of the panel. Each tab represents different categories of thread information. Try placing the cursor on different tabs and press enter to see additional thread and SQL information for the thread.

e) To see dynamic SQL information **Position the cursor on the PREP tab** and **Press Enter**.

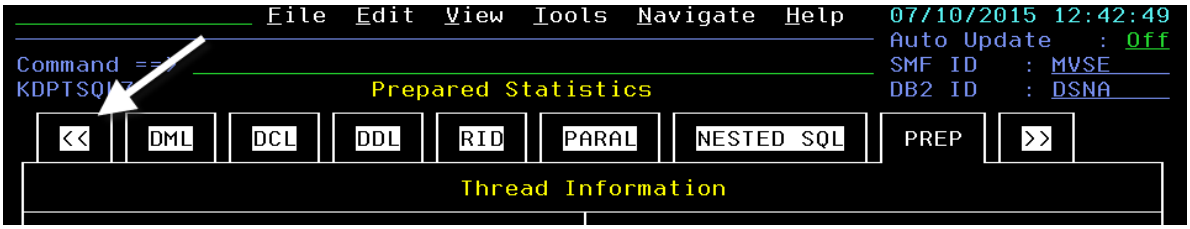


You are now looking at counts of Prepare activity for the DB2 thread.

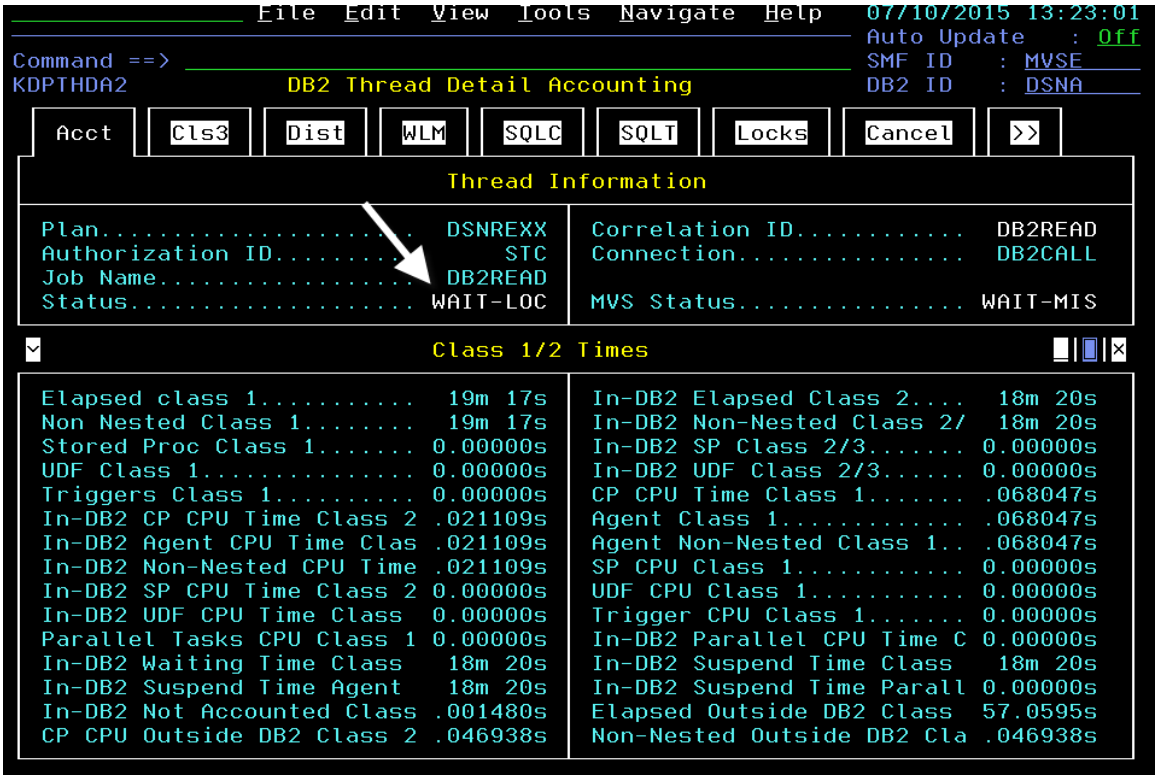


You will want to look at other thread specific information, such as the DB2 Accounting information that shows thread elapsed, IN-DB2, and CPU times.

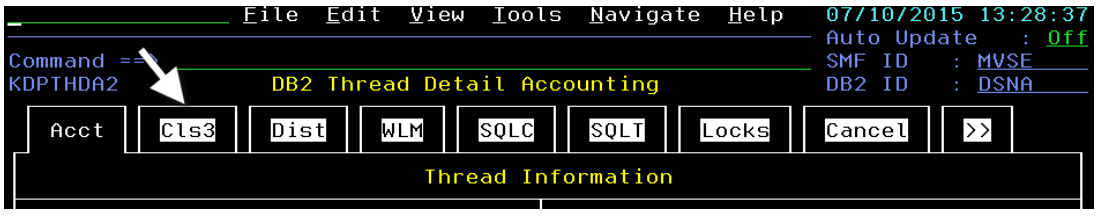
f) To see thread accounting **Position the cursor on the << tab** and **Press Enter**.



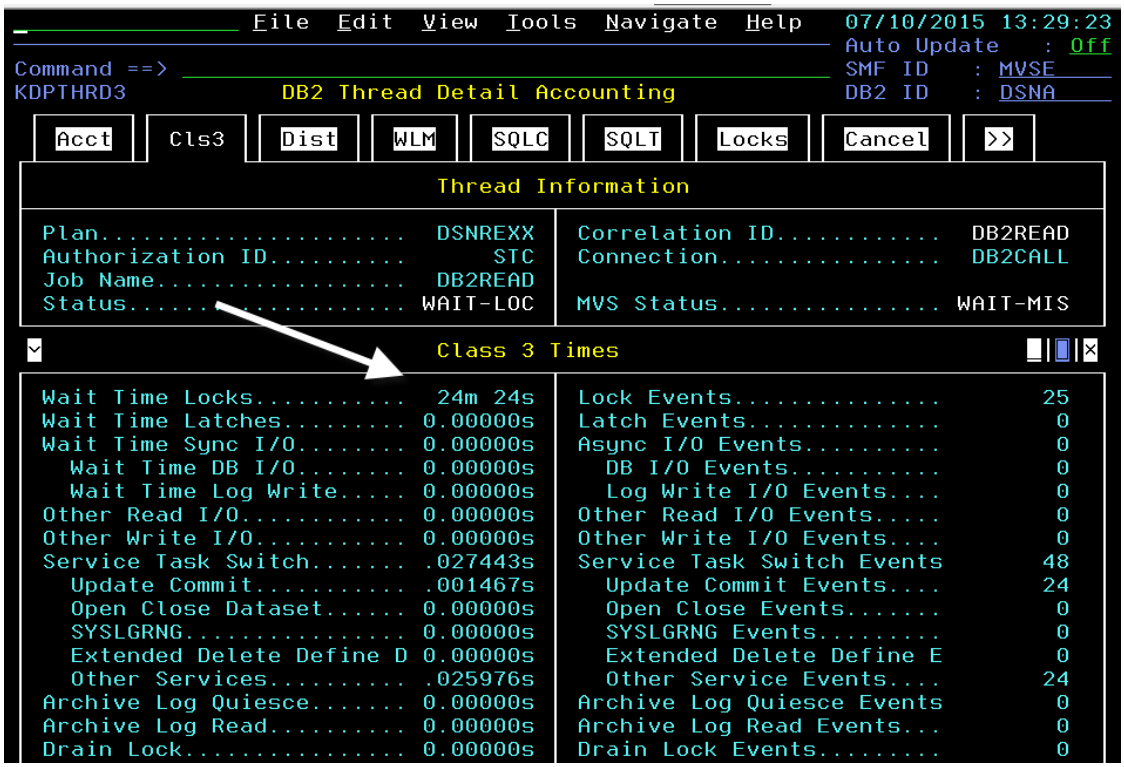
You are now looking at the Thread Accounting display. This panel shows the thread elapsed, IN-DB2, and CPU information. Note the thread status of WAIT-LOC which indicates the thread is waiting for a lock.



g) To see more information on what the thread is waiting for, pposition **the cursor on the Cls3 tab** and **Press Enter**.

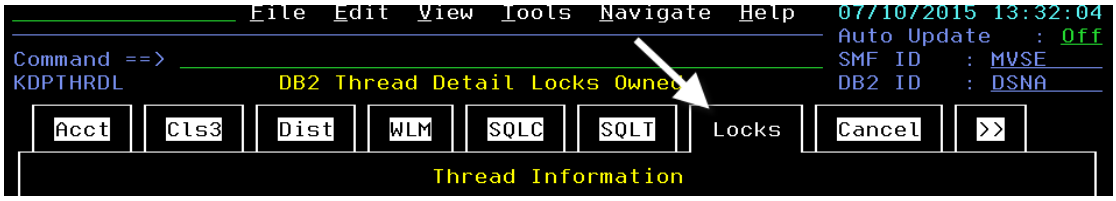


You are now looking at the DB2 Class 3 wait time display. This panel shows the number of wait events by type, and time for the waits.

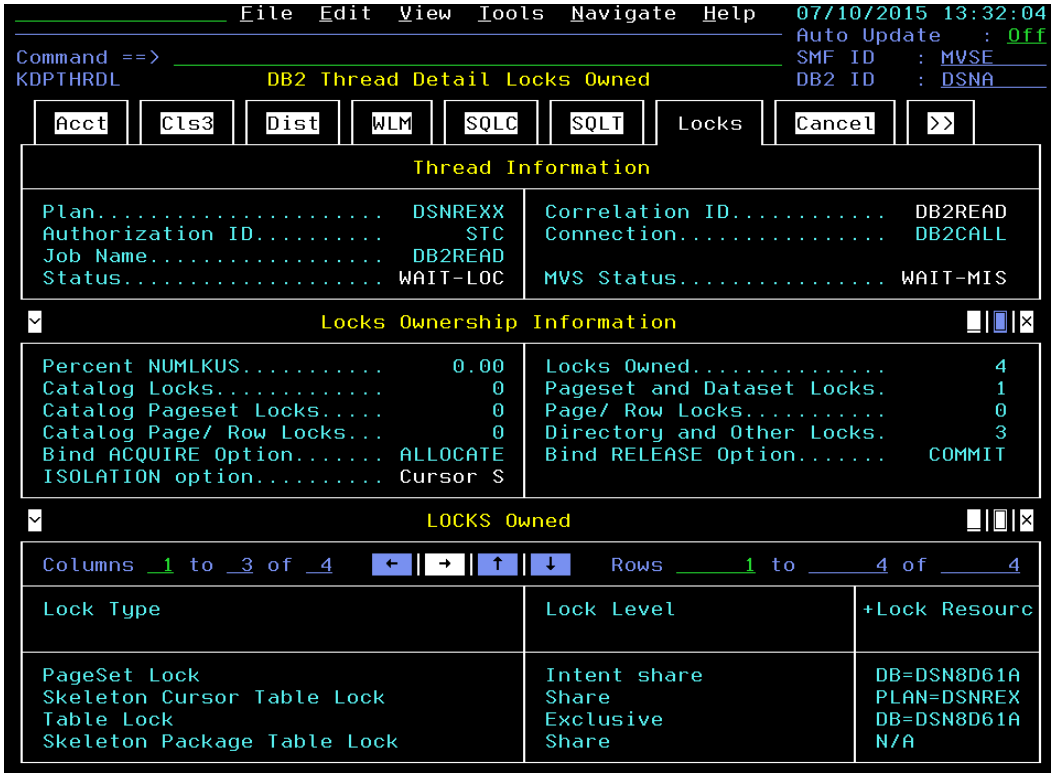


Note that in the example the major wait reason is Wait Time Locks. In other words the thread is wating for locks.

h) To see more information on what DB2 resources are locked, position **the cursor on the Locks tab** and **Press Enter**.

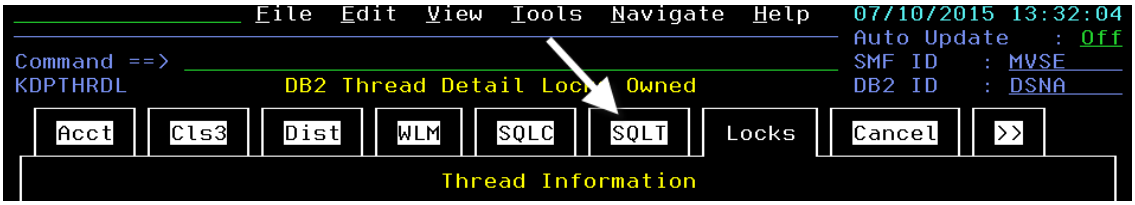


You are now looking at the thread lock detail for the thread. This panel shows what objects the thread has locked, what is the lock type and intent level of the locks.

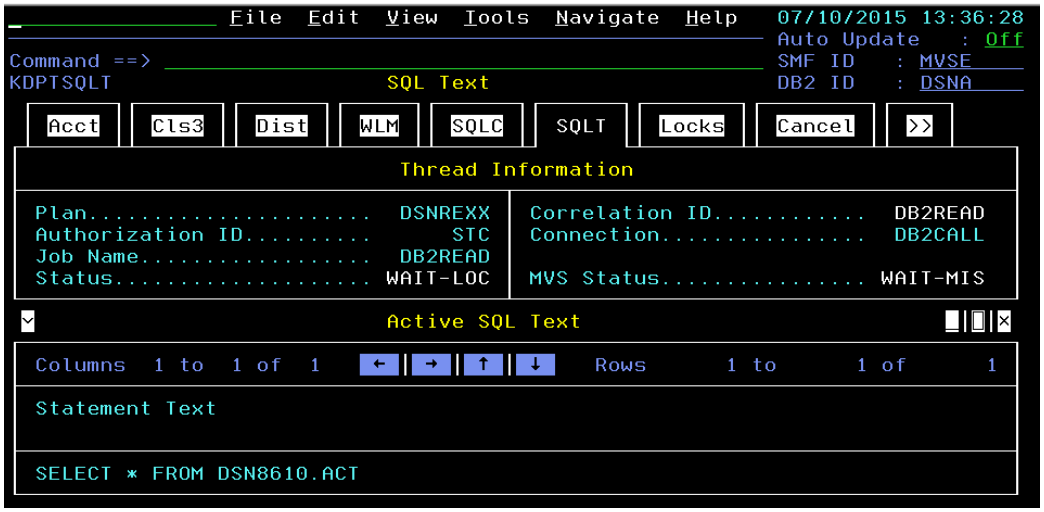


You may also want to see the SQL call being executed by thre thread.

i) To see the SQL call text, position **the cursor on the SQLT tab** and **Press Enter**.



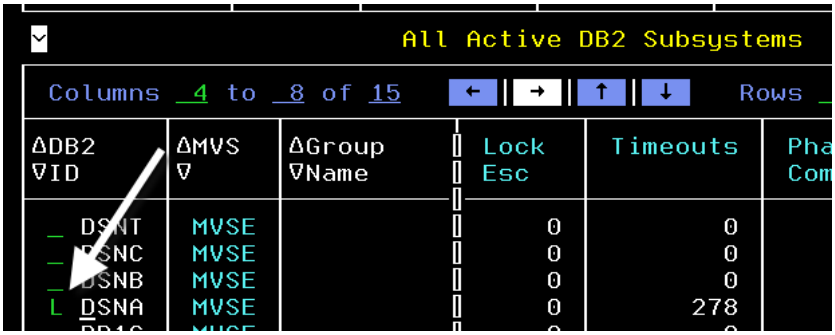
You now see the SQL call text being executed.



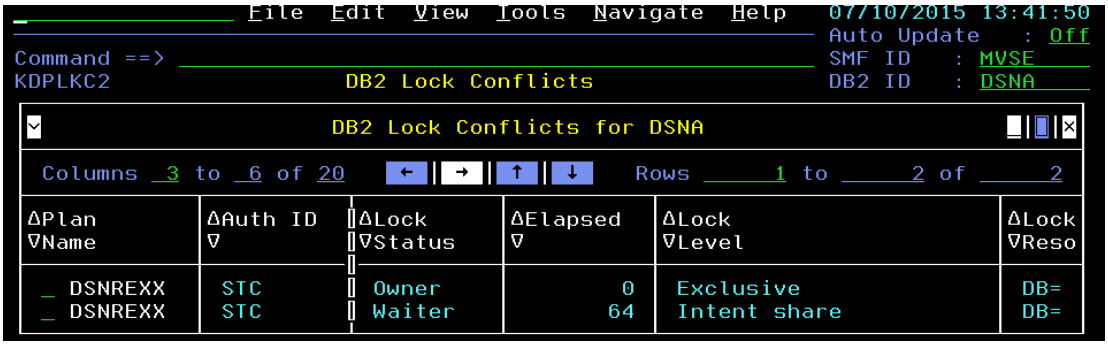
An indication of high DB2 lock wait time indicates the high likelihood of DB2 lock conflicts. Now that you know that lock waits is the main wait reason for this thread, you may investigate the source of the lock conflicts.

h) **Press F3**, then **Press F3** again. You should be at the KDPSTART DB2 Main Screen.

i) P**osition the cursor** next to the DB2ID (in this example again choose DSNA), **enter L** for lock conflicts and **Press Enter**.

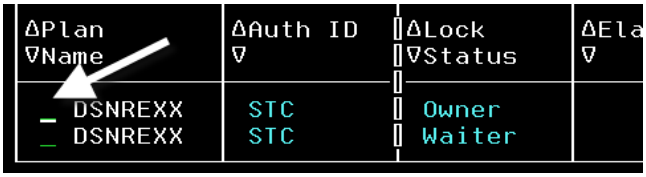


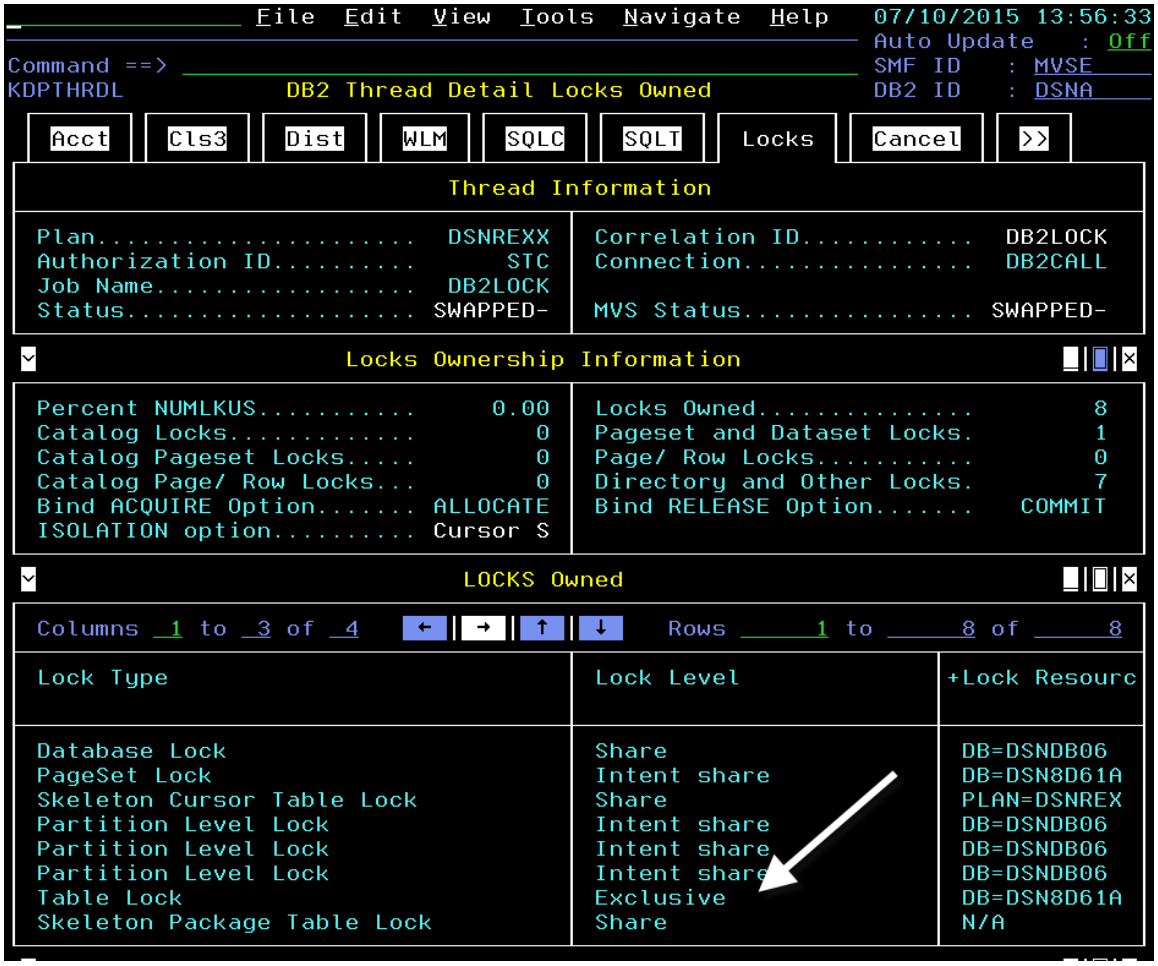
You are now looking at the Lock Conflict display for DB2 subsystem DSNA. This display will show the lock owner and the lock waiter for the lock conflict, and will show the DB2 resource in the conflict (which you can see in the Lock Resource column).



Note which thread is the owner, and which thread is the waiter, look for a column that shows Lock Status. From this display you see that the conflict is between two batch applications, each running the DSNREXX plan. Now let's look at what the other batch thread is doing.

j) **Position the cursor** next to the Owner thread and **Press Enter**





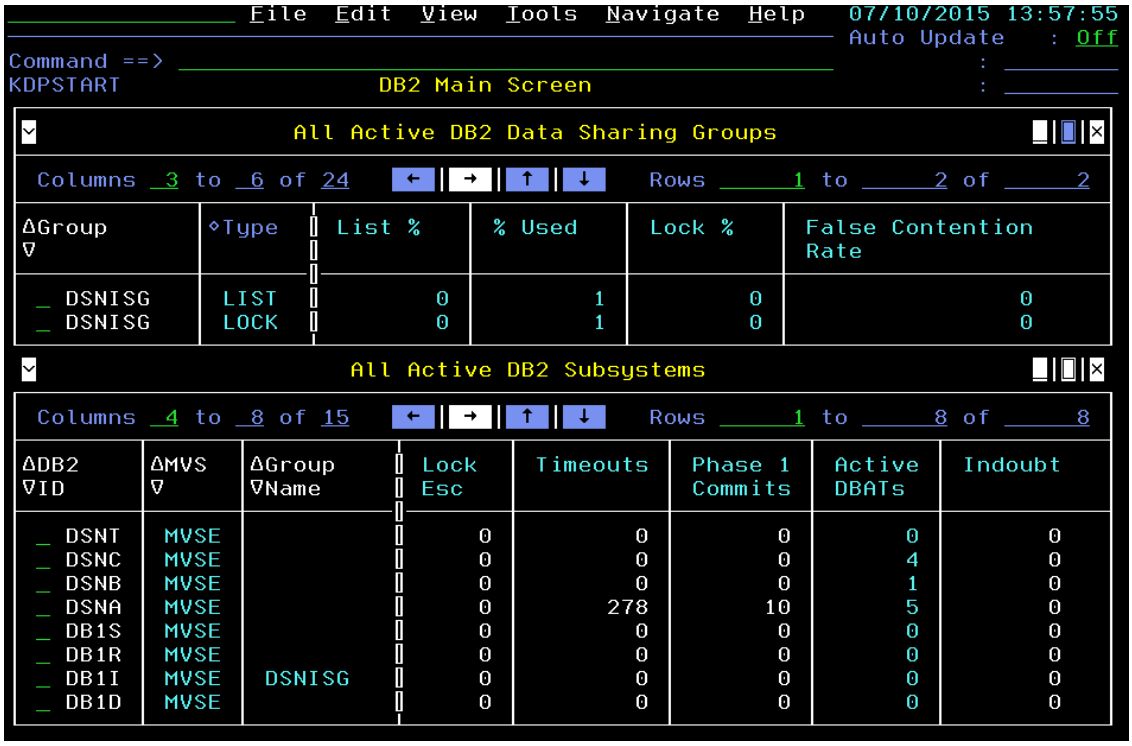
Note that this thread is performing holding an exclusive level lock. You also see the resource locked to the right.

k) **Press F3**, then **Press F3** again. You should be at the KDPSTART DB2 Main Screen.

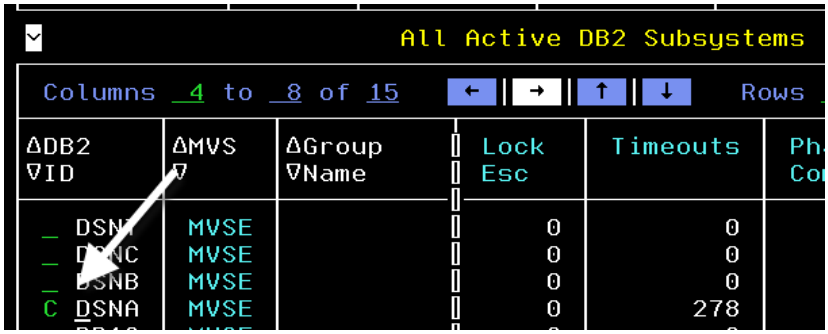
1.5 DB2 Thread Analysis – CICS Threads

In the prior thread analysis scenario you looked at batch threads and a lock conflict analysis scenario. Now, using OMEGAMON DB2 enhanced 3270 user interface, you can also look at CICS threads connected to DB2.

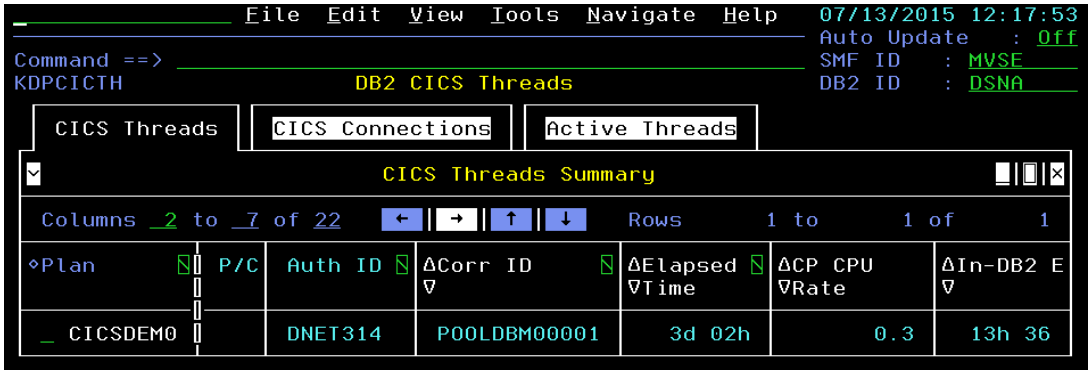
Begin at the KDPSTART DB2 Main Screen panel.



a) **Position the cursor** next to the DB2ID (in this example DSNA), **enter C** and **Press Enter**.

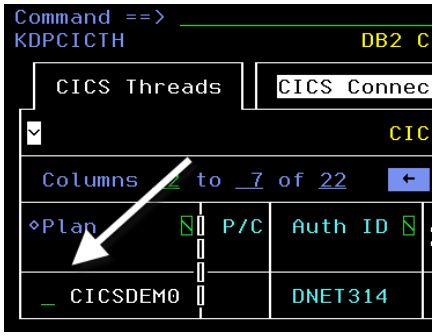


You are now looking at a summary of DB2 threads from various CICS regions connected to DB2 subsytsem. From this display you see summary detail for each thread.



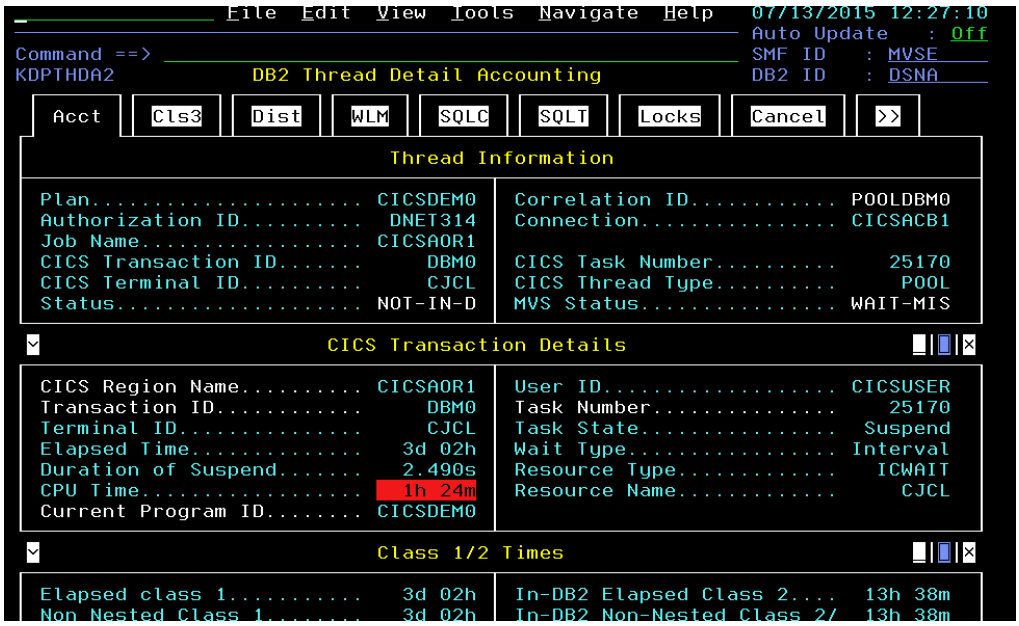
You may also drill down into the thread to see additional thread detail.

b) **Position the cursor** next to the thread and **Press Enter**.



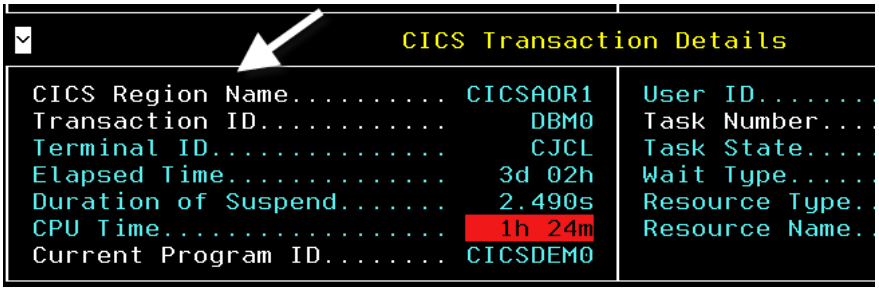
You are now looking at the Accounting detail for the thread. You have seen examples of this information in prior exercises.

Note the CICS region name and CICS Transaction id being used by this DB2 thread (in this example the transaction ID is DBM0).



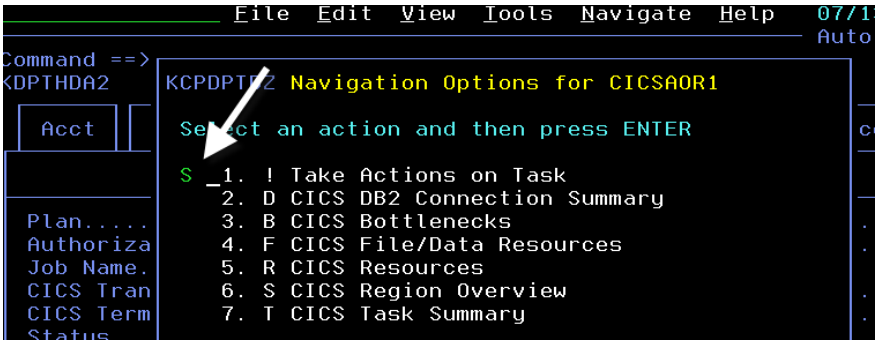
The enhanced 3270 user interface offers a feature called embedded data. Embedded data allows for easy cross component navigation across various OMEGAMON monitoring components. In this example you may navigate from the DB2 thread display to the OMEGAMON CICS region information.

c) To see the CICS region information**, Position the cursor on the CICS Region Name label and Press Enter.**

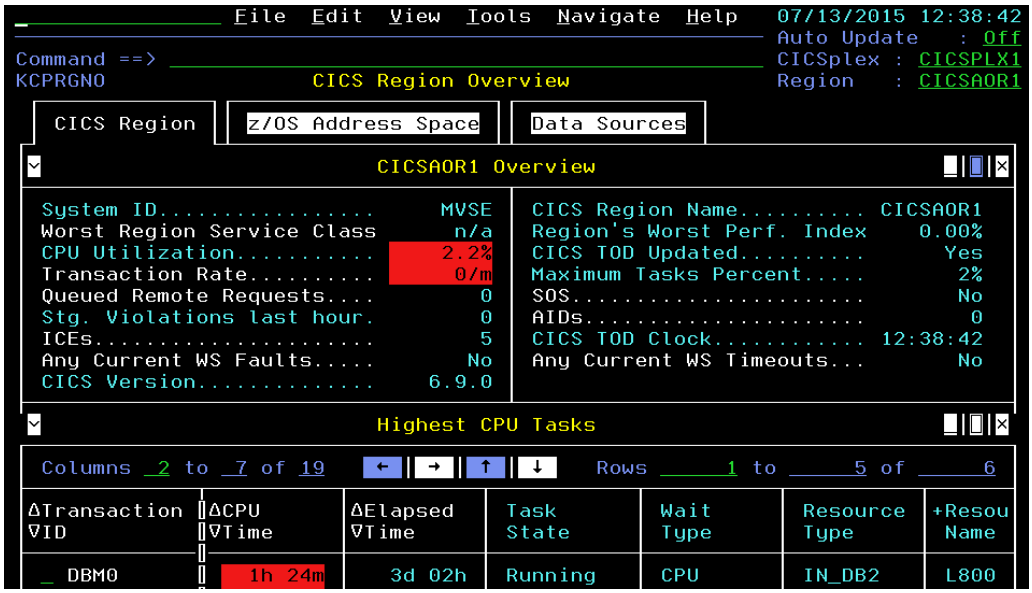


You will be presented with a set of options from a popup.

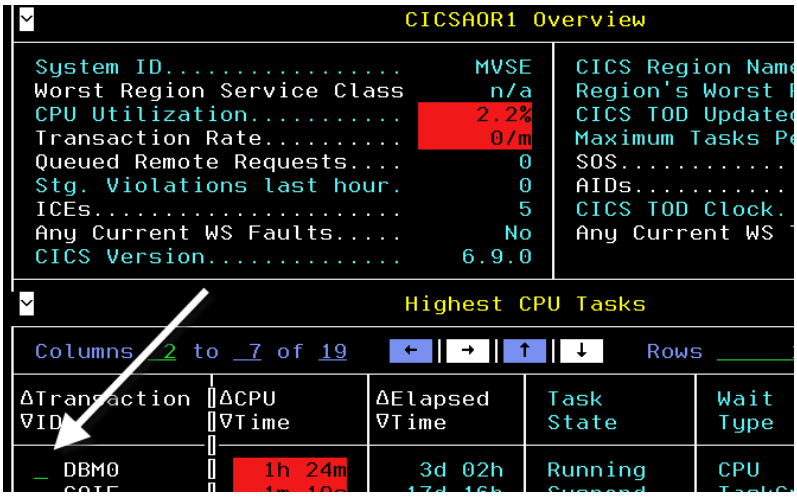
d) From the popup **Select option S** for CICS Region Overview and **Press Enter**.



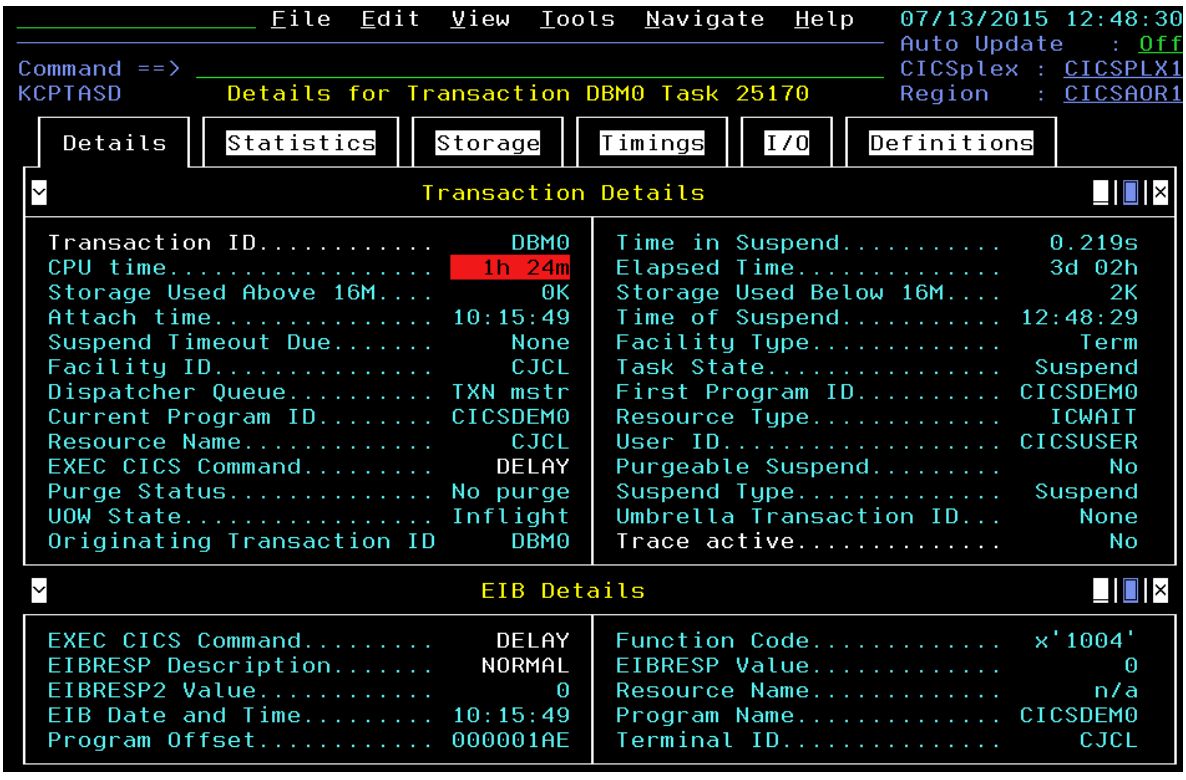
You are now looking the CICS Region overview information for the region. This is monitoring information collected by OMEGAMON CICS.



e) You may drill down for more CICS transaction detail. To see this detail **Position the cursor** next to the transaction (DBM0) and P**ress enter**.



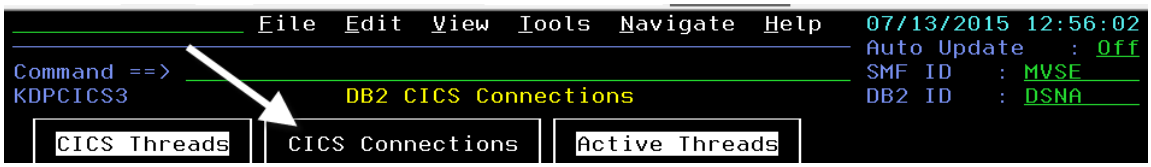
You are now looking at the CICS counters for the transaction.



f) **Press F3 three times** to return to the DB2 CICS Threads display (KDPCICTH).

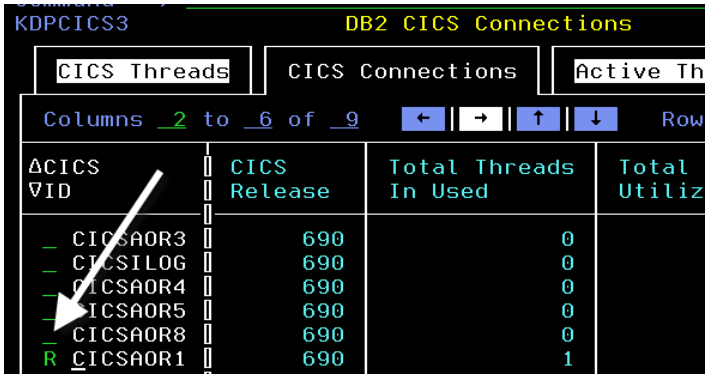
From the DB2 CICS Connections display you may navigate to see more information on the various CICS regions connected to the DB2 subsystem.

g) **Position the cursor** on the CICS Connections tab and **Press Enter**

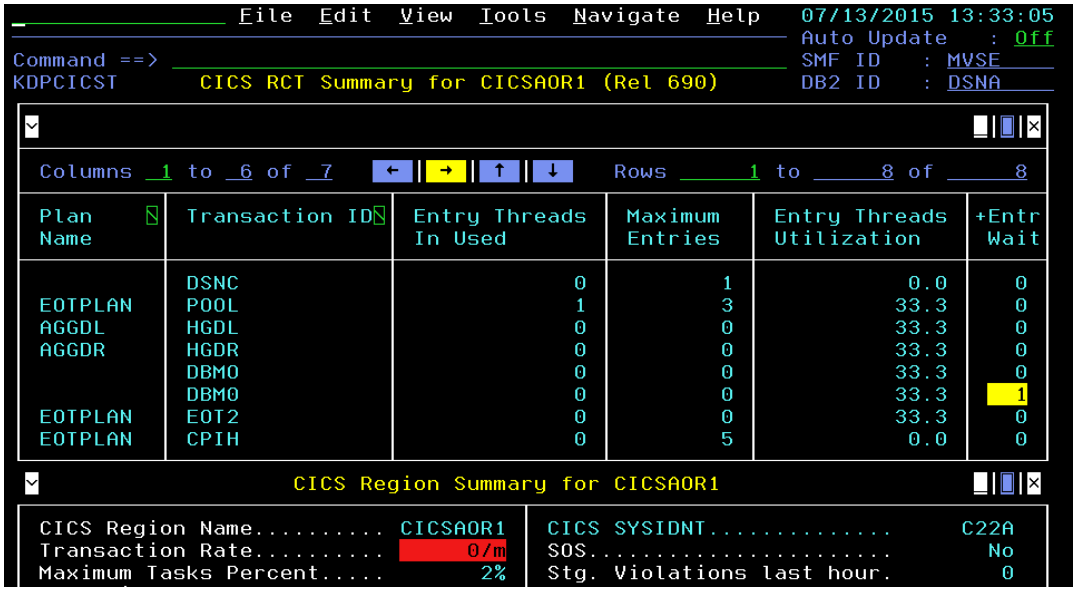


You are now looking at the overview of CICS regions connected to the DB2 subsystem. From this display you may see what CICS regions are connected to DB2, the number of CICS threads utilized.

h) To see information on the CICS RCT **Position the cursor** by the CICS region, **Enter R** and **Press Enter**



You are now looking at the CICS RCT summary for the CICS region, and also CICS region summary information. This display shows more detail on CICS threads used, and bu thread type.



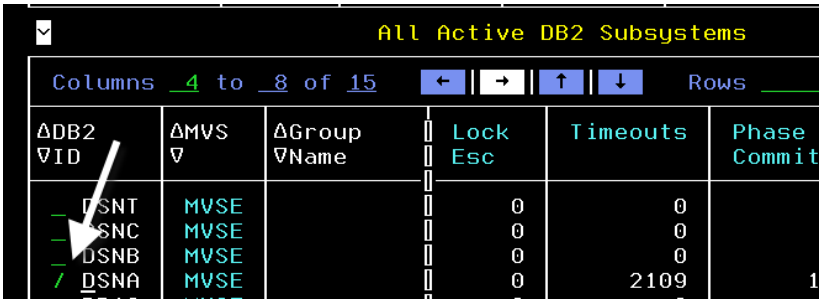
You hane now seen the DB2 and CICS information available in OMEGAMON, and how to drill down and navigate between the monitoring tools within the enhanced 3270 user interface.

i) **Press F3 twice** to return to the KDPSTART panel.

1.6 DB2 Subsystem Analysis

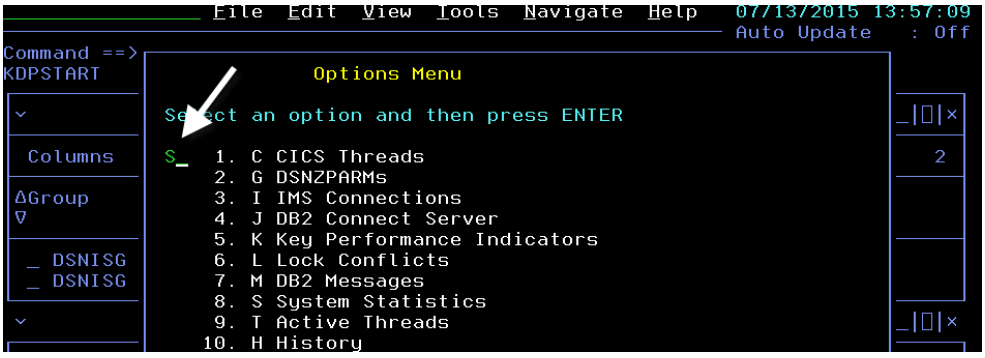
The final phase of this lab exercise will look at resource usage from the subsystem perspective.

a) **Position the cursor** next to the DB2ID (in this example DSNA), enter / and **Press Enter**.

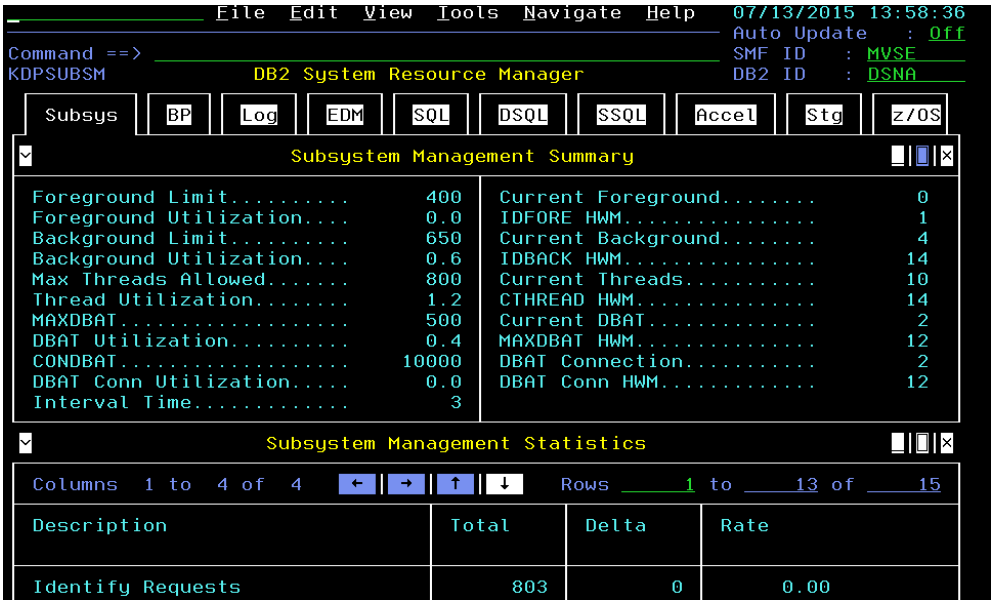


You are presented with a popup with various drill down options.

b) **Enter S** for Subsystem Statistics in the popup option and **Press Enter**

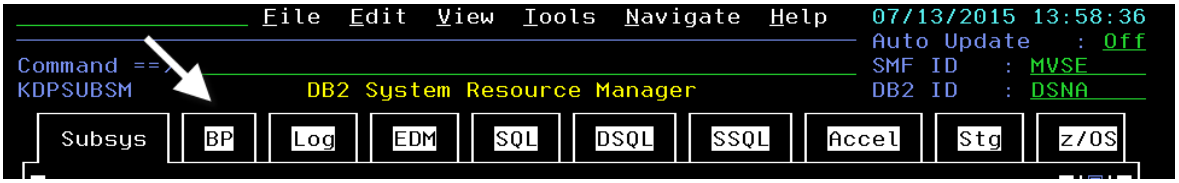


You are now looking at the DB2 subsystem statistics for the current interval.

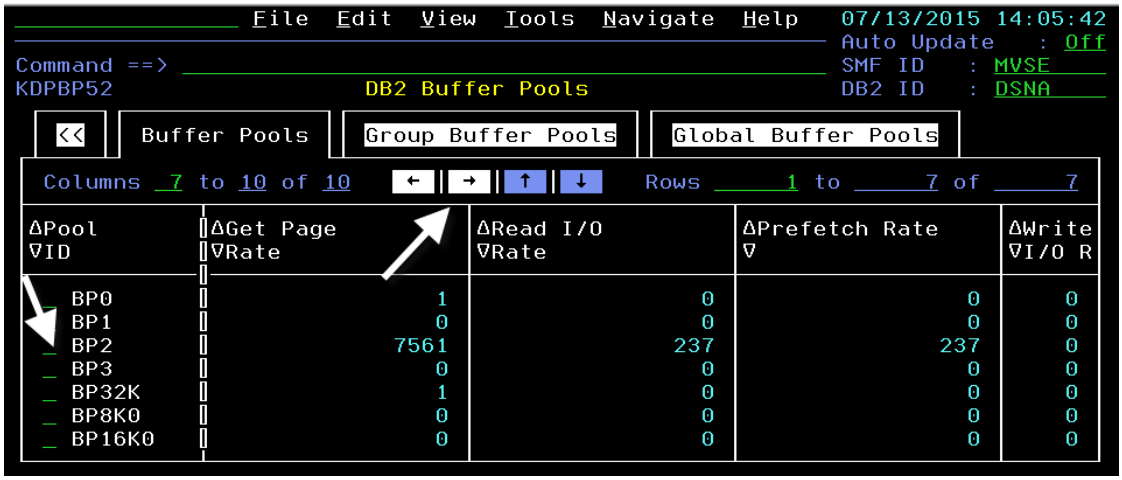


Note that there are push button tabs at the top of the panel that will allow you to easily navigate to other relevant DB2 subsystem statistics level information. You may navigate from here to see information on DB2 buffer pools, logging, EDM pool utilization, SQL activity for the subsystem, and subsystem storage utilization.

c) To see buffer pool information **Position the cursor on the BP tab** and **Press Enter.**

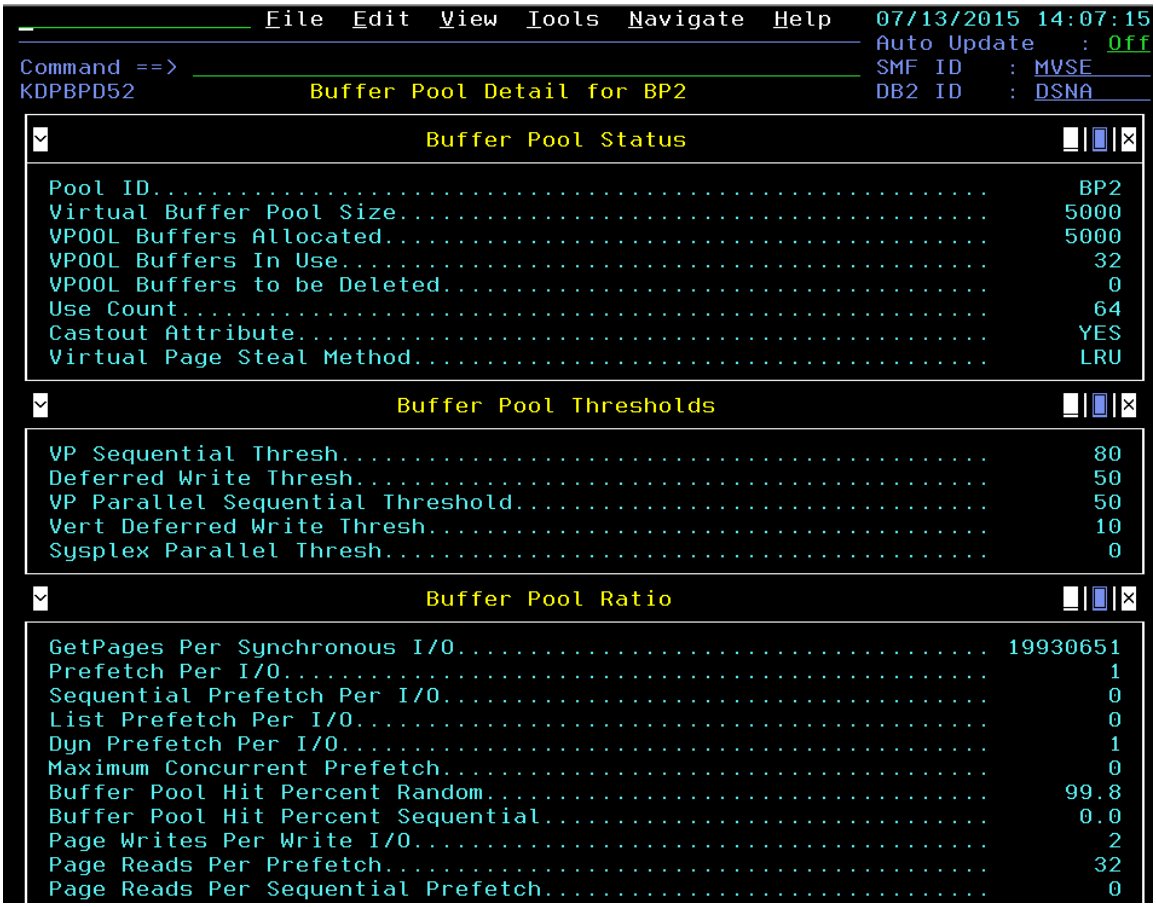


You are now looking at the DB2 Buffer Pool display for a given DB2 subsystem. If you scroll the panel to the right (either **Press F11** or use the **arrow navigation**) you will see a column labeled Getpage Rate (as in the example below).



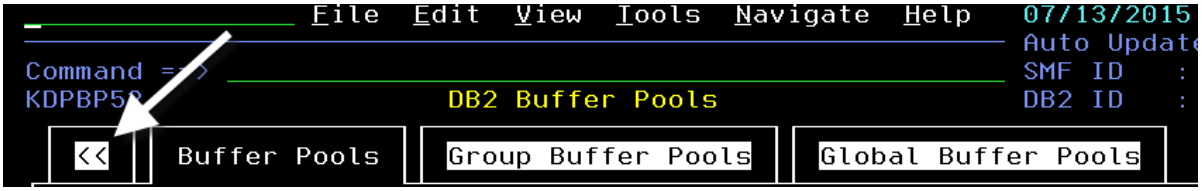
d) You may drill in to see more detail on a specific buffer pool. To do so, **position the cursor** by one of the virtual pools and **Press Enter.**

You are now looking at the getpage and I/O information for the virtual pool, including hit ratios.



e) **Press F3**, to return to the DB2 Buffer Pools display.

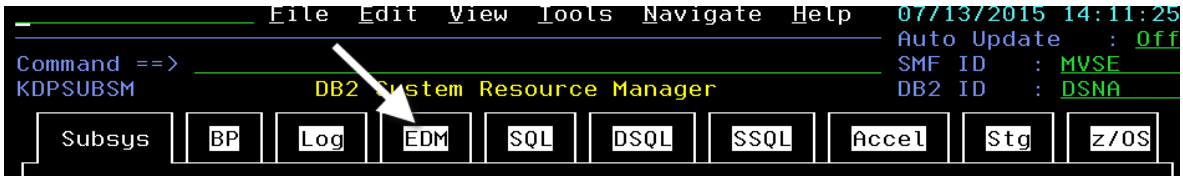
f) P**osition the cursor on the << tab** and **Press Enter**



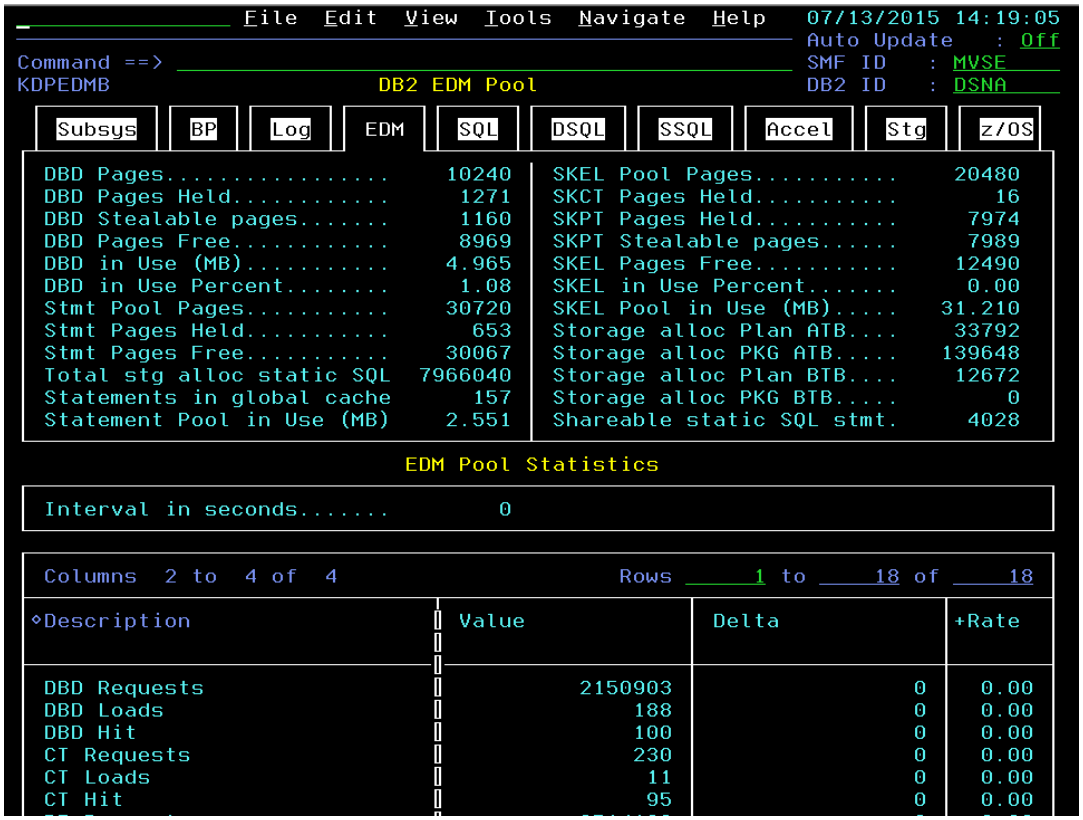
You should now be back at the DB2 System Resource Manager panel (KDPSUBSM).

Another important structure to look at in the DB2 subsystem is the EDM pool.

g) To see EDM pool information **Position the cursor on the EDM tab** and **Press Enter.**

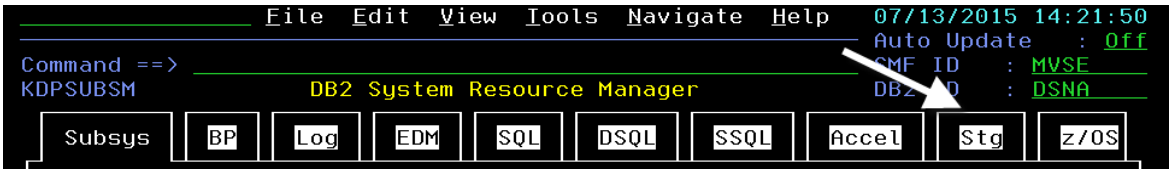


You are now looking at the EDM pool utilization information for the DB2 subsystem. This includes structure size and usage, utilization rates, and storage utilization above and below the bar.

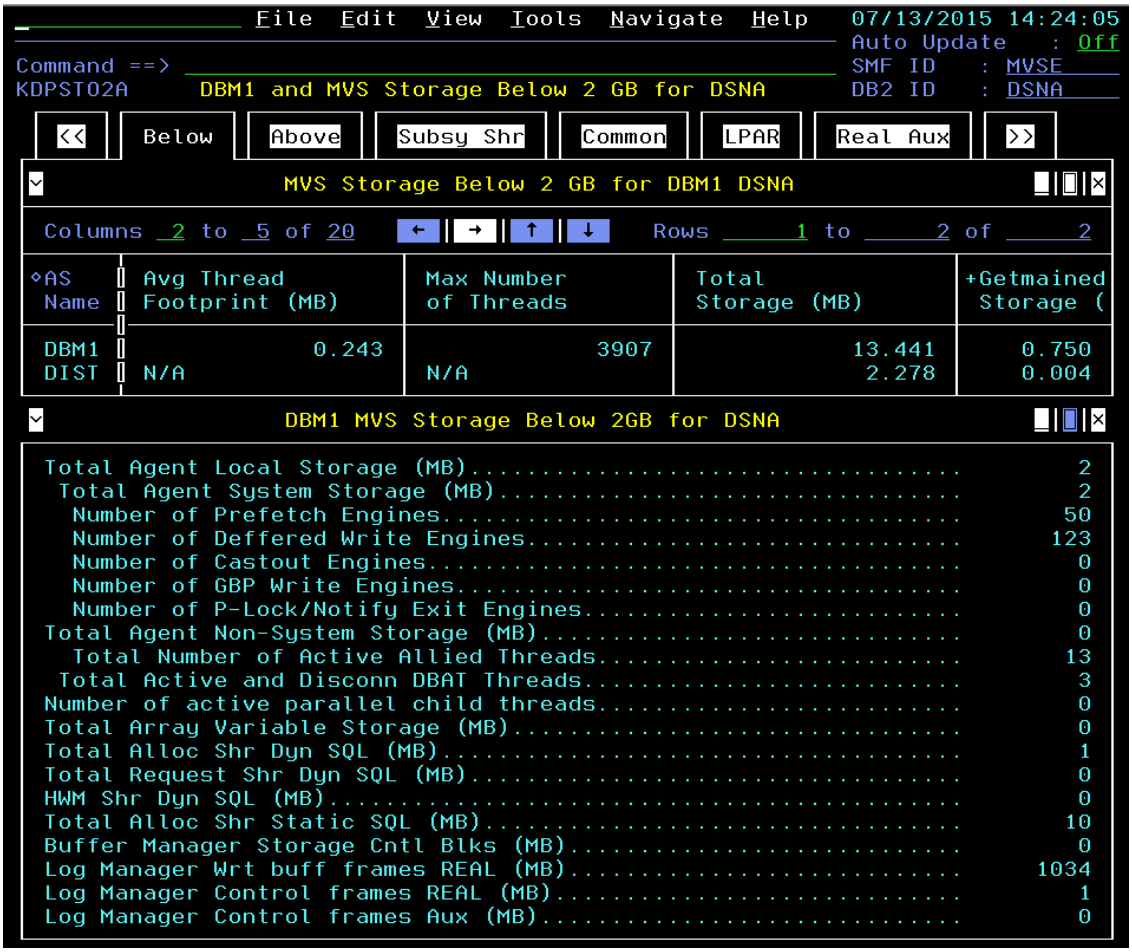


You may also see more detailed information on storage utilization of the DB2 subsystem.

h) To see storage consumption information P**osition the cursor on the Stg tab** and **Press Enter**.



You are now looking at a detailed display of storage consumption within the DB2 subsystem.

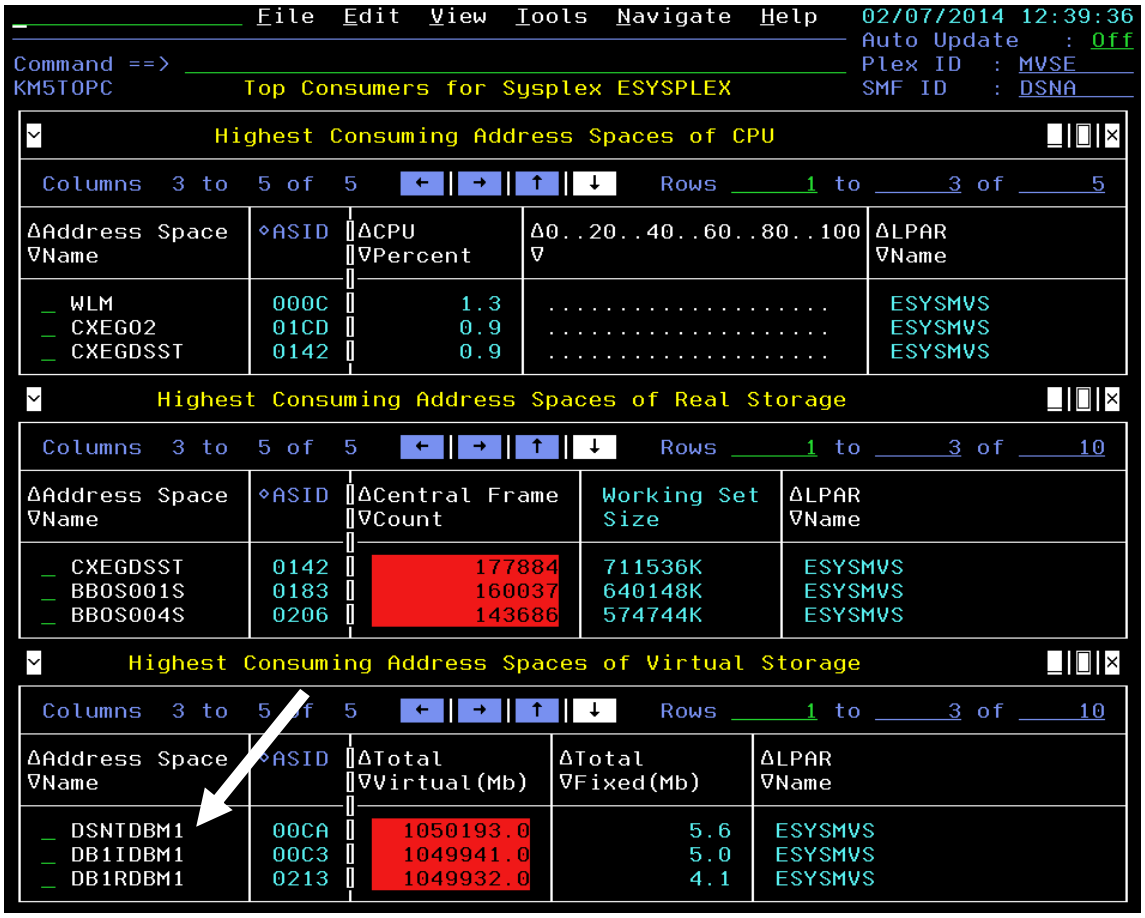


k) Note that there are other tab options on the storage panel. Feel free to tab and enter to display the various panels.

1.7 Navigate to OMEGAMON z/OS Top Consumers Panel

Now that you have looked at the DB2 storage information, there is one more option you may explore to get relevant DB2 subsystem performance information.

a) From the command line **enter =KM5TOPC** and **Press Enter**.



If you have done the OMEGAMON z/OS lab exercise you may already be familiar with this panel. The KM5TOPC panel shows the highest resource utilization tasks in the z/OS environment for CPU, storage, and I/O. Notice that you will probably see some DB2 address spaces show up in this display. From this display you may optionally drill in for additional detail.

Congratulations. You have now completed the OMEGAMON DB2 V5.30 enhanced 3270 user interface Test Drive.

Please feel free to ask your lab instructor for additional exercises.

Lab #2 Using Classic Interface To Monitor Critical DB2 Subsystem Resources

The first lab focused on the OMEGAMON DB2 enhanced 3270 user interface. The second set of lab exercises will focus on the capabilities of OMEGAMON DB2 Classic Interface.

This lab is performed from the default 'start' panel, ZMENU. The first classic interface scenario illustrates several of the detailed displays available in OMEGAMON XE for DB2.

2.1 View DB2 subsystem performance data

a) Sign on to the Classic 3270 user interface.

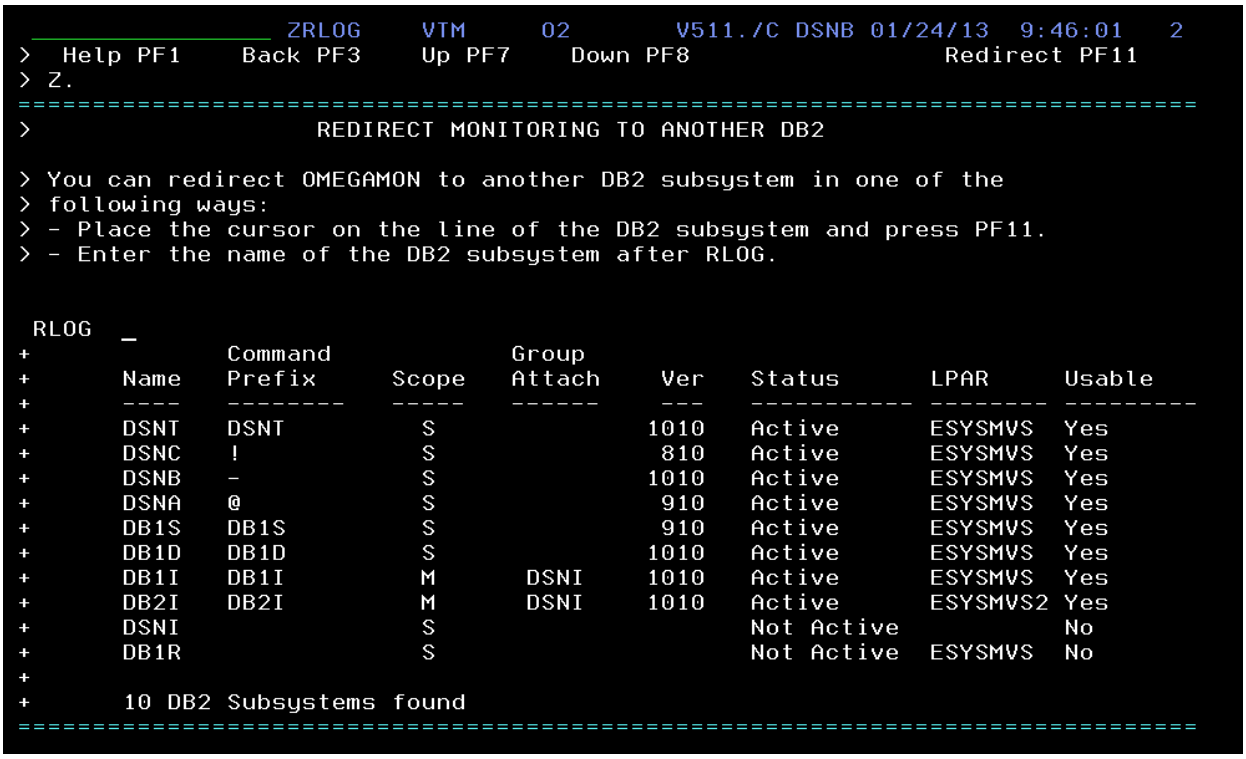
Press ENTER.

You are now looking at the default initial screen for OMEGAMON DB2 Classic Interface, ZMENU. This panel is the main panel for the OMEGAMON DB2 Classic interface and provides menu options for all the other various displays within the tool.

To navigate from the main panel you may enter the appropriate letter commands to the top left portion of the panel (to the left of the panel name), or you may select the letter by entering an S next to the panel option and pressing Enter.

Note that in the above example you are looking at DB2 subsystem DSNB. For purposes of the lab exercises we will be looking at DB2 workload running on DB2 subsystem DSNA.

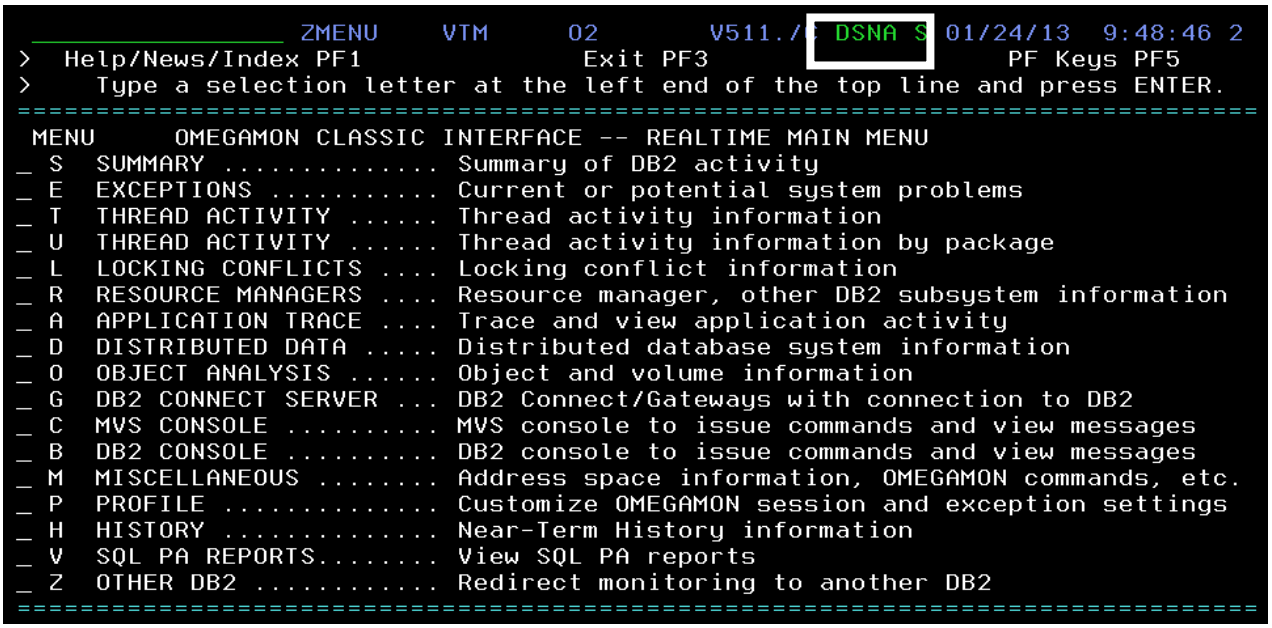
To switch to DSNA, p**osition the cursor** on the command line (upper left corner of the panel). Enter **Z** and **Press Enter.**



You are now looking at all the DB2 subsystems being monitored by OMEGAMON DB2. Understand that one OMEGAMON collector task may monitor multiple DB2 subsystems.

To switch to DSNA, **position the cursor** on DSNA and **Press F11.**

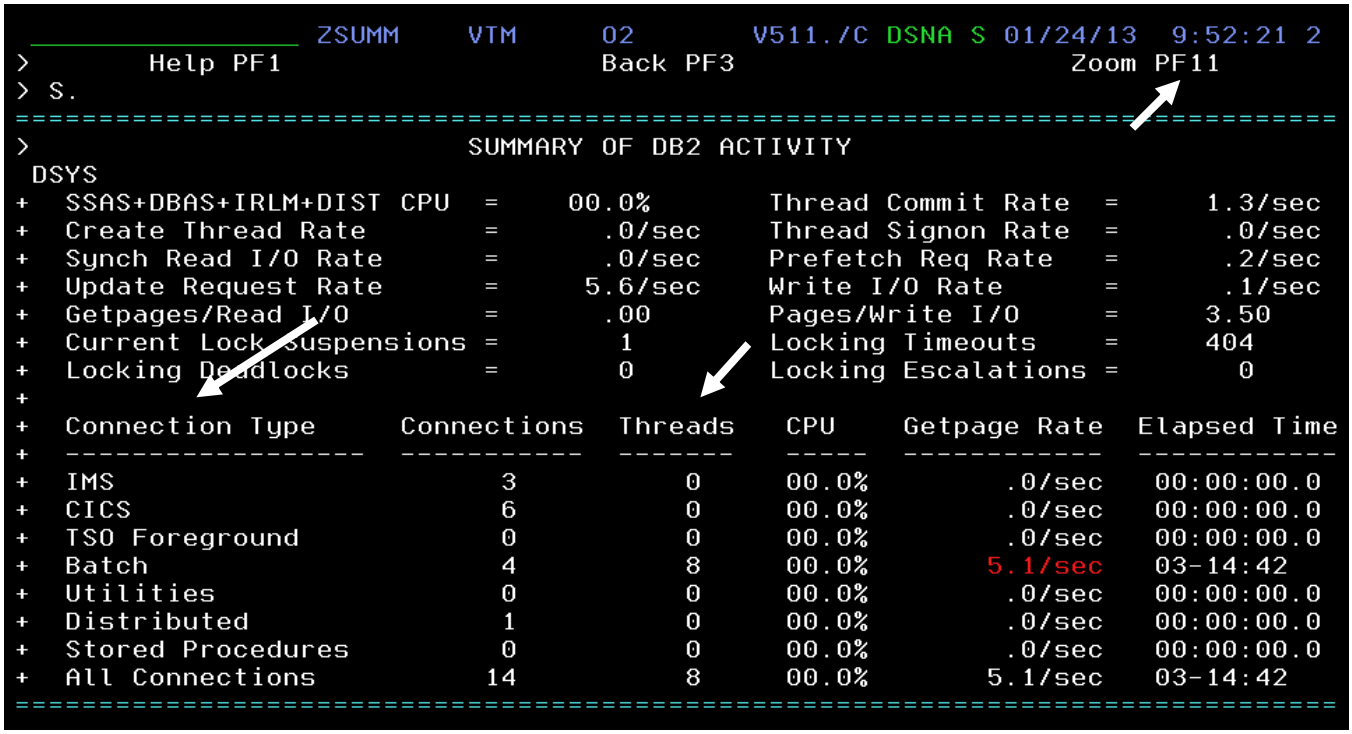
Next **Press Enter.**



You should now be looking at the ZMENU panel for DB2 subsystem DSNA (see above).

b) View the DB2 activity summary

Position the cursor on the command line (upper left corner of the panel). Enter **S** and **Press Enter.**

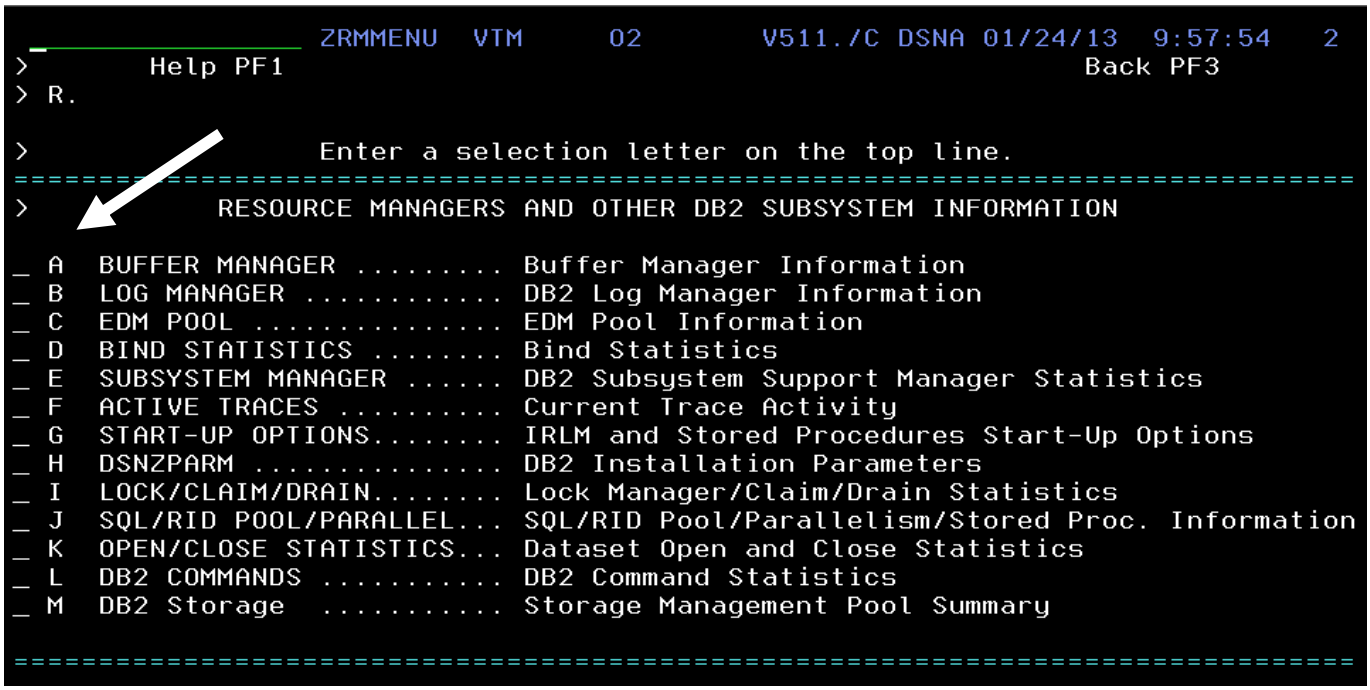


You are now looking at the DB2 summary activity display. From here you can see relevant information about how various threads are connecting to the DB2 subsystem. From this display you can use the F11 zoom option to drill into various threads.

Press F3 to return to ZMENU.

c) View DB2 resources

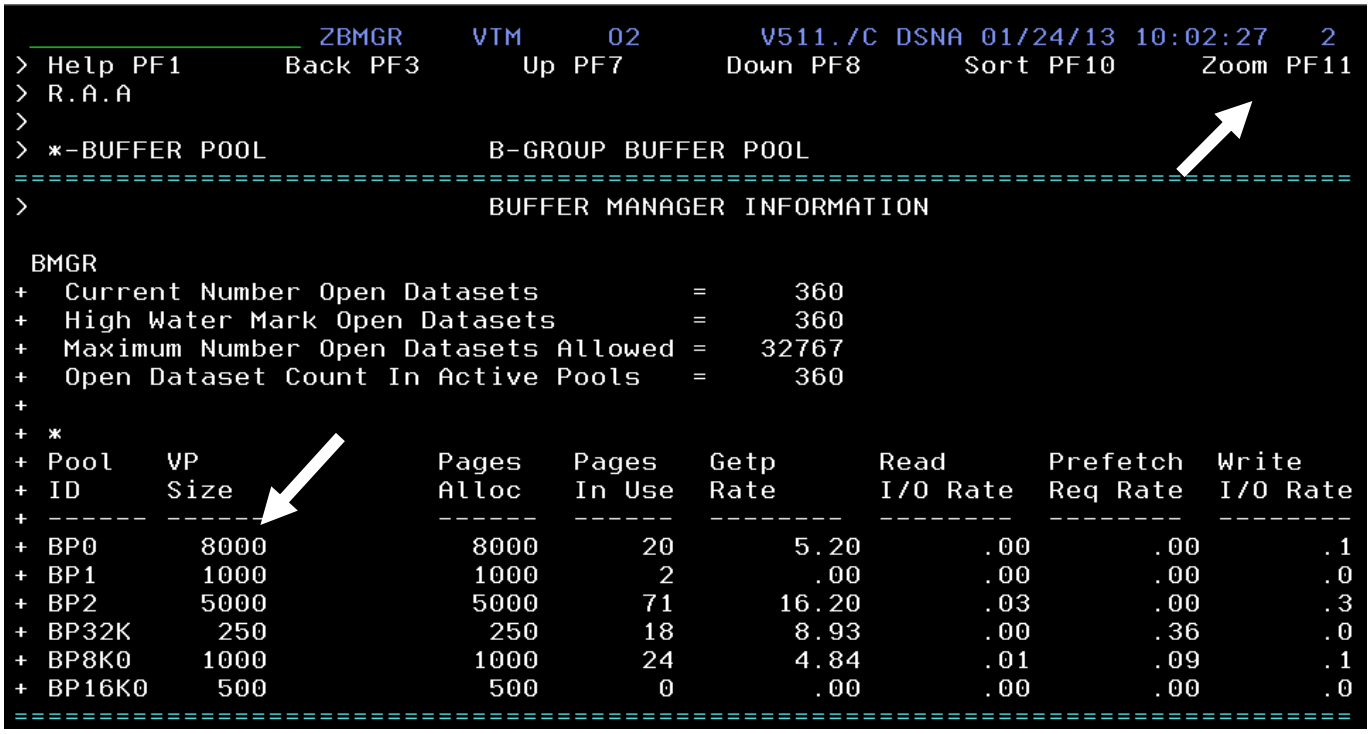
Position the cursor on the command line (upper left corner of the panel). Enter **R** and **Press Enter.**



You are now looking at the Resource manager selection panel (ZRMMENU). From here you can navigate to look at all the key DB2 subsystem resources, such as buffer pools, logging, EDM pool, and DB2 locking. The panel contains letter options for each of these resources.

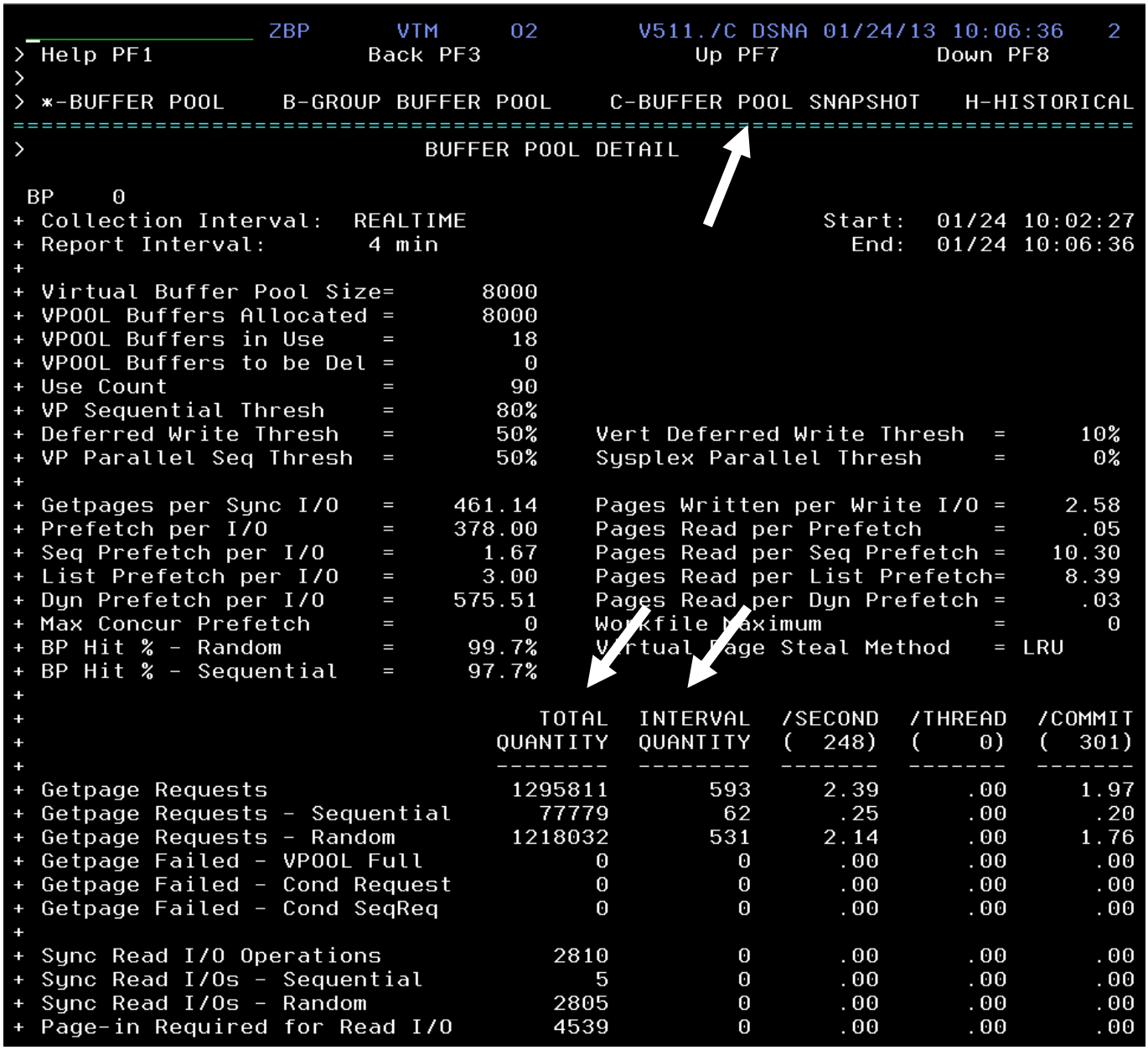
d) View DB2 buffer pool activity

To see buffer manager information, **position the cursor** on the command line (upper left corner of the panel). Enter **A** and **Press Enter** .



You are now looking at the overview of virtual pools being used in the DB2 subsystem. From here you can see the number of pages currently in use, getpage activity for each buffer pool, and various I/O rates for each buffer pool.

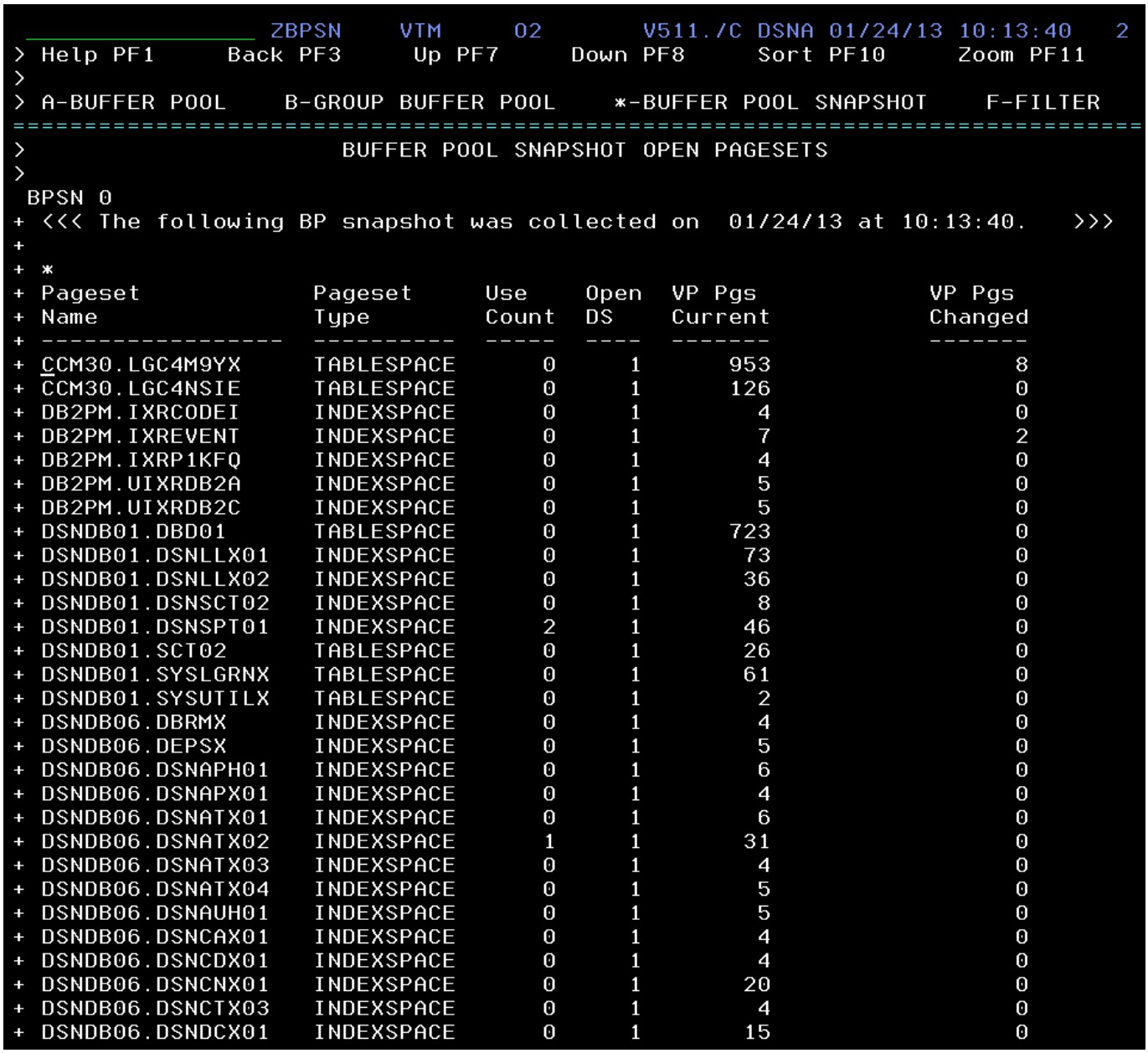
To see more information on a specific pool **position the cursor** on a BP (such as BP0) and **Press F11** to zoom.



You are now looking at the buffer pool detail display for a specific buffer pool. From here you see many critical buffer pool measures, such as virtual pool size, getpage rates, and pool hit ratios. Note that you also see counts broken out in multiple columns. For example, you can see getpage requests since DB2 started in the “Total Quantity” column, and then the number of getpages for the current interval (meaning between screen refreshes).

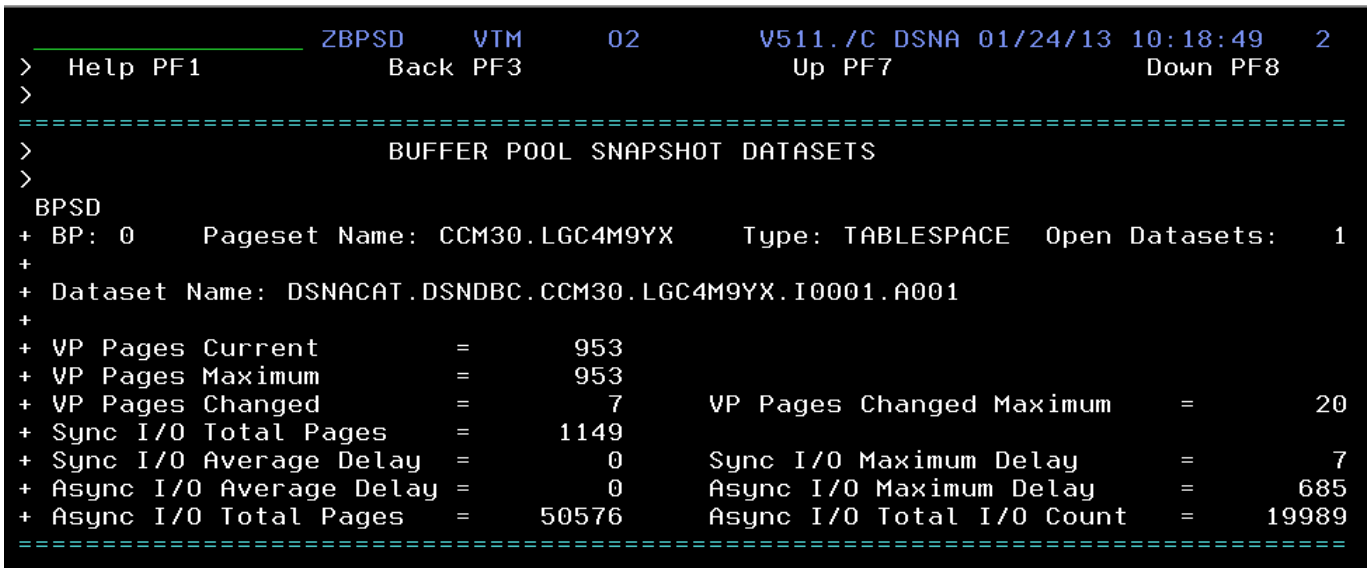
From this detail display you can drill in for even more detail. Let’s look at the Buffer Pool snapshot.

Position the cursor on the command line (upper left corner of the panel). Enter **C** and **Press Enter** .



You are now looking at the snapshot of objects active within the particular DB2 buffer pool. This display will show such information as the number of pages currently used within the buffer pool for the given object.

To see detailed information about a specific object within the buffer pool, **position the cursor** on an object name and **Press F11** to zoom.



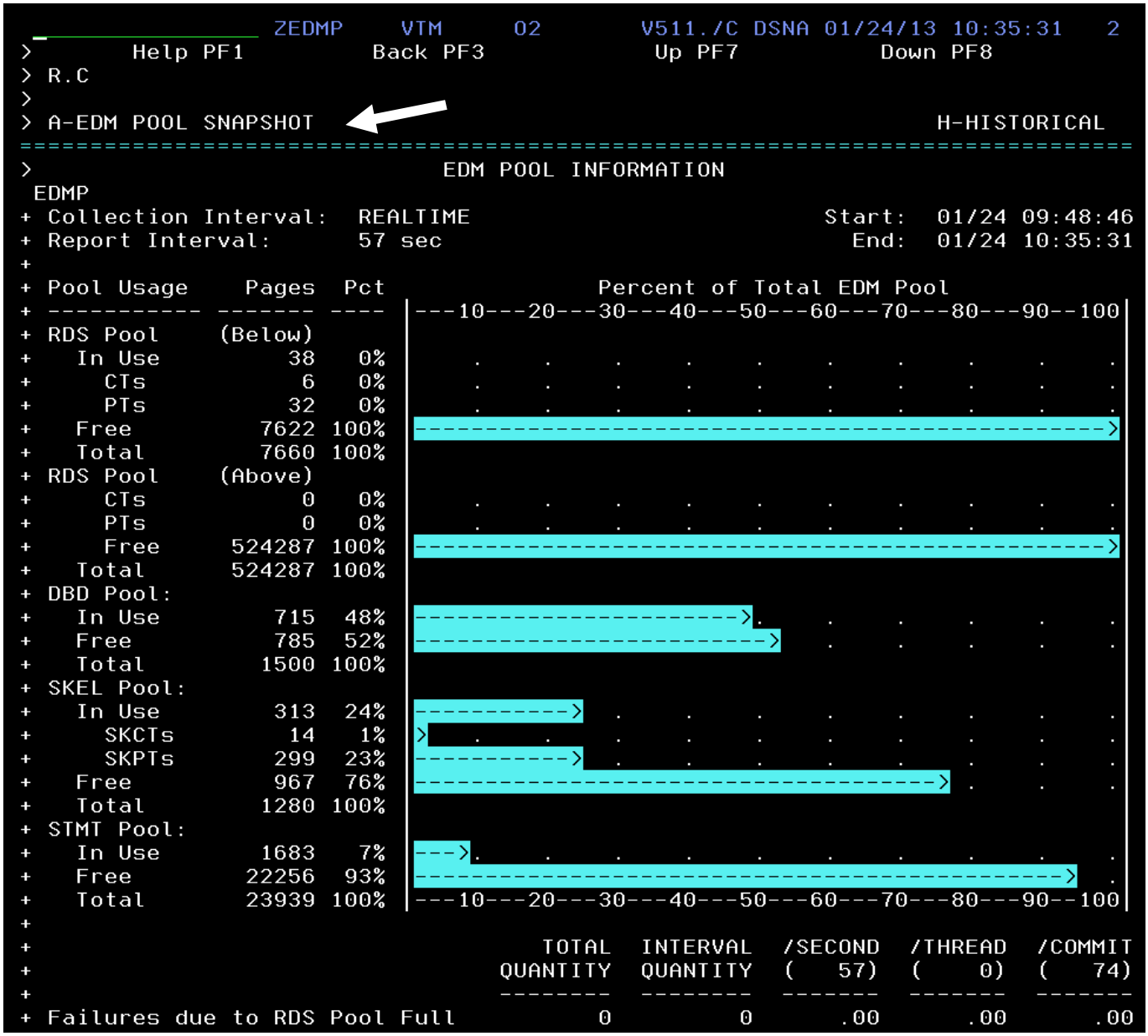
You are now looking at the details for specific object in the virtual pool, including the number of pages in the pool, and how much I/O has been performed for this object as well s the type of I/O (sync versus async I/O).

Press F3 then **Press F3** again and **Press F3** one more time to return to the resource manager menu, ZRMMENU.

e) View EDM pool activity

From the resource manager menu, ZRMMENU, you can also drill down to see EDM pool activity.

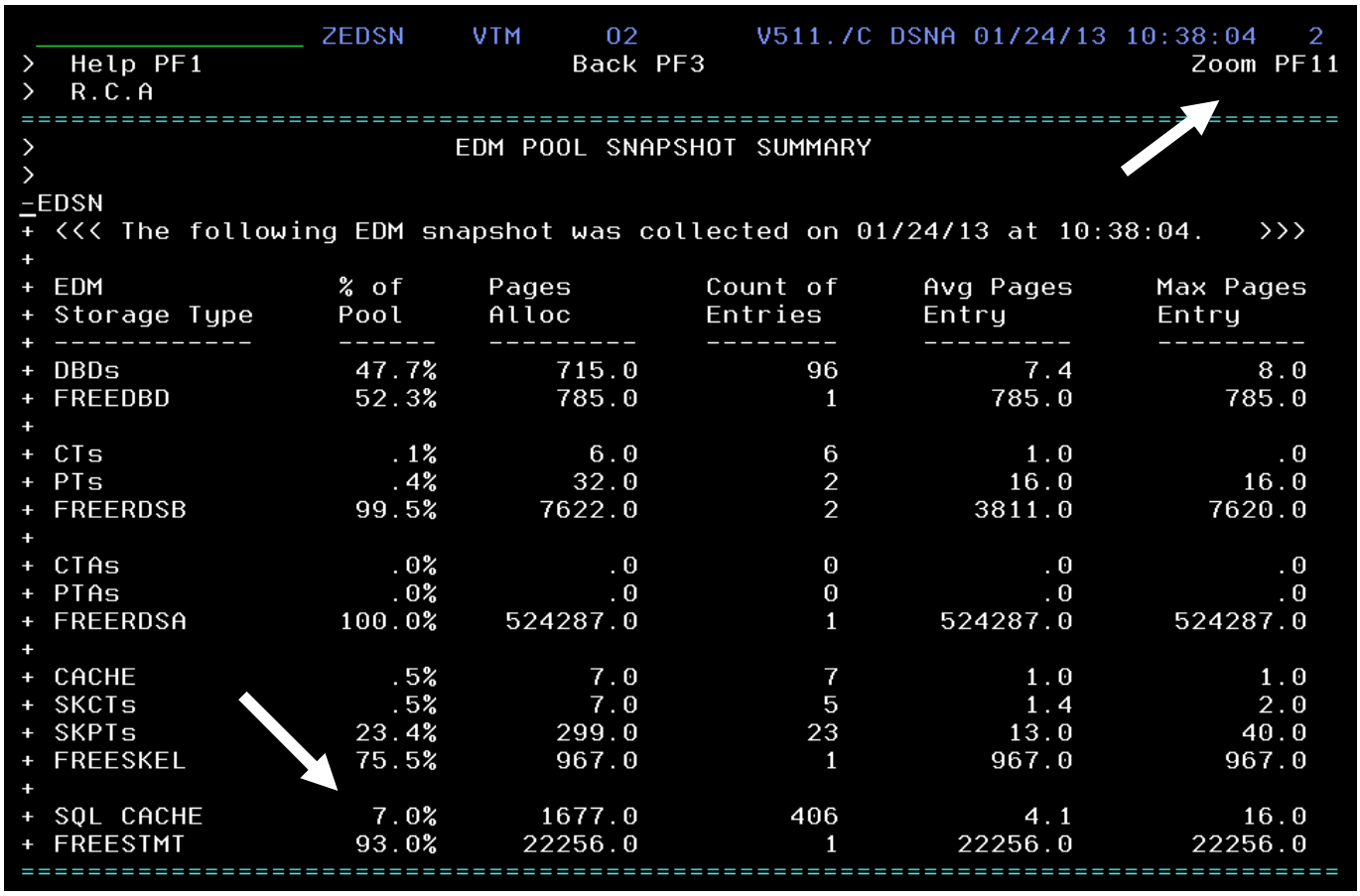
Position the cursor on the command line (upper left corner of the panel). Enter **C** and **Press Enter** .



You are now looking at the EDM pool information panel. From here you can see the breakdown of EDM pool activity by component of the EDM pool, including usage and amount of free space.

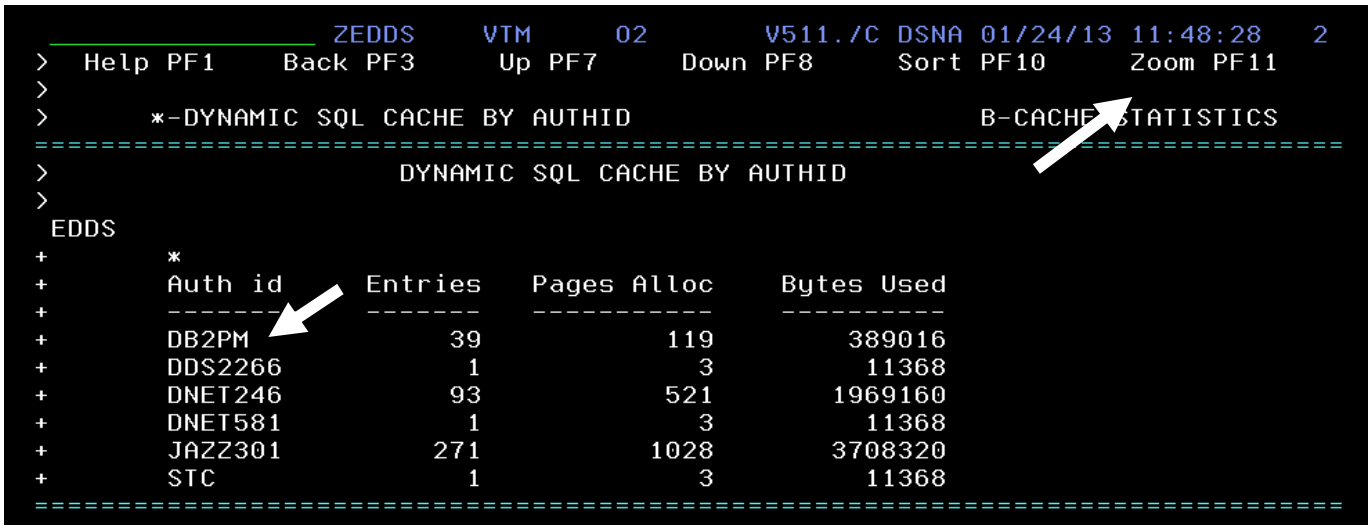
To see more detail on the EDM pool you can use the EDM pool snapshot facility. The snapshot facility will provide detail as to the usage of the various EDM pool structures.

Position the cursor on the command line (upper left corner of the panel). Enter **A** and **Press Enter** .



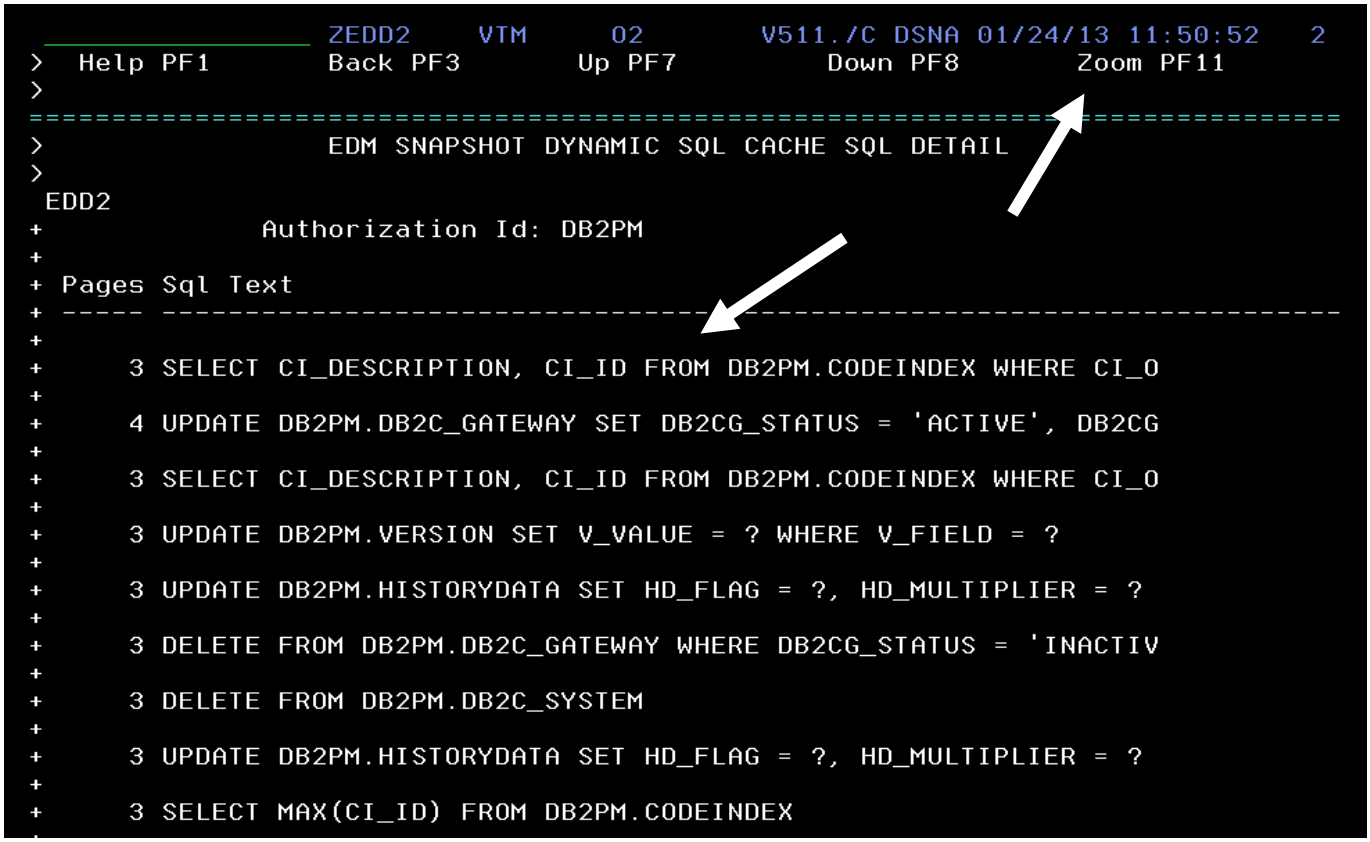
You are now looking at the EDM pool snapshot display. From here you can see pages allocated for each component of the EDM pool. Plus, you can drill in to see details of usage within each component.

To see details, **position the cursor** on an EDM component (such as SQL cache) and **Press F11** to zoom.



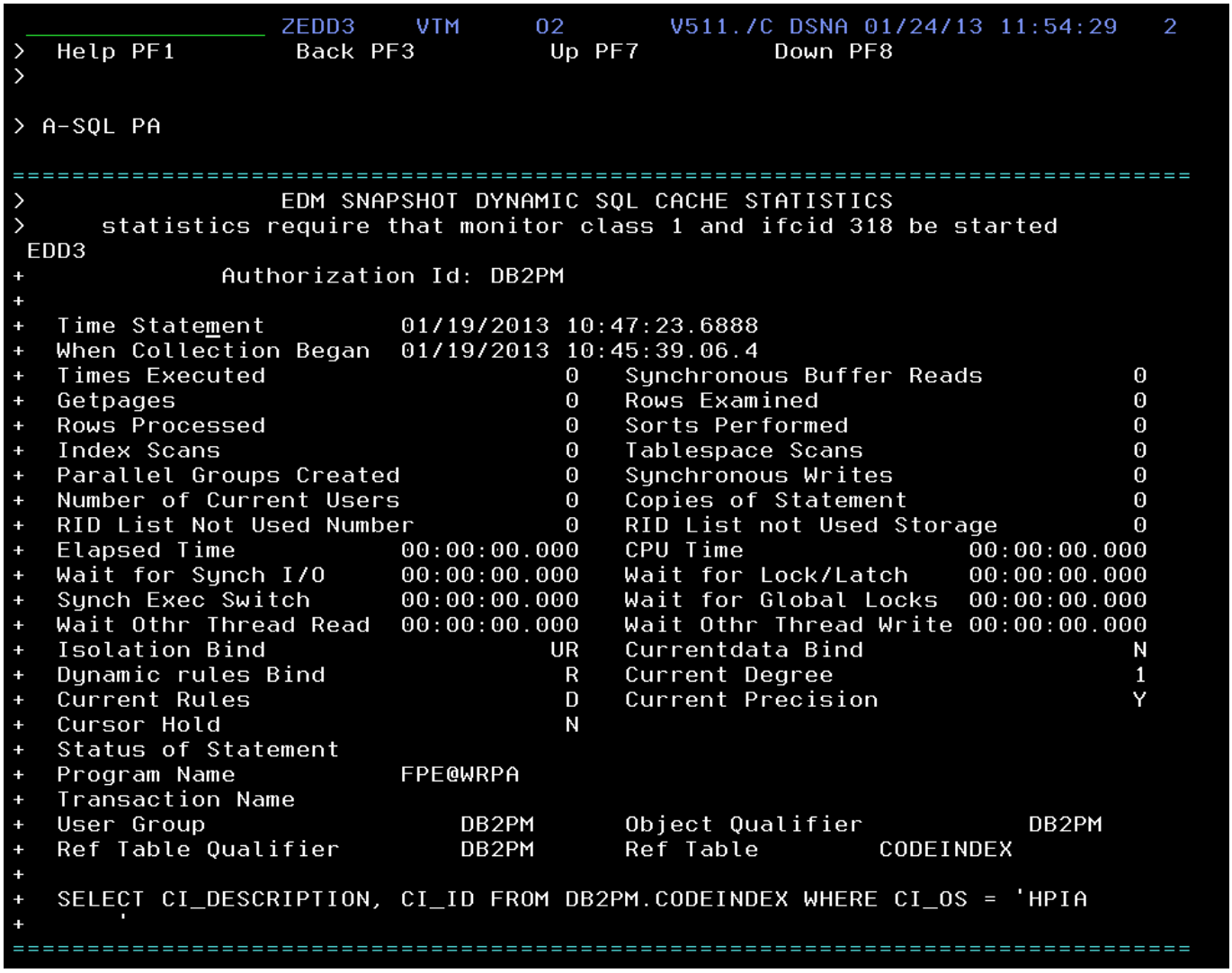
You are now looking at a snapshot of the Dynamic SQL cache, with usage sorted by DB2 authid. Dynamic SQL cache is potentially important for DB2 dynamic SQL performance, and it is often useful to know what SQL calls reside within the dynamic SQL cache.

To see details on the usage for a given authid, **position the cursor** on an authid and **Press F11** to zoom.



You are now looking at the SQL statement contents of the dynamic SQL cache. This shows the text of the SQL calls within the cache. Each line represents an SQL call that is stored within the cache.

To see performance and usage details on a specific call, **position the cursor** on a SQL call text and **Press F11** to zoom.



You are now looking at the performance details within the SQL cache for the specific SQL call.

Now please return to the ZRMMENU menu panel. **Press F3** multiple times until you get back to that panel.

f) View DB2 lock activity

From the resource manager menu, ZRMMENU, you can also drill down to see DB2 locking activity.

Position the cursor on the command line (upper left corner of the panel). Enter **I** and **Press Enter** .

```
_________________ ZLOKM    VTM     02       V511./C DSNA 01/24/13 12:00:41   2
>       Help PF1         Back PF3            Up PF7           Down PF8
> R.I.A
>
> *-LOCK STATISTICS           B-GLOBAL LOCK STATISTICS           H-HISTORICAL
================================================================================
>                          LOCK MANAGER INFORMATION

 LOKM
+ Collection Interval:  REALTIME                          Start:  01/24 10:34:31
+ Report Interval:      42 sec                             End:  01/24 12:00:41
+
+                                  TOTAL  INTERVAL  /SECOND  /THREAD  /COMMIT
+                                QUANTITY  QUANTITY  (   42)  (    2)  (  312)
+                                --------  --------  -------  -------  -------
+ Deadlocks Detected                    0         0      .00      .00      .00
+ Timeouts Detected                   452        25      .60    12.50      .08
+ Susp Detected - Lock Only          4126        74     1.76    37.00      .24
+ Susp Detected - Latch Only         3841        34      .81    17.00      .11
+ Susp Detected - Other             12589        63     1.50    31.50      .20
+
+ Lock Escalations - to Shared          0         0      .00      .00      .00
+ Lock Escalations - to Exclusive       0         0      .00      .00      .00
+
+ Lock Requests                   4124540     56257  1339.45  28128.5   180.31
+ Unlock Requests                 1667654     23124   550.57  11562.0    74.12
+ Query Requests                   12588        63     1.50    31.50      .20
+ Change Requests                 338388      4587   109.21  2293.50    14.70
+ Other IRLM Requests                  3         0      .00      .00      .00
+
+ CLAIM/DRAIN INFORMATION
+ ------------------------------
+ Claim Requests                  3587129     51198  1219.00  25599.0   164.10
+ Unsuccessful Claim Requests           0         0      .00      .00      .00
+ Drain Requests                     1595        27      .64    13.50      .09
+ Unsuccessful Drain Requests           0         0      .00      .00      .00
================================================================================
```

You are now looking at the DB2 subsystem level locking statistics display (ZLOKM) for a given time interval. This panel will show information on DB2 deadlocks and timeouts.

There are many other resource displays that you may review as part of the DB2 resource displays. We have looked at some of the representative examples.

Please remain on panel ZLOKM to continue with the next scenario.

2.2 Using DB2 historical data to isolate subsystem issues

This scenario continues to explore the Classic 3270 interface details. This lab will explore how the historical analysis facilities of OMEGAMON for DB2 Classic interface may be used to identify and isolate potential locking and performance issues.

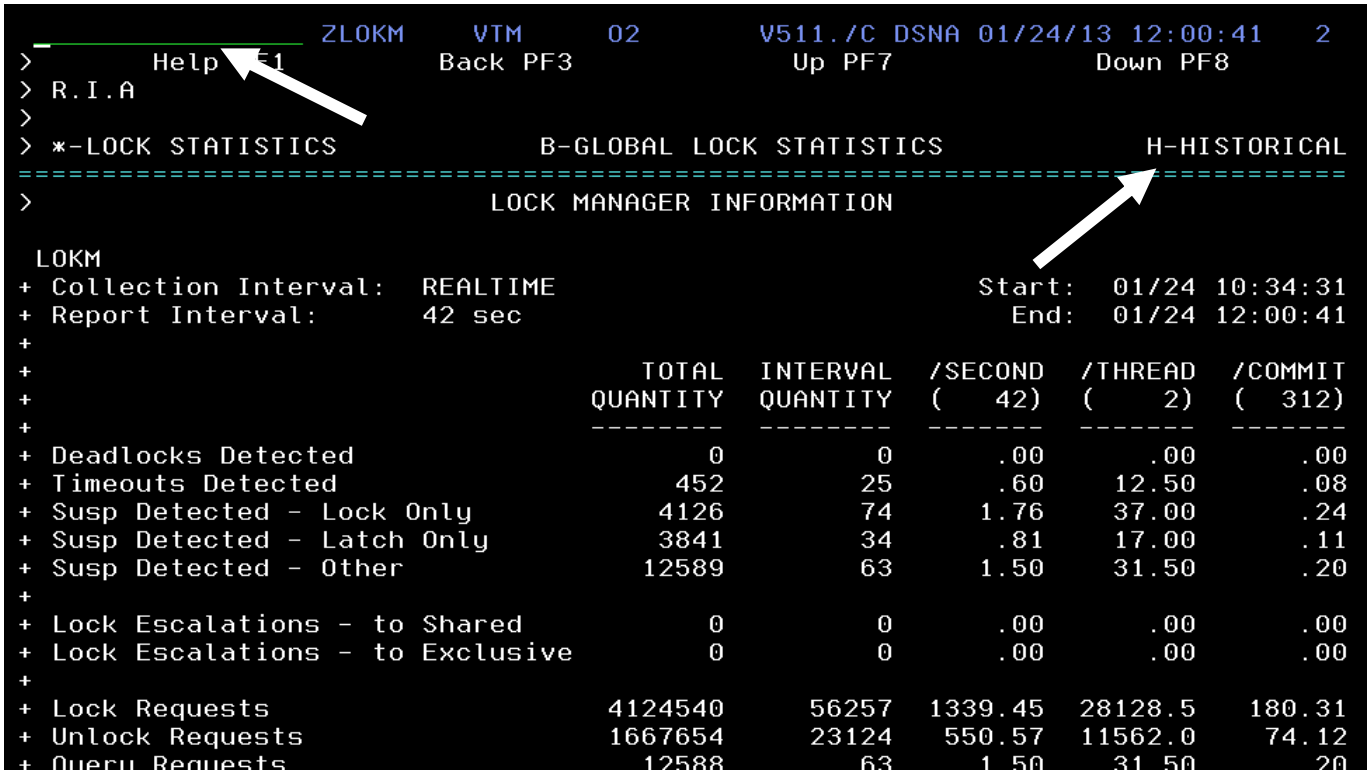
View DB2 Near Term History (NTH)

In this Scenario, you will view Near Term History that captures subsystem and thread activity in the past few hours.

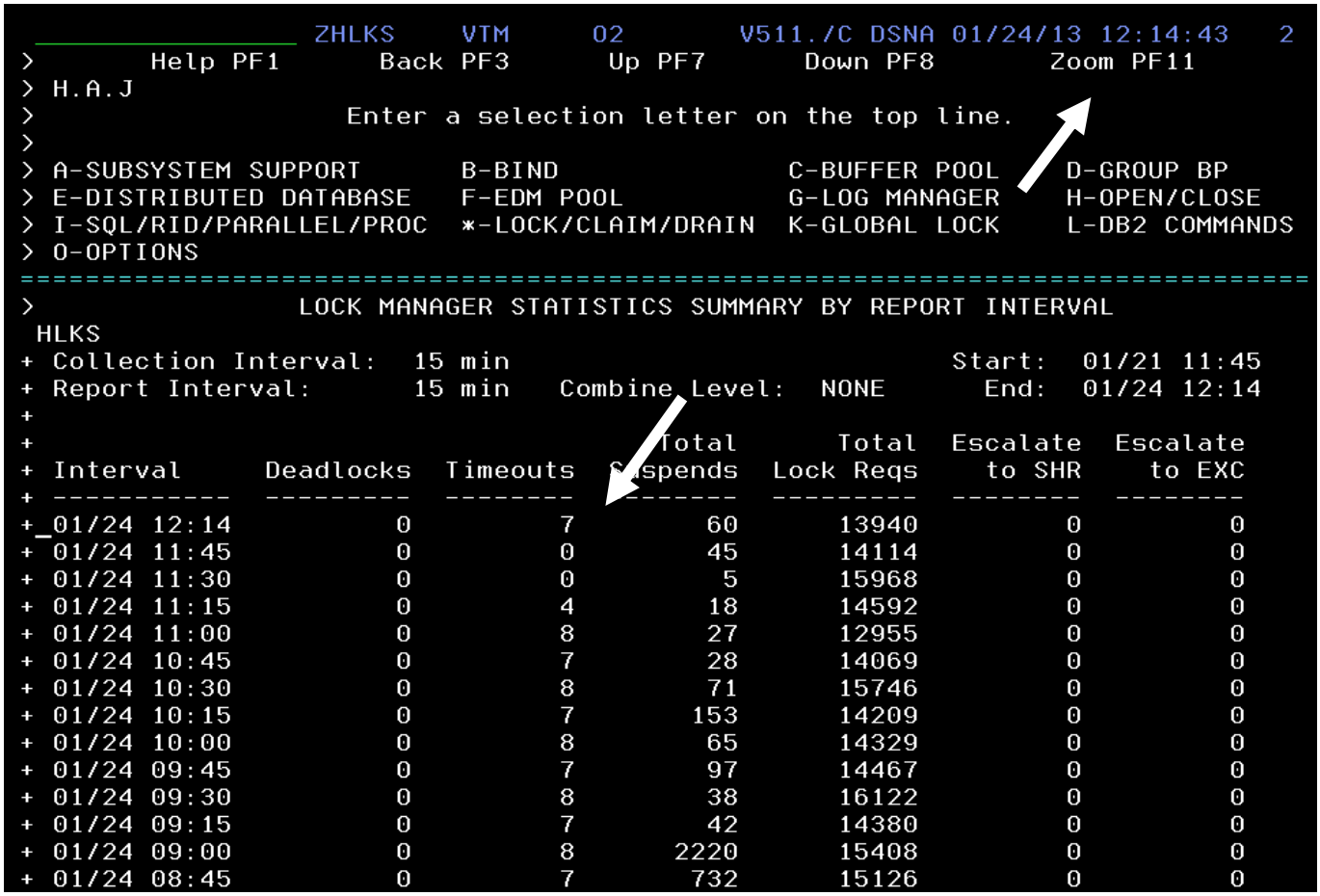
Near Term History is an important component of OMEGAMON DB2. Near Term History captures DB2 subsystem statistics data and DB2 thread activity data, and stores the information in a set of VSAM files allocated to OMEGAMON DB2 for easy analysis and retrieval. History is an important mechanism for analyzing performance issues after the fact.

This Scenario begins on the ZLOKM panel where we finished the prior scenario.

a) From the ZLOKM panel drill down to see historical data.



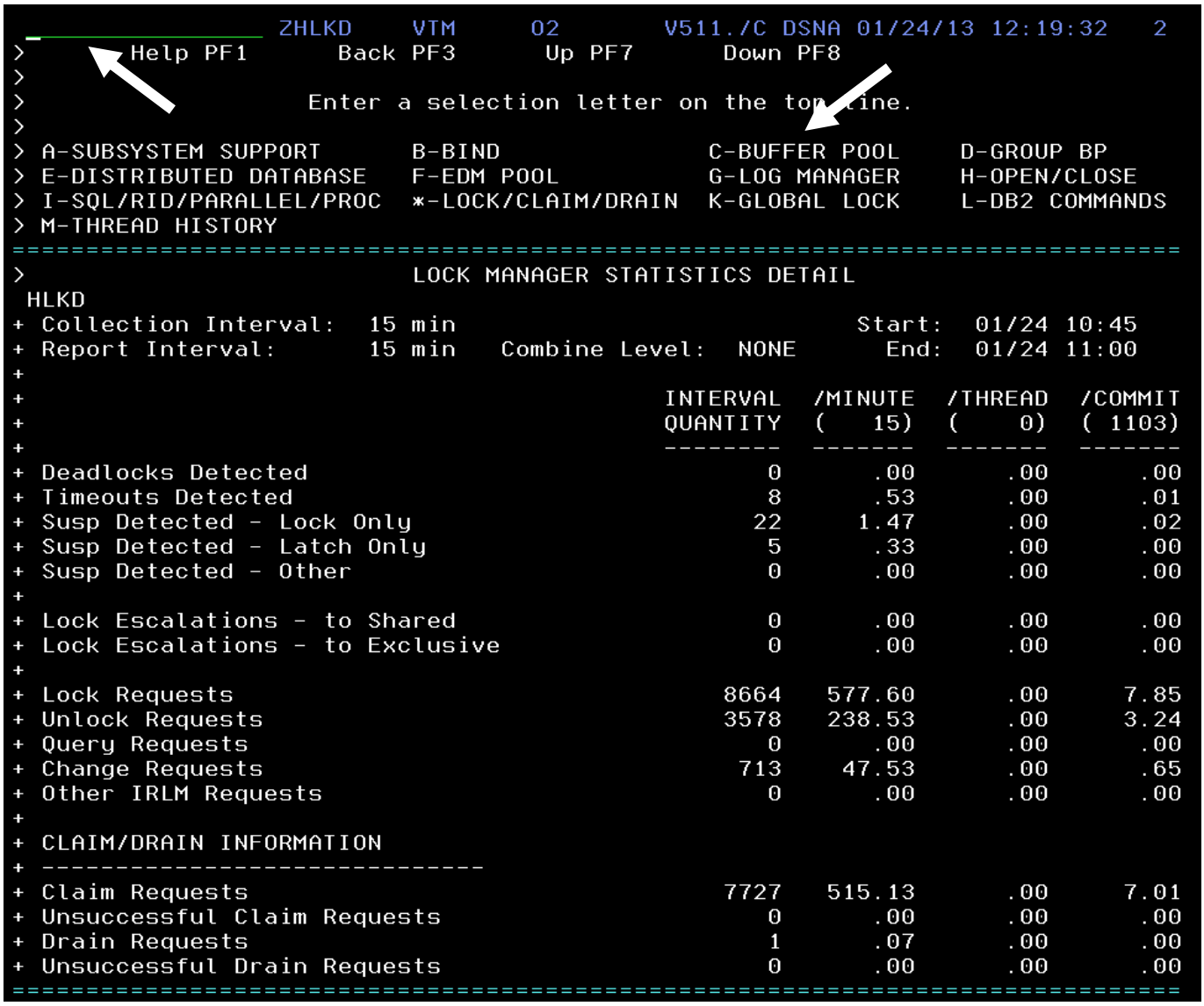
Position the cursor on the command line (upper left corner of the panel). Enter **H** and **Press Enter** .



You are now looking at the Near Term History for DB2 locking over several hours. Each line represents 15 minutes of locking statistics information.

Focus on information such as the timeout and deadlock columns. Here you can see a time interval where timeouts and/or deadlocks occurred.

To see details on locking for a specific time interval, **position the cursor** on a line (preferably where timeouts occurred) and **Press F11** to zoom to history detail.



You are now looking at the DB2 subsystem locking statistics for a specific time interval. From here you can see how many deadlocks and timeouts occurred. You can also see how many lock suspensions occurred and if any lock escalation occurred.

From this display you may also view other relevant subsystem statistics for this same time interval. For example you can easily navigate from locking statistics to buffer pool statistics.

b) View Buffer pool history

Too see buffer pool information, p**osition the cursor** on the command line (upper left corner of the panel). Enter **C** and **Press Enter** .

```
                 ZHBPD    VTM     02       V511./C DSNA 01/24/13 12:24:58   2
>       Help PF1      Back PF3      Up PF7      Down PF8
>
>                   Enter a selection letter on the top line.
>
> A-SUBSYSTEM SUPPORT      B-BIND              *-BUFFER POOL    D-GROUP BP
> E-DISTRIBUTED DATABASE   F-EDM POOL          G-LOG MANAGER    H-OPEN/CLOSE
> I-SQL/RID/PARALLEL/PROC  J-LOCK/CLAIM/DRAIN  K-GLOBAL LOCK    L-DB2 COMMANDS
> M-THREAD HISTORY
================================================================================
>                         BUFFER POOL STATISTICS DETAIL
 HBPD  0
+ Collection Interval:  15 min                              Start:  01/24 10:45
+ Report Interval:      15 min   Combine Level:  NONE        End:  01/24 11:00
+
+ Virtual Buffer Pool Size=      8000
+ VPOOL Buffers Allocated =      8000
+ VPOOL Buffers in Use    =        20
+
+ VPOOL Buffers to be Del =         0
+ Use Count               =         0
+ VP Sequential Thresh    =       80%
+ Deferred Write Thresh   =       50%    Vert Deferred Write Thresh  =      10%
+ VP Parallel Seq Thresh  =       50%    Sysplex Parallel Thresh     =       0%
+
+ Getpages per Sync I/O   =       .00    Pages Written per Write I/O =    2.28
+ Prefetch per I/O        =       .00    Pages Read per Prefetch     =     .00
+ Seq Prefetch per I/O    =       .00    Pages Read per Seq Prefetch =     .00
+ List Prefetch per I/O   =       .00    Pages Read per List Prefetch=     .00
+ Dyn Prefetch per I/O    =       .00    Pages Read per Dyn Prefetch =     .00
+ Max Concur Prefetch     =         0    Workfile Maximum            =       0
+ BP Hit % - Random       =    100.0%    Virtual Page Steal Method   = LRU
+ BP Hit % - Sequential   =    100.0%
+
+                                        INTERVAL  /MINUTE  /THREAD  /COMMIT
+                                        QUANTITY  (   15)  (    0)  ( 1103)
+                                        --------  -------  -------  -------
+ Getpage Requests                           2035   135.66      .00     1.84
+ Getpage Requests - Sequential               139     9.26      .00      .12
+ Getpage Requests - Random                  1896   126.40      .00     1.71
+ Getpage Failed - VPOOL Full                   0      .00      .00      .00
+ Getpage Failed - Cond Request                 0      .00      .00      .00
+ Getpage Failed - Cond SeqReq                  0      .00      .00      .00
```

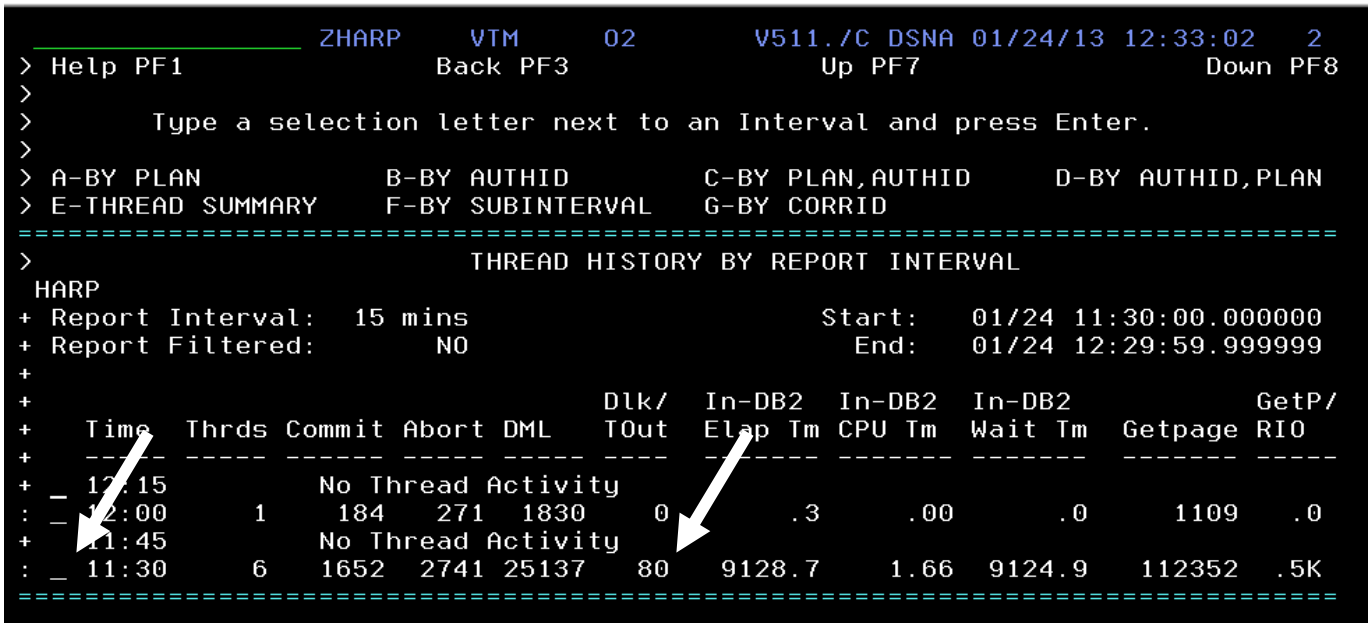
You are now looking at DB2 buffer pool statistics data for the same time interval. From here you can see relevant information such as getpage activity and buffer pool hit ratios for the DB2 subsystem for the given time interval.

You may also use the other letter commands to see other relevant categories of DB2 subsystem near term history, such as EDM pool, SQL counts, distributed processing statistics.

Another relevant question when using history to investigate a potential performance issue is to understand what DB2 application threads were executing wihtin the given time frame. OMEGAMON DB2 Near Term History also collects thread history information. Using the Near Term facility the user may see what DB2 threads completed within the given time interval.

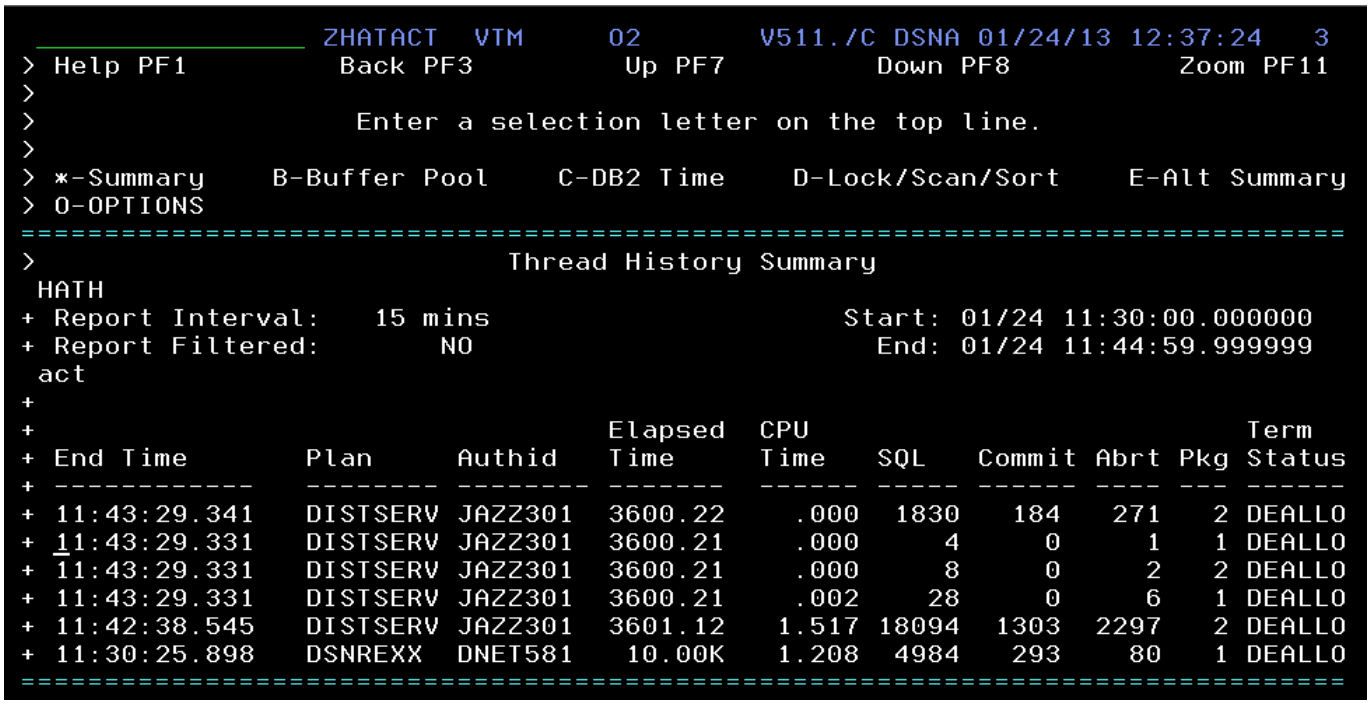
c) See an example of thread history

Position the cursor on the command line (upper left corner of the panel). Enter **M** and **Press Enter**.



You are now looking at a Near Term thread history display, sorted out by time interval. Note that in the middle of the display you can see the number of deadlocks/timeouts for that interval.

To see threads for a given time interval, p**osition the cursor** next to the desired time interval . Enter **E** (for Thread Summary) and **Press Enter**.



You are now looking at the DB2 thread history summary. From here you can drill down into a specific thread to see more detail on the thread.

To see more detail for a given thread, **position the cursor** on the desired thread **Press F11** to zoom in for detail.

```
_____________ ZHTACT   VTM     02       V511./C DSNA 01/24/13 12:41:34   2
> Help PF1              Back PF3              Up PF7                Down PF8

>          THREAD HISTORY:  Enter a selection letter on the top line.

> *-THREAD DETAIL   B-LOCK COUNTS  C-LOCK WAITS     D-GLOBAL LOCKS  E-SORT/SCAN
> F-DYNAMIC SQL     G-SQL COUNTS   H-DISTRIBUTED    I-BUFFER POOL   J-GROUP BP
> K-PACKAGE SUMMARY L-RES LIMIT    M-PARALLEL TASKS
================================================================================
>                            THREAD HISTORY DETAIL
 HPLN
+ Thread:  Plan=DSNREXX   Connid=DB2CALL  Corrid=DB2READ      Authid=DNET581
+ Attach:  CALLATCH       DB2=DSNA        MVS=MVSE
+ Time  :  Start=01/24/2013 08:43:45.437643     End=01/24/2013 11:30:25.898836
+ Luwid=USIBMNR.EOSDB201.CAD1A1F3DF75
 act
+ Termination Status    =         DEALLOC     Commits              =        293
+ Total Elapsed Time    =    02:46:40.461     Aborts               =         80
+ Total CP CPU Time     =    00:00:01.208
+ ZIIP CPU Time         =    00:00:00.000     Parallel Tasks       =          0
+ Total Stored Proc CPU =    00:00:00.000
+ Stored Proc Wait      =    00:00:00.000     Stored Proc Wait Cnt =          0
+
+ In-DB2 Times                                    Total
+ ----------------------------                    ------------
+ Elapsed Time                                    02:32:03.448
+ CP CPU Time                                     00:00:00.364
+ IIP CPU Time                                    00:00:00.000
+ Stored Procedure CPU Time                       00:00:00.000
+ UDF CP CPU Time                                 00:00:00.000
+ UDF IIP CPU Time                                00:00:00.000
+ UDF Elapsed Time Main                           00:00:00.000
+
+ Waits                                  Count    Total
+ ------------------------------- ----------    ------------
+ Synchronous I/O Wait                    0    00:00:00.000
+ Asynchronous Read I/O Wait              0    00:00:00.000
+ Asynchronous Write I/O Wait             0    00:00:00.000
+ Local Lock/Latch Wait                 160    02:32:02.898
```

You are now looking at detail for a specific DB2 thread. This is DB2 accounting information that shows such information as DB2 elapsed time, In-DB2 time, and DB2 wait time. From here you can analyze what the thread was doing and what bottlenecks it incurred.

This completes the scenario. Please **Press F3** multiple times until you return to the ZMENU main panel.

2.3 Analysis of DB2 threads and lock conflicts

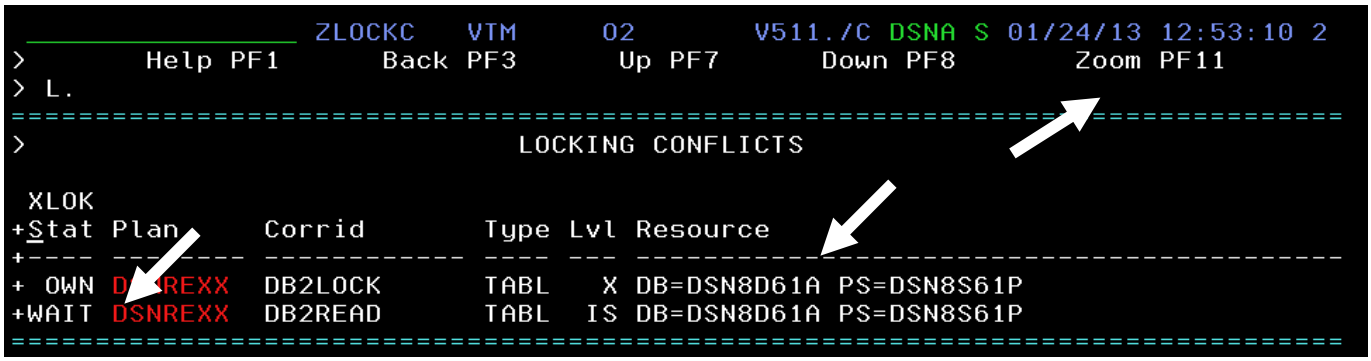
This scenario continues to explore the Classic 3270 interface details. This lab will explore how OMEGAMON for DB2 allows the user to identify and analyze DB2 lock conflicts in real time. From here we will also look at detailed thread analysis.

2.3.1 Identification and Isolation of Lock Conflicts

In this scenario you will see how to quickly identify DB2 threads experiencing lock conflicts while they are occurring.

a) From the ZMENU main menu select the Lock Conflict display

Position the cursor on the command line (upper left corner of the panel). Enter **L** and **Press Enter.**

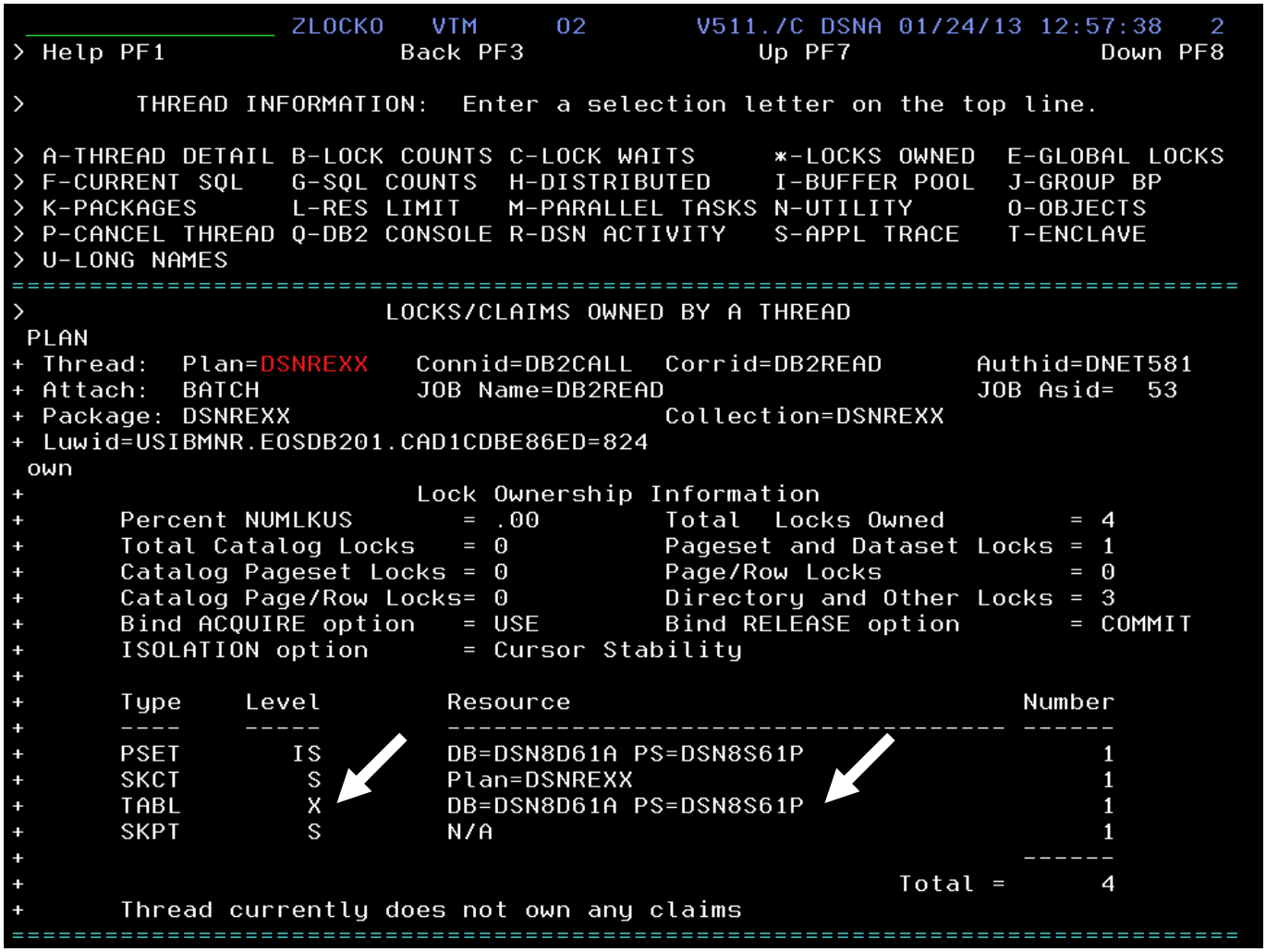


You are now looking at the OMEGAMON DB2 lock conflict display. This display will show the holder thread and the waiter thread for a given DB2 resource.

b) Determine what the thread is waiting on

From the lock conflict display you may drill down to see detail on the holder and waiter for the conflict resource. First note which thread is the OWNer and which thread is the WAITer. Also, note the resource being referenced in the lock conflict.

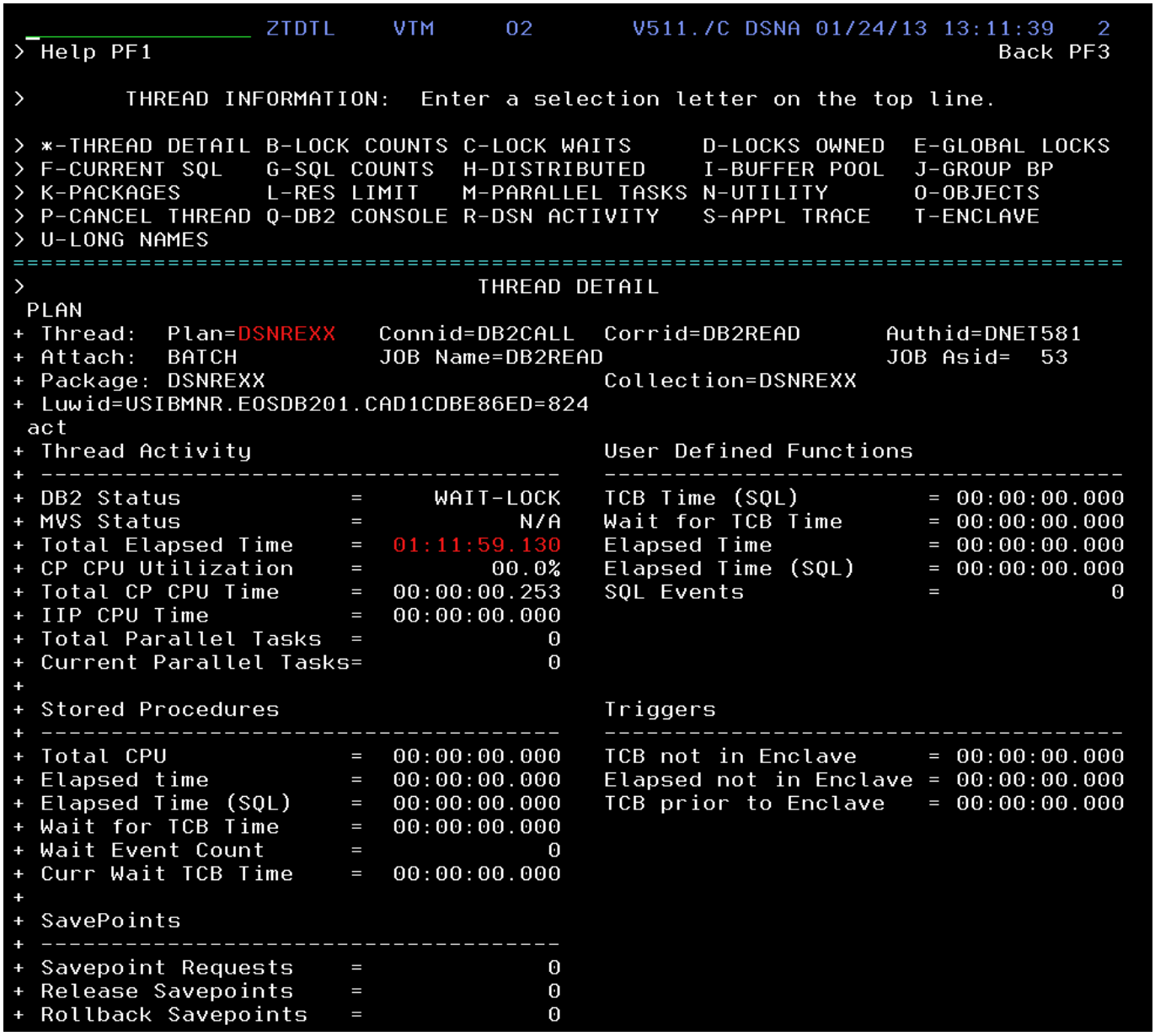
To see the detail, **position the cursor** on the WAIT thread and **Press F11** to zoom in for detail.



You are now looking at the thread lock detail for thread that is waiting for the lock resource. Here you can see the resource being locked by the thread, and the level of the lock being requested.

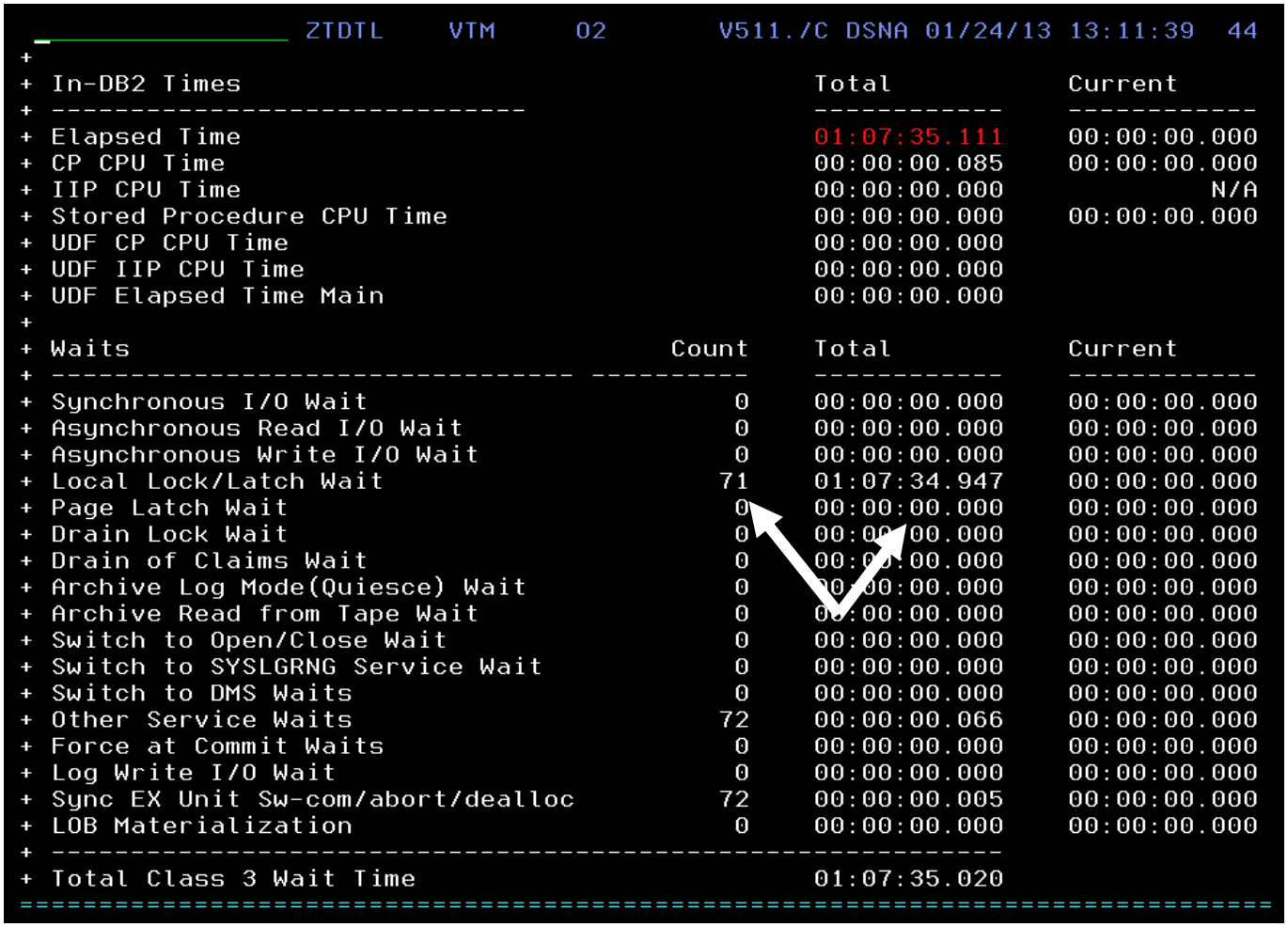
From this display you may look at other aspects of the thread data to detemine the impact of the lock conflicts on thread performance.

Position the cursor on the command line (upper left corner of the panel). Enter **A** and **Press Enter.**



You are now looking at the thread detail display which shows the DB2 thread Accounting class 1 (DB2 elapsed), class 2 (In-DB2) and class 3 (DB2 wait) times. To see to what extent the lock conflicts have impacted the thread scroll down to see the DB2 class 3 wait times.

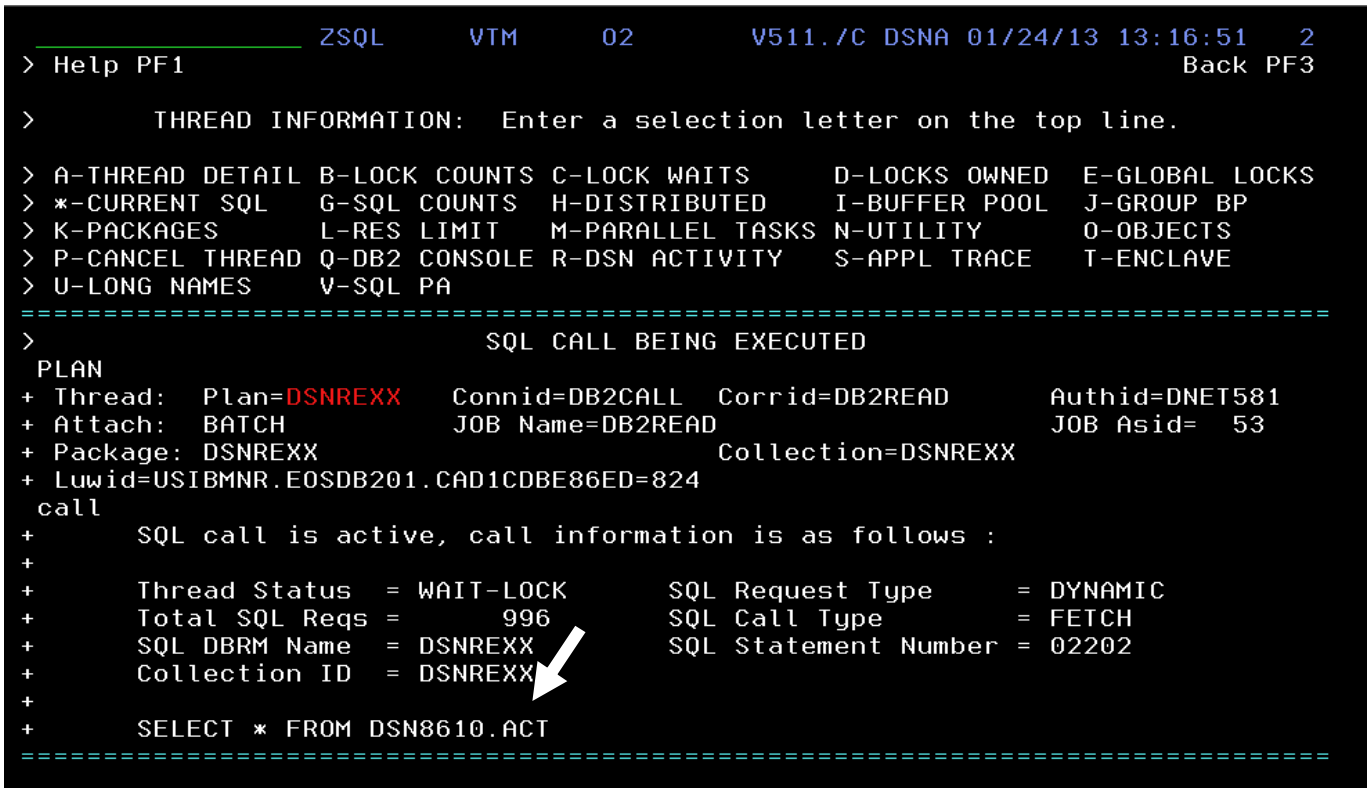
Press F8 to scoll the display.



You are now looking at the Class 3 DB2 wait times for the thread. Here you can see that the thread experienced 71 waits in this example, and the amount of time that the thread was delayed by the waits.

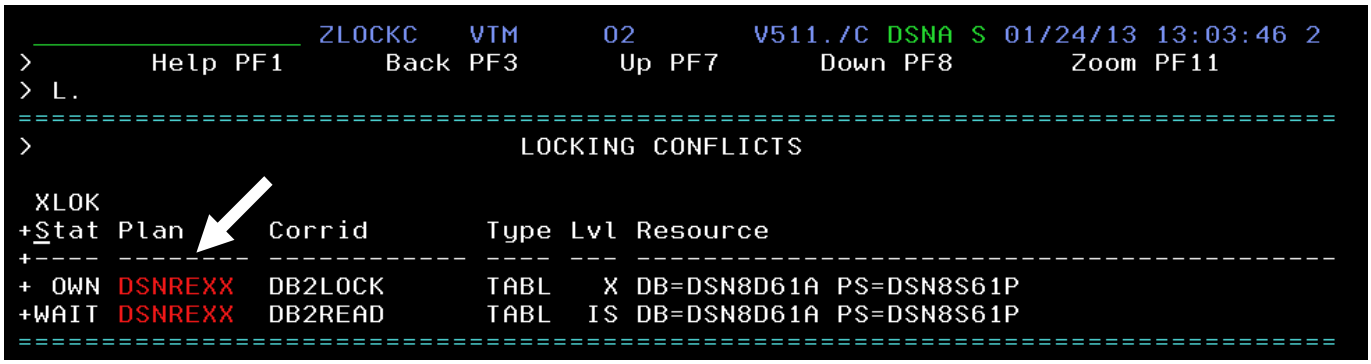
By using other letter commands on the top of the thread panel, you can also see information such as the number of SQL calls excuted by the thread, and see the SQL call currently being executed.

To see the current SQL call being excuted, p**osition the cursor** on the command line (upper left corner of the panel). Enter **F** and **Press Enter**



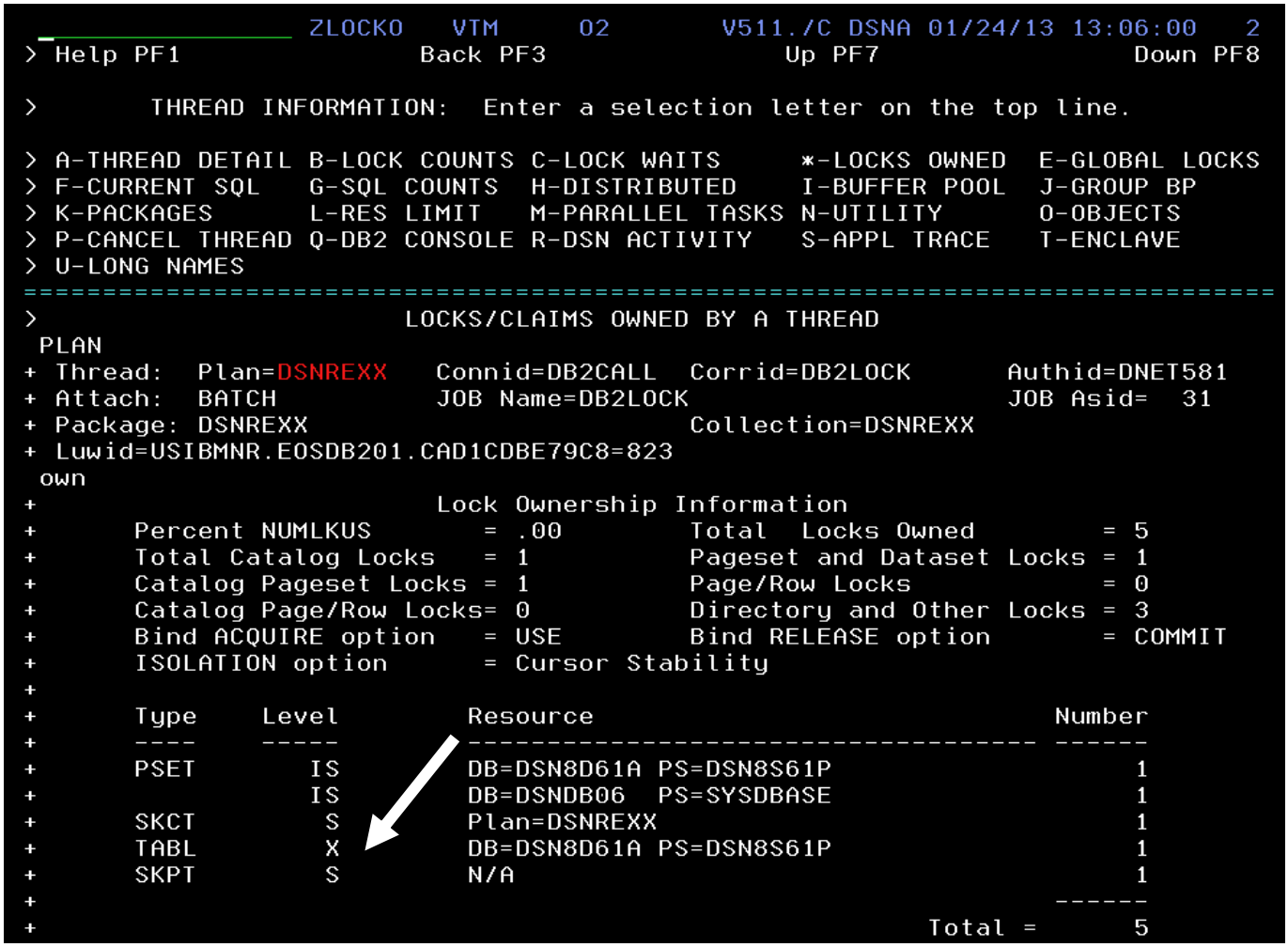
You may issue other letter commands to see other aspects of thread activity.

Once finished with the thread analysis, **Press F3** to return to the lock conflict display.



You are back at the locking conflict display. Next you may look at the thread that is owning the resource in question.

To see the detail, **position the cursor** on the OWN thread and **Press F11** to zoom in for detail.



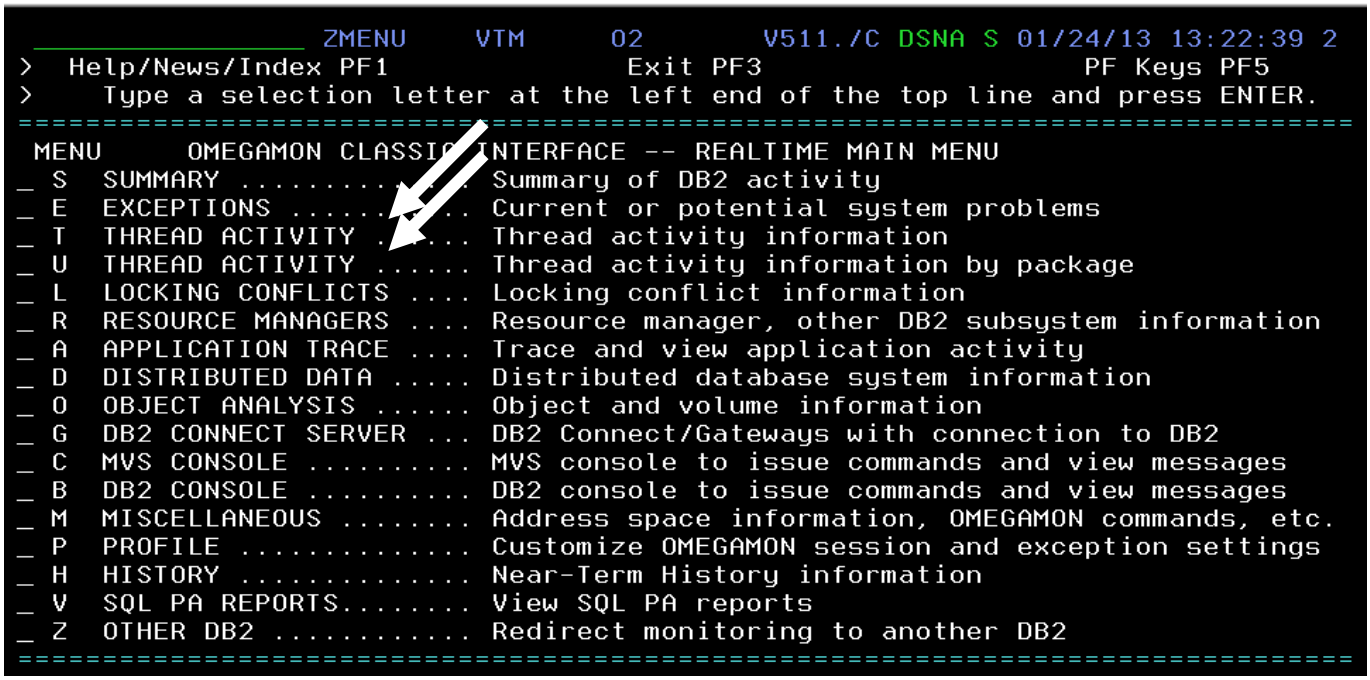
You are now looking at the thread detail for the thread holding the resource. The same commands may be applied that were used on the waiting thread.

Now that we have looked at the lock conflict scenario and identified the holding and waiting threads, return to the main menu ZMENU.

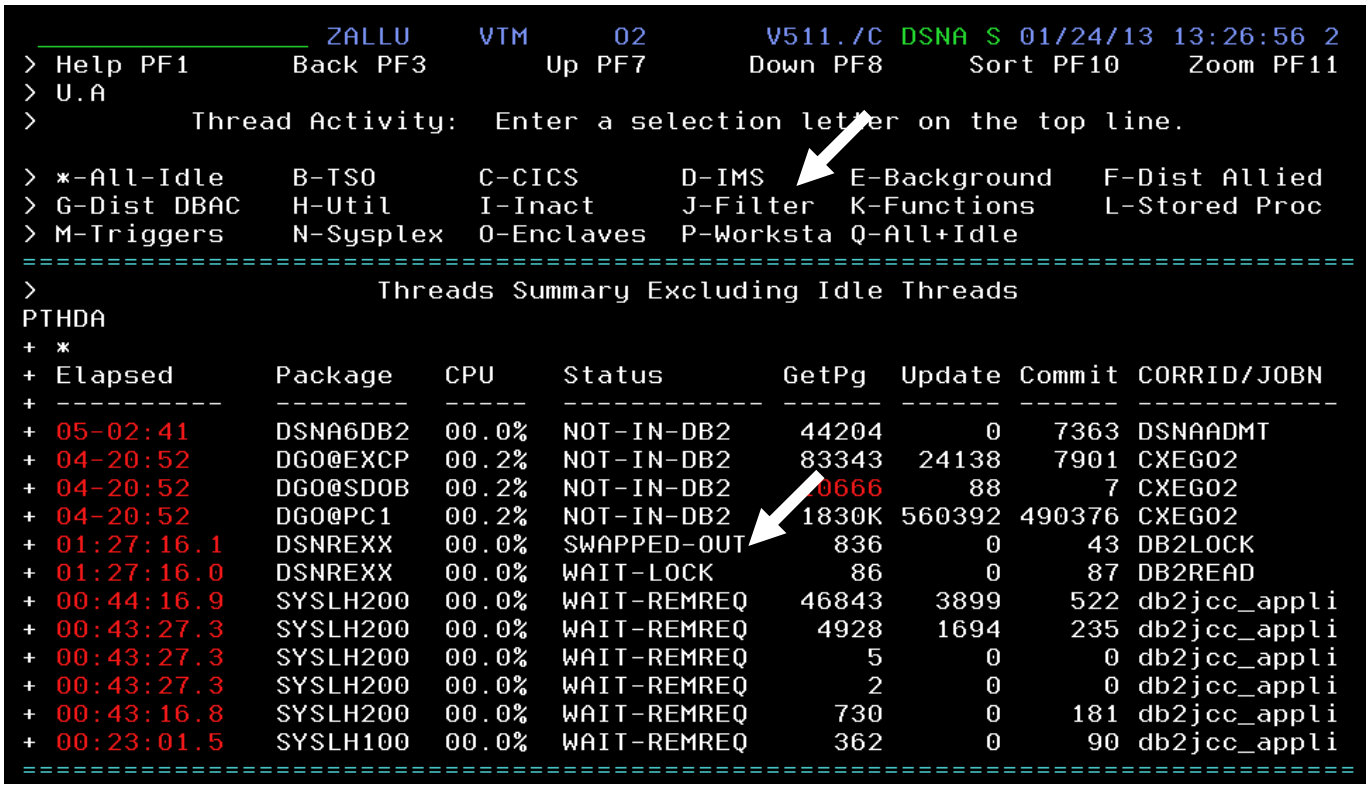
Press F3 and then **Press F3** again to return to ZMENU.

c) Look at DB2 thread activity

From the ZMENU main panel there are two ways to view DB2 threads. Option T shows threads listed by plan, Option U shows threads listed by DB2 package.

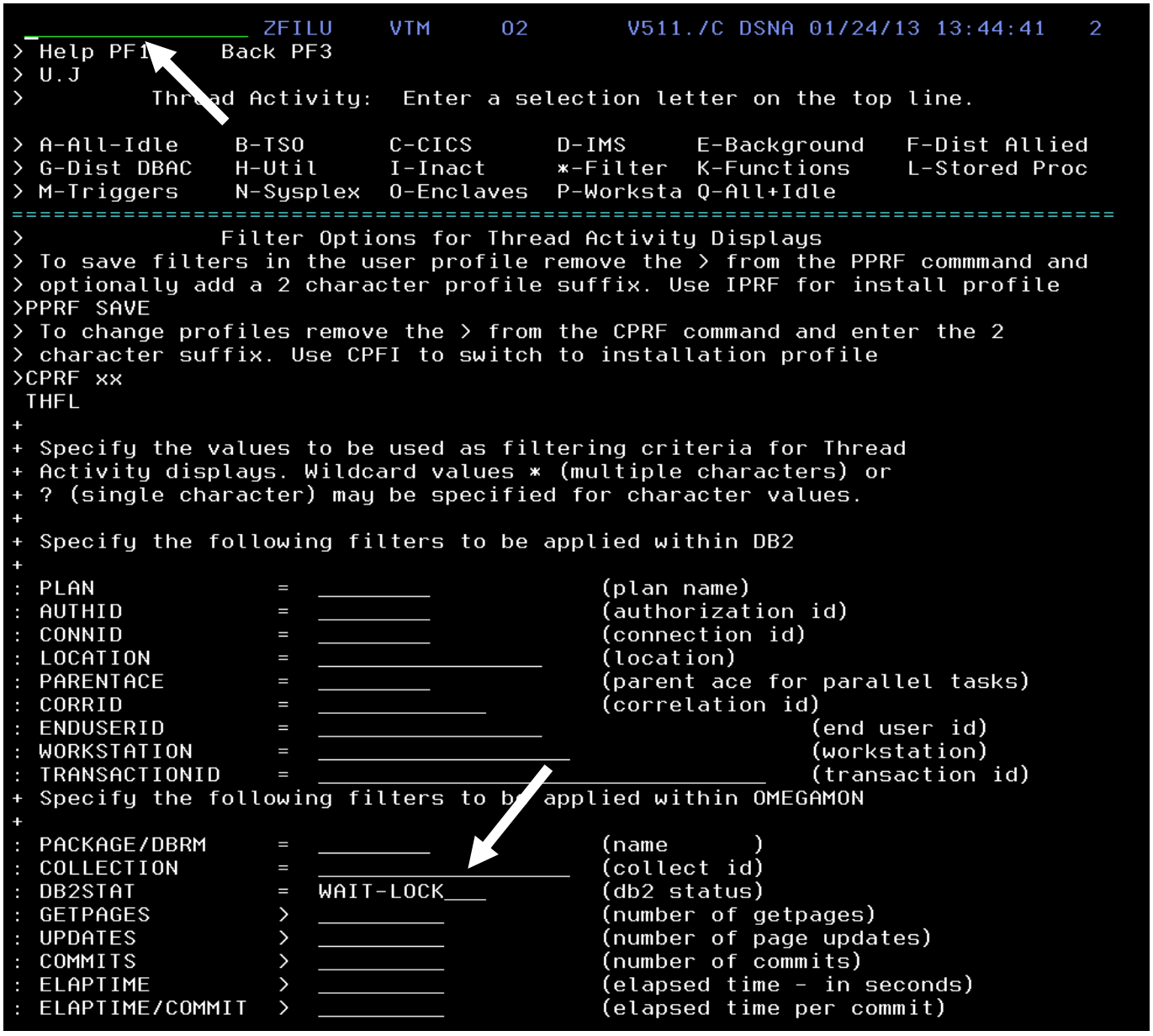


To see an overview of threads excuting in the subsystem, p**osition the cursor** on the command line (upper left corner of the panel). Enter **U** and **Press Enter.**



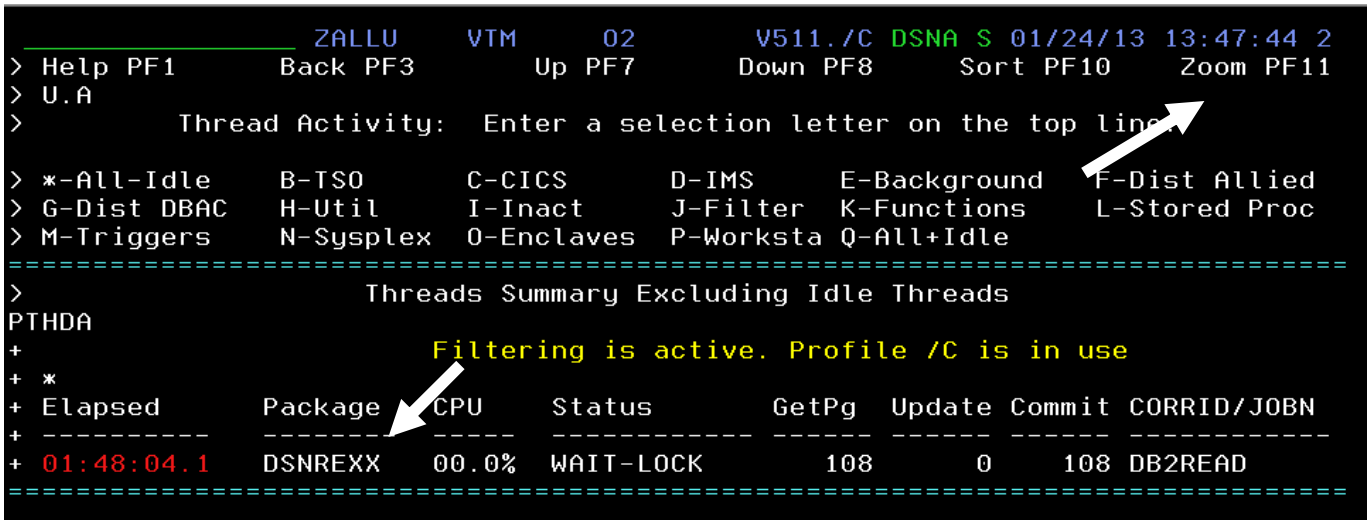
You are now looking at the DB2 thread overview listed by package. You have the ability to filter the threads viewed based on various criteria using option J as shown above.

Position the cursor on the command line (upper left corner of the panel). Enter **J** and **Press Enter.**



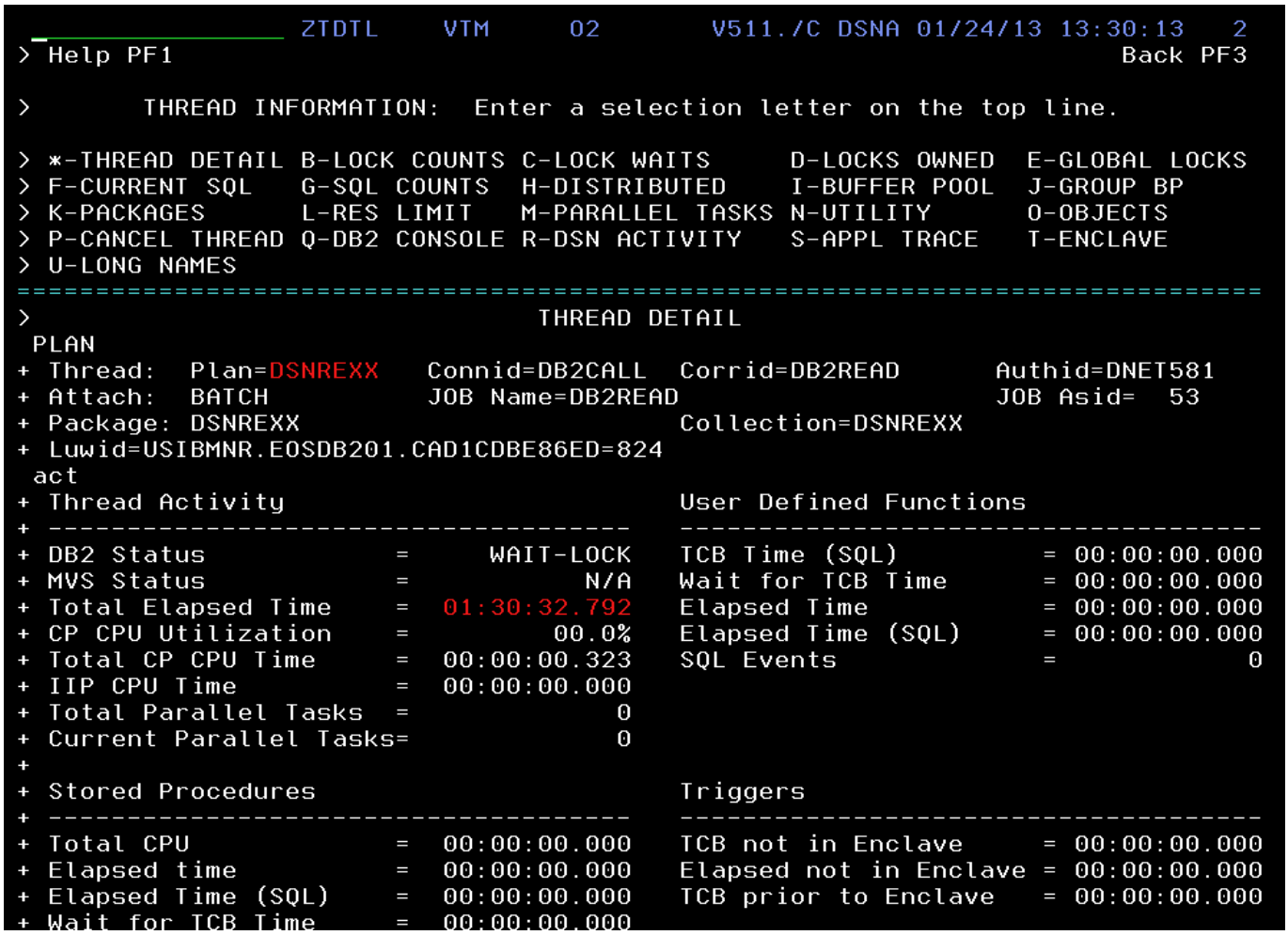
To filter the threads viewed and only view threads waiting for a lock, **Enter WAIT-LOCK** in the DB2STAT field.

Position the cursor on the command line (upper left corner of the panel). Enter **A** and **Press Enter.**



You are now looking at the thread display filtered to just show threads in a wait for lock status.

To see thread detail, **position the cursor** on the specific thread **Press F11** to zoom in for detail



You have now seen several ways to analyze DB2 lock conflicts, how to display threads with lock conflicts, how to detrermine the impact of lock conflicts on a thread, and finally how to filter and manage various thread views.

Once you are finished looking at the thread views**, Press F3** until you have returned to the ZMENU main menu.

You have now completed the scenario of the OMEGAMON DB2 module.

2.4 How to view DB2 application trace information

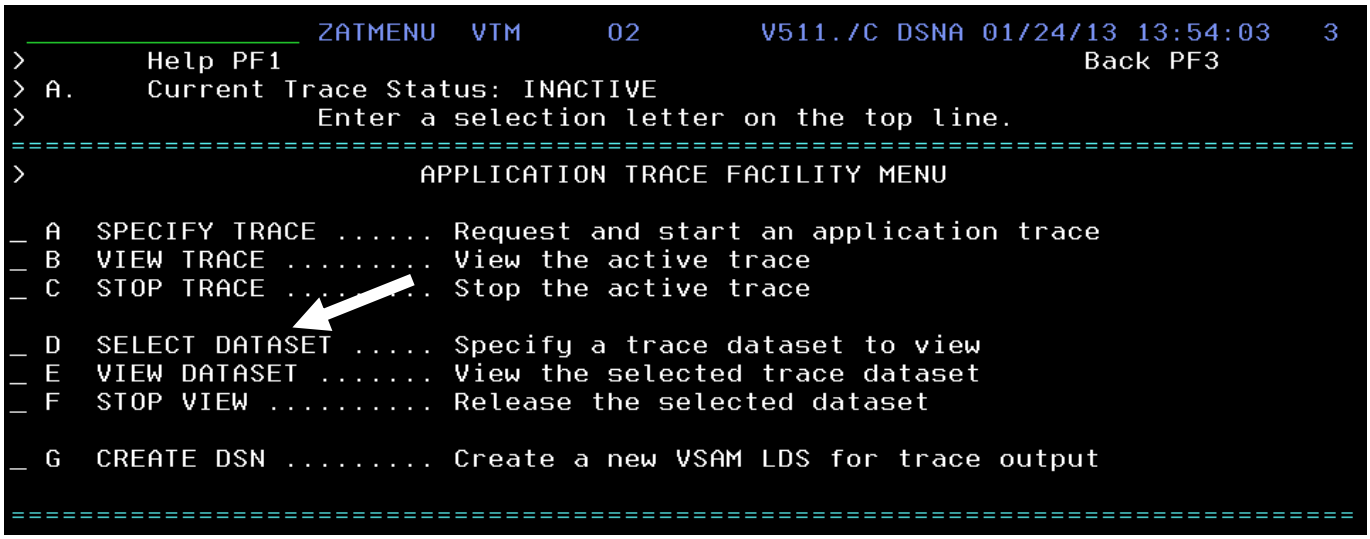
This scenario continues to explore the Classic interface details. One of the powerful features of the Classic interface is the OMEGAMON DB2 application trace facility. The OMEGAMON DB2 application trace facility is able to capture traces of DB2 application executions, and retain this detailed information for later analysis.

This scenario will explore how to navigate within OMEGAMON to use the application trace function.

2.4.1 Look at OMEGAMON DB2 trace information

a) From the ZMENU main panel drill down to see Application trace information

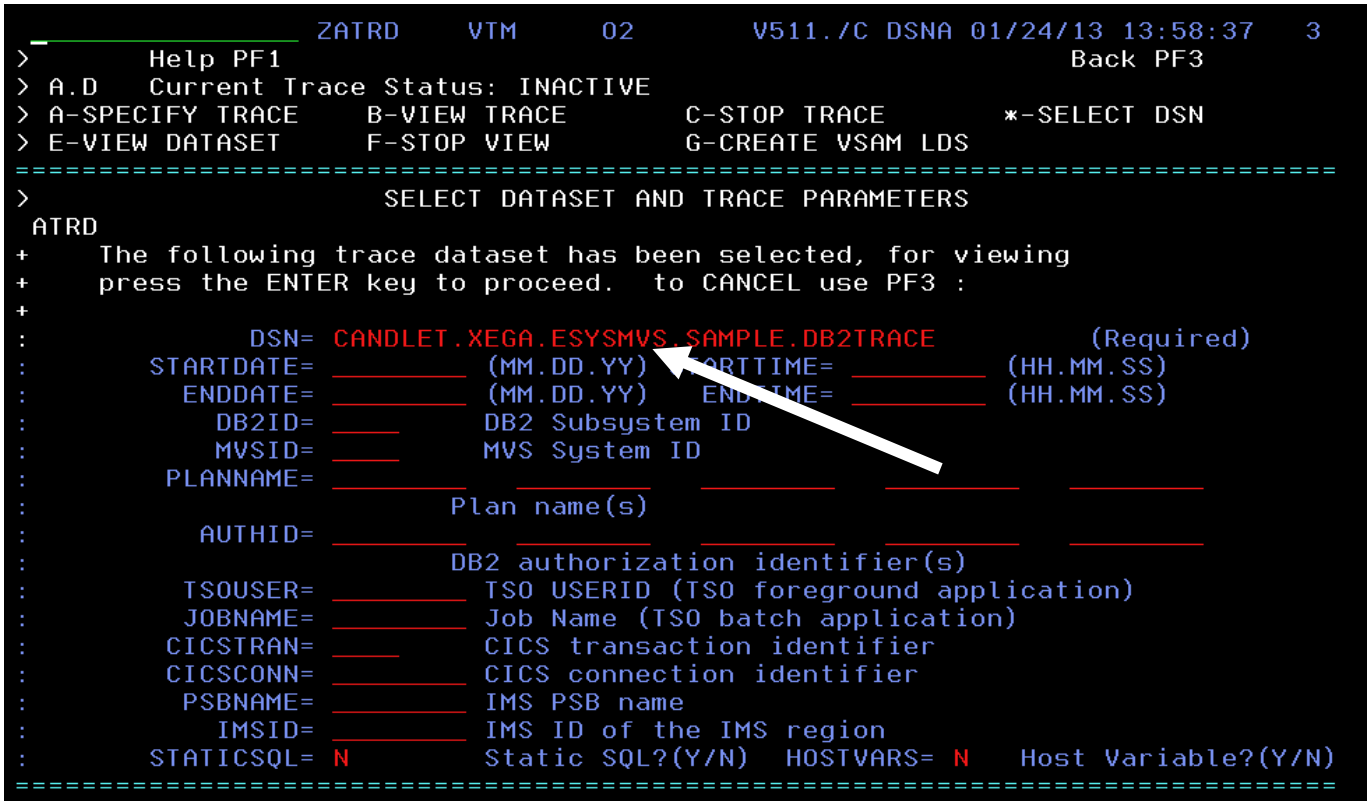
Position the cursor on the command line (upper left corner of the panel). Enter **A** and **Press Enter.**



You are now looking at the OMEGAMON DB2 application trace facility menu. From here you may go to panels to start/stop traces, and view the trace output. Traces may be done to memory buffers within the OMEGAMON task, or be traced to a dataset for later analysis.

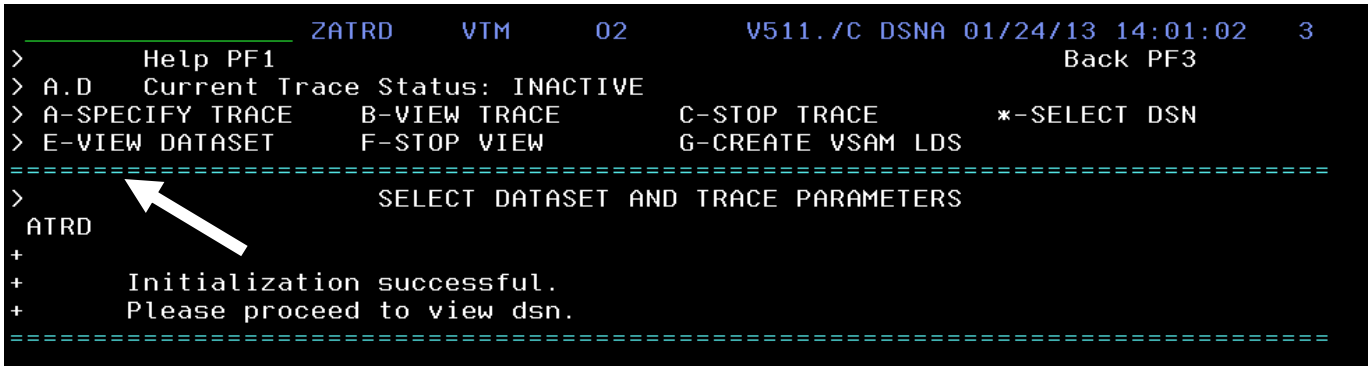
In this lab Scenario 4 you will not be running traces, but will be looking at trace data already captured in an OMEGAMON DB2 trace dataset.

Position the cursor on the command line (upper left corner of the panel). Enter **D** and **Press Enter.**



You are now looking at the DB2 trace specification panel. From here you can enter the trace dataset name.

Postion the cursor in the DSN= field enter the dataset name CANDLET.XEGA.ESYSMVS.SAMPLE.DB2TRACE and **press Enter**.



b) View DB2 trace information

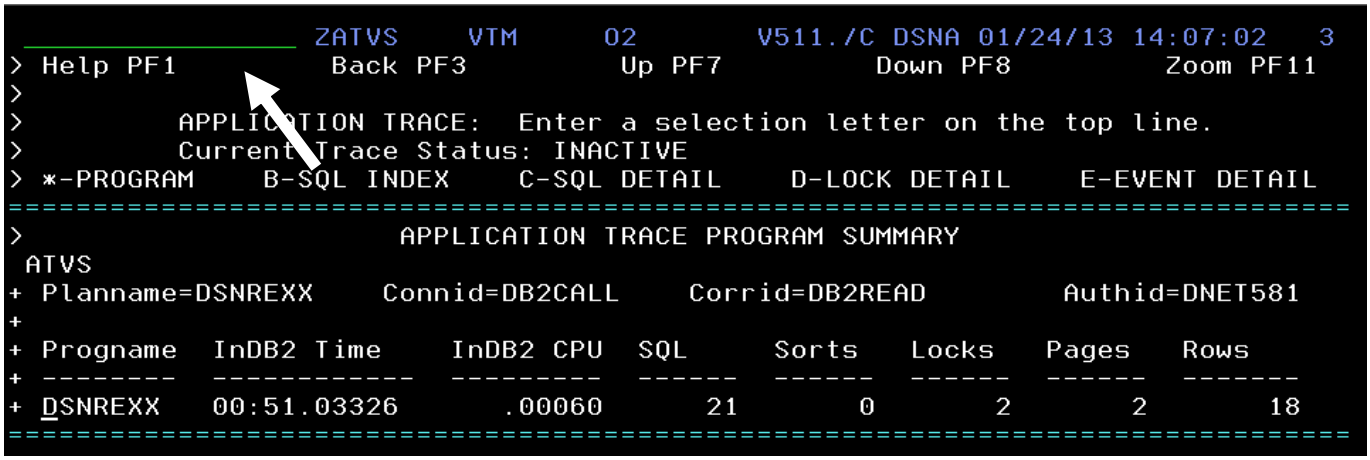
To view the trace information, position the cursor on the command line (upper left corner of the panel). Enter **E** and **Press Enter.**

```
_____________ ZATVD    VTM     02       V511./C DSNA 01/24/13 14:02:35   3
>       Help PF1      Back PF3      Up PF7      Down PF8       Zoom PF11
> A.E   Current Trace Status: INACTIVE
> A-SPECIFY TRACE    B-VIEW TRACE       C-STOP TRACE       D-SELECT DSN
> *-VIEW DATASET     F-STOP VIEW        G-CREATE VSAM LDS
================================================================================
>               APPLICATION TRACE THREAD SUMMARY -- VIEW DATASET
 ATVD
+
+ DSN               = CANDLET.XEGA.ESYSMVS.SAMPLE.DB2TRACE
+
+ Planname Connid   Corrid      Authid   DB2  MVS  InDB2 CPU Commits  Aborts
+ -------- -------- ----------- -------- ---- ---- --------- -------- ------
+ DSNREXX  DB2CALL  DB2READ     DNET581  DSNA MVSE    .00166        2      2
+ DSNREXX  DB2CALL  DB2LOCK     DNET581  DSNA MVSE    .00122        1      0
================================================================================
```

You are now looking at the trace information. Here you see two threads that have been traced. To see trace detail, **position the cursor** on a specific thread and **Press F11** to zoom in for trace detail.

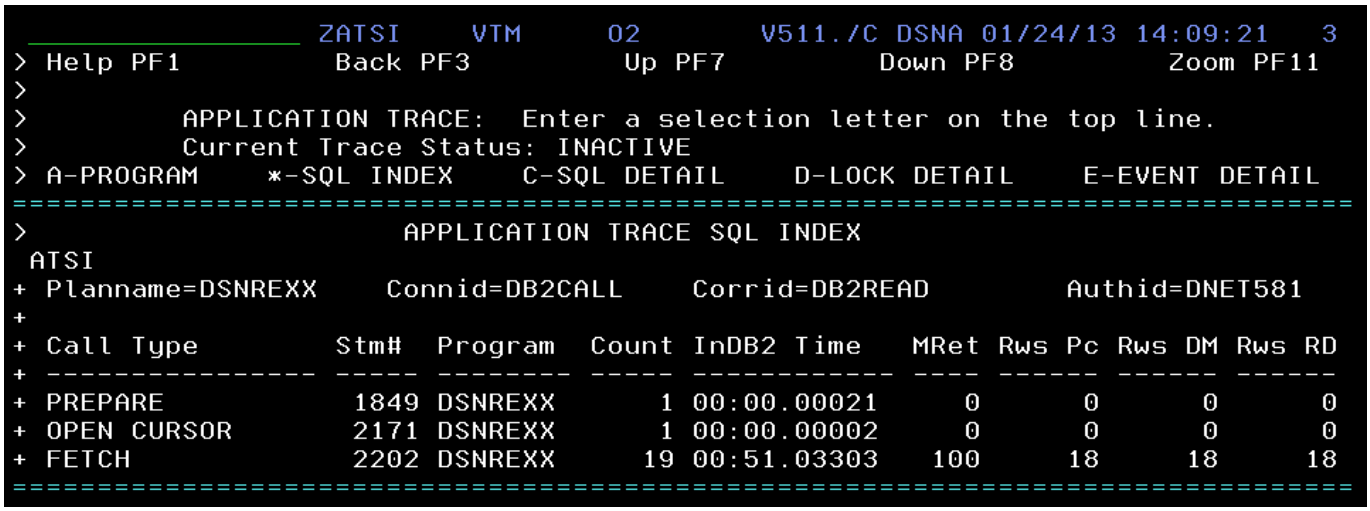
```
_____________ ZATVC    VTM     02       V511./C DSNA 01/24/13 14:04:53   3
> Help PF1       Back PF3       Up PF7       Down PF8       Zoom PF11
>       Current Trace Status: INACTIVE
================================================================================
>                   APPLICATION TRACE UNIT OF WORK SUMMARY
 ATVC
+ Planname=DSNREXX    Connid=DB2CALL    Corrid=DB2READ       Authid=DNET581
+
+ Date   Start Time  Progname InDB2 Time   InDB2CPU  SQL    Sort Locks  Rows
+ _____ ____________ -------- ------------ --------- ------ ---- ------ -------
+ 01/24 09:04:51.473 DSNREXX  00:00.00016     .00016      2    0      2       0
+ 01/24 09:03:45.506 DSNREXX  00:00.00018     .00016      2    0      2       0
+*01/24 09:02:51.468 DSNREXX  00:51.03326     .00060     21    0      3      18
+ 01/24 09:01:45.497 DSNREXX  00:00.00012     .00011      2    0      2       0
+*01/24 09:00:51.464 DSNREXX  00:51.02806     .00063     21    0      3      18
================================================================================
```

You are now looking at the Unit of Work summary for the traced DB2 thread. The next step is to drill down to SQL call level detail. To see call level detail, **position the cursor** on a specific thread and **Press F11** to zoom in for trace detail.



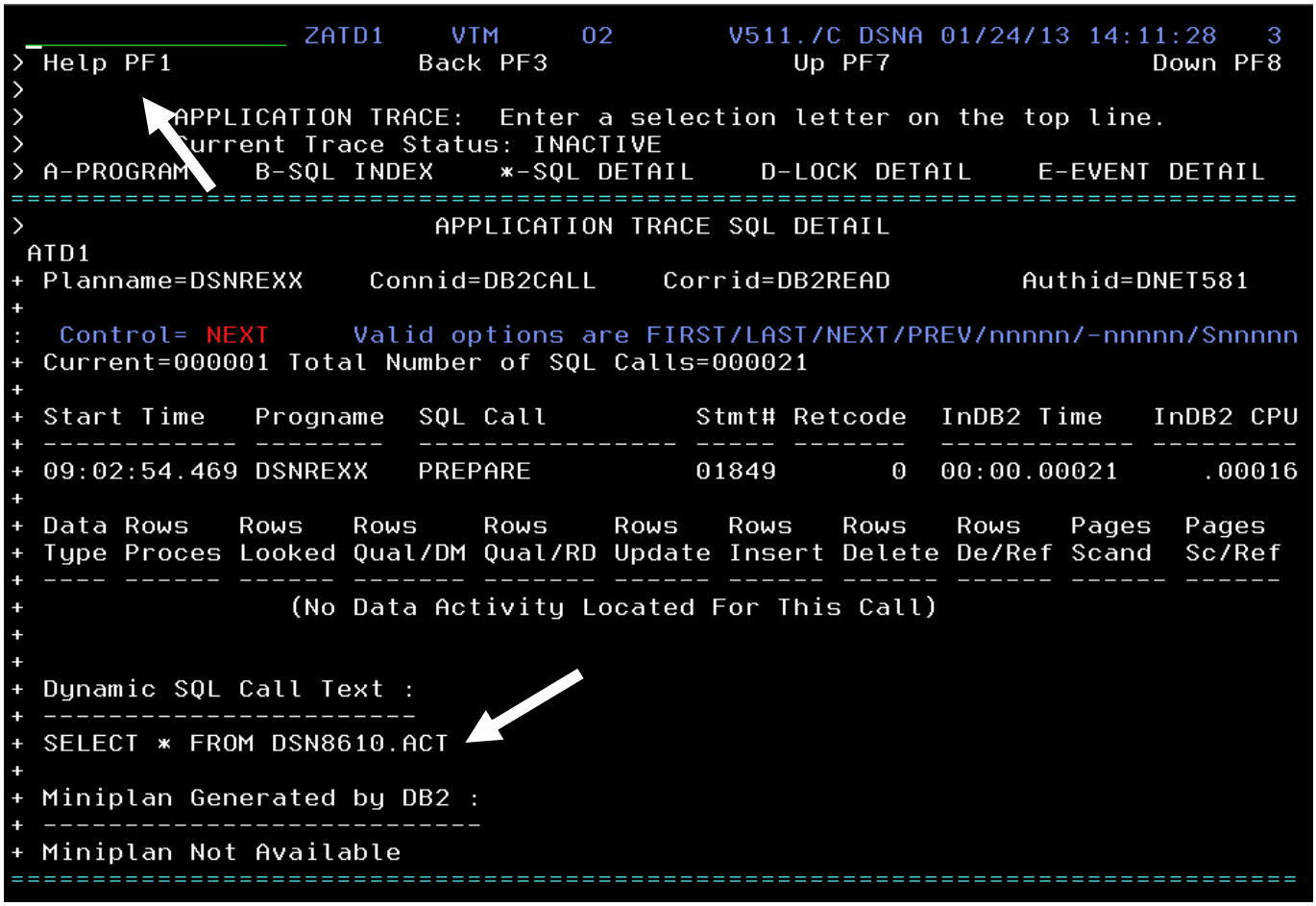
You are now looking at the summary for the application trace. To see more detail use the letter commands at the top of the panel.

To view the SQL Index, **position the cursor** on the command line (upper left corner of the panel). Enter **B** and **Press Enter.**



You are now looking at the SQL Index for the trace. This shows what calls were executed, how many times a call statement was executed, and how many rows were processed by the calls.

To view the SQL Detail, **position the cursor** on the command line (upper left corner of the panel). Enter **C** and **Press Enter.**



Your are now looking at the SQL Detail, Here is where you can see information such as the SQL call text if the call is dynamic SQL.

To view the Lock Detail**,** **position the cursor** on the command line (upper left corner of the panel). Enter **D** and **Press Enter.**

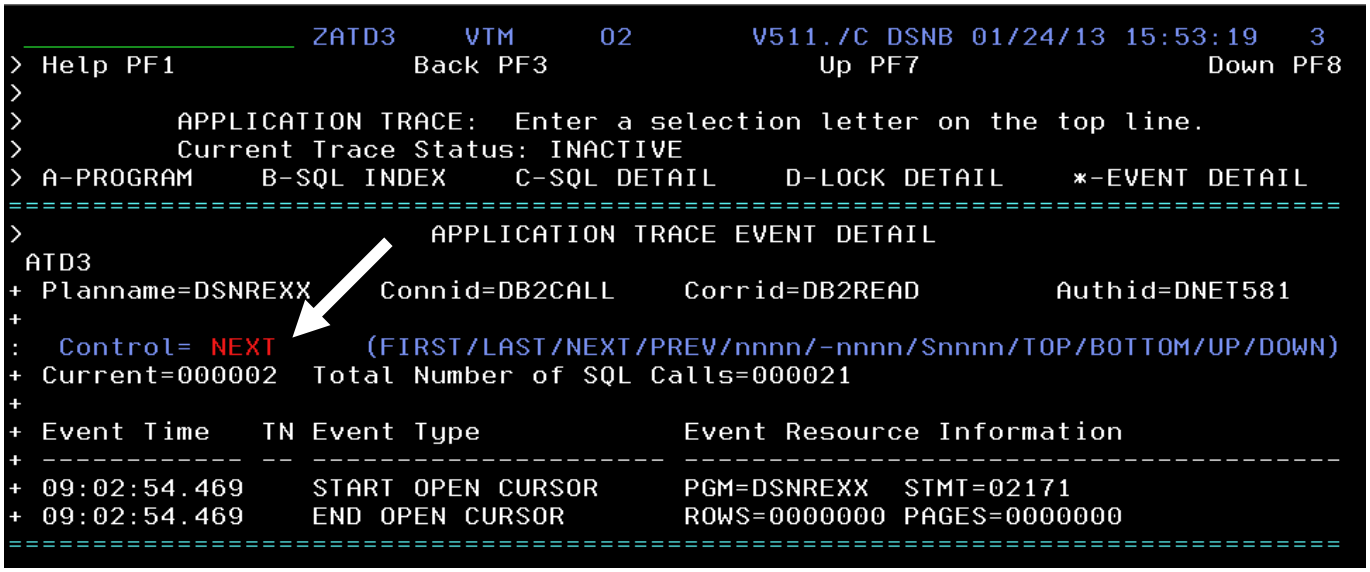
```
_______________ ZATD2    VTM     02       V511./C DSNA 01/24/13 14:13:21   3
> Help PF1              Back PF3               Up PF7                 Down PF8
>
>         APPLICATION TRACE:  Enter a selection letter on the top line.
>         Current Trace Status: INACTIVE
> A-PROGRAM    B-SQL INDEX    C-SQL DETAIL    *-LOCK DETAIL    E-EVENT DETAIL
================================================================================
>                        APPLICATION TRACE LOCK DETAIL
 ATD2
+ Planname=DSNREXX    Connid=DB2CALL    Corrid=DB2READ        Authid=DNET581
+
:  Control= FIRST    Valid options are FIRST/LAST/NEXT/PREV/nnnnn/-nnnnn/Snnnnn
+ Current=000001  Total Number of SQL Calls=000021
+
+ Start Time   Progname  SQL Call         Stmt#  PSET    DPAG    IPAG    OTHER
+ ------------ --------  ---------------- -----  ------  ------  ------  ------
+ 09:02:54.469 DSNREXX   PREPARE          01849       0       0       0       0
+
+                        Locks Acquired By Sql Call
+          ------------------------------------------------------
+                   (No Locks Were Acquired By SQL Call)
+
+                 Locks Owned At Start Of SQL Call Execution
+          ------------------------------------------------------
+          Type     Level   Resource                        Count
+          ----     -----   ------------------------------  -------
+          SKPT      S      TOKEN=1847604208116CAA                1
+                                                           -------
+                                       Total Locks Owned =       1
================================================================================
```

To view the Event Detail**,** **position the cursor** on the command line (upper left corner of the panel). Enter **E** and **Press Enter.**

```
_______________ ZATD3    VTM     02       V511./C DSNA 01/24/13 14:15:02   3
> Help PF1              Back PF3               Up PF7                 Down PF8
>
>         APPLICATION TRACE:  Enter a selection letter on the top line.
>         Current Trace Status: INACTIVE
> A-PROGRAM    B-SQL INDEX    C-SQL DETAIL    D-LOCK DETAIL    *-EVENT DETAIL
================================================================================
>                        APPLICATION TRACE EVENT DETAIL
 ATD3
+ Planname=DSNREXX    Connid=DB2CALL    Corrid=DB2READ        Authid=DNET581
+
:  Control= NEXT     (FIRST/LAST/NEXT/PREV/nnnn/-nnnn/Snnnn/TOP/BOTTOM/UP/DOWN)
+ Current=000001  Total Number of SQL Calls=000021
+
+ Event Time   TN Event Type           Event Resource Information
+ ------------ -- -------------------- ----------------------------------------
+ 09:02:54.469    START PREPARE        PGM=DSNREXX  STMT=01849
+ 09:02:54.469    END PREPARE          ROWS=0000000 PAGES=0000000
================================================================================
```

You are now looking at the event detail level of tracing for the application. The event detail will show the start stop times for each event as part of the DB2 application.

To see the next event in the trace **Press Enter**, then **Press Enter** again to see subsequent events. Note that the relative statement number increments each time you press enter.



Feel free to step through the DB2 trace and look at the various events and records documented by the application trace.

Congratulations! You have completed the OMEGAMON DB2 Classic interface lab.

Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Appendix B. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Appendix C. Documentation Revision History

Date of Revision	Number	Completed by	Revision Log
9/9/2014	V6.0	Ed Woods	Principal author Lab design and lab document creation Combined into one integrated lab doc.
7/13/2015	V7.0	Ed Woods	Updated e3270 module for V5.30
8/6/2015	V7.1	Lih Wang	Edited for SHARE August-Orlando Lab Session

NOTES

NOTES

Please Recycle

IBM Software