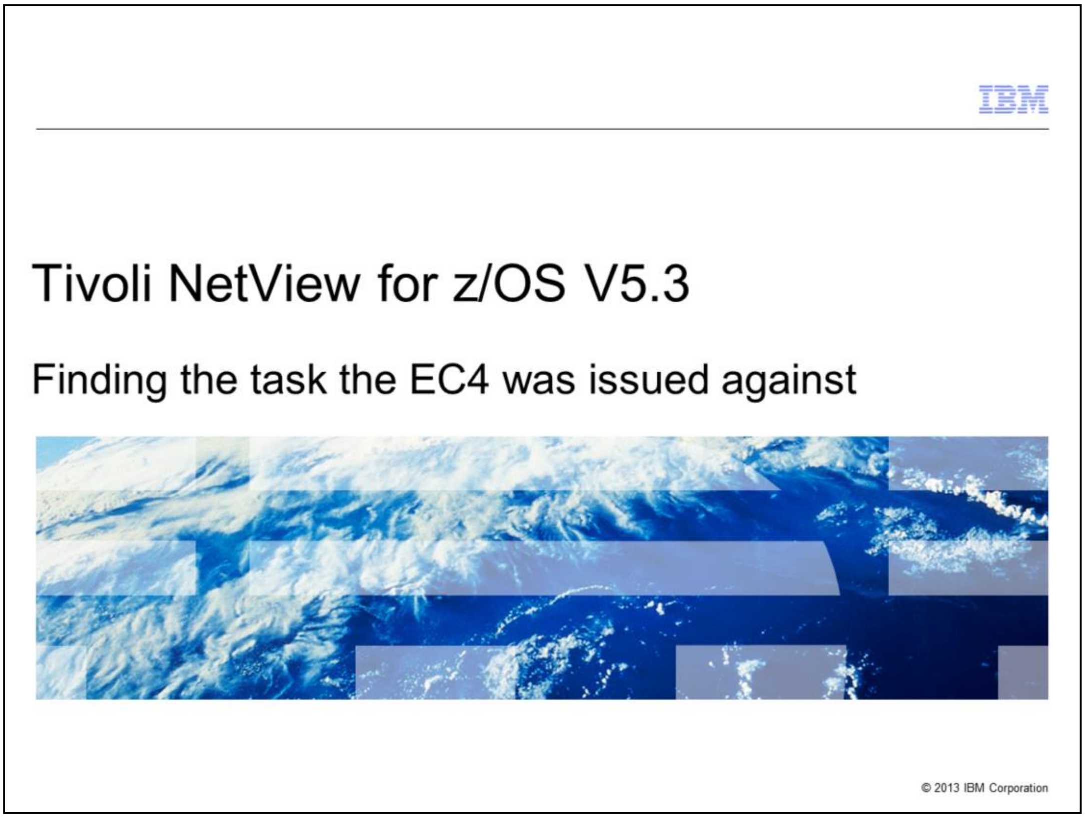


Tivoli NetView for z/OS V5.3

Finding the task the EC4 was issued against

Tivoli® NetView® for z/OS® V5.3, Finding the task the EC4 was issued against.

This module is about finding the resource that did not terminate within a minute and caused NetView to issue an ABENDEC4. It is for a single EC4 abend. The process does not work if there are multiple EC4 abends.

Objectives

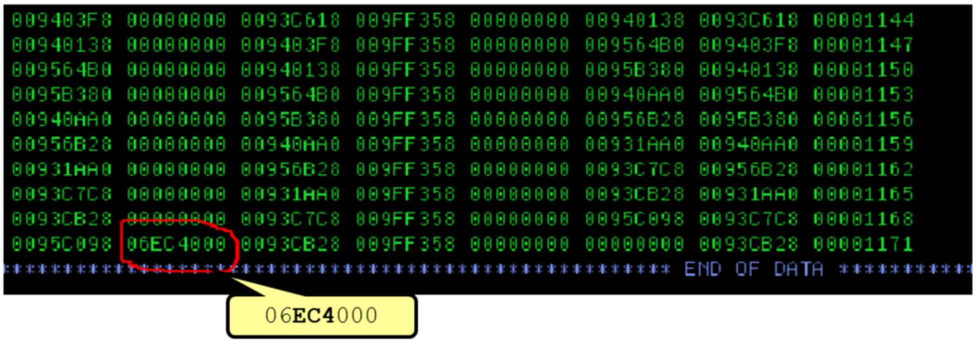
When you complete this module, you can locate the resource that did not terminate within a minute, which caused NetView to issue an ABENDEC4

2 Finding the task the EC4 was issued against © 2013 IBM Corporation

When you complete this module, you can locate the resource that did not terminate within a minute, which caused NetView to issue an ABENDEC4.

EC4 problem determination (1 of 2)

- Run the command **IP SUMM FORMAT** in the dump
- Max to the bottom
- Look for the **TCB** that shows an **EC4** abend



```
009403F8 00000000 0093C618 009FF358 00000000 00940138 0093C618 00001144
00940138 00000000 009403F8 009FF358 00000000 009564B0 009403F8 00001147
009564B0 00000000 00940138 009FF358 00000000 0095B380 00940138 00001150
0095B380 00000000 009564B0 009FF358 00000000 00940AA0 009564B0 00001153
00940AA0 00000000 0095B380 009FF358 00000000 00956B28 0095B380 00001156
00956B28 00000000 00940AA0 009FF358 00000000 00931AA0 00940AA0 00001159
00931AA0 00000000 00956B28 009FF358 00000000 0093C7C8 00956B28 00001162
0093C7C8 00000000 00931AA0 009FF358 00000000 0093CB28 00931AA0 00001165
0093CB28 00000000 0093C7C8 009FF358 00000000 0095C098 0093C7C8 00001168
0095C098 06EC4000 0093CB28 009FF358 00000000 00000000 0093CB28 00001171
********************************************************** END OF DATA ************
```

06**EC4**000

3 Finding the task the EC4 was issued against © 2013 IBM Corporation

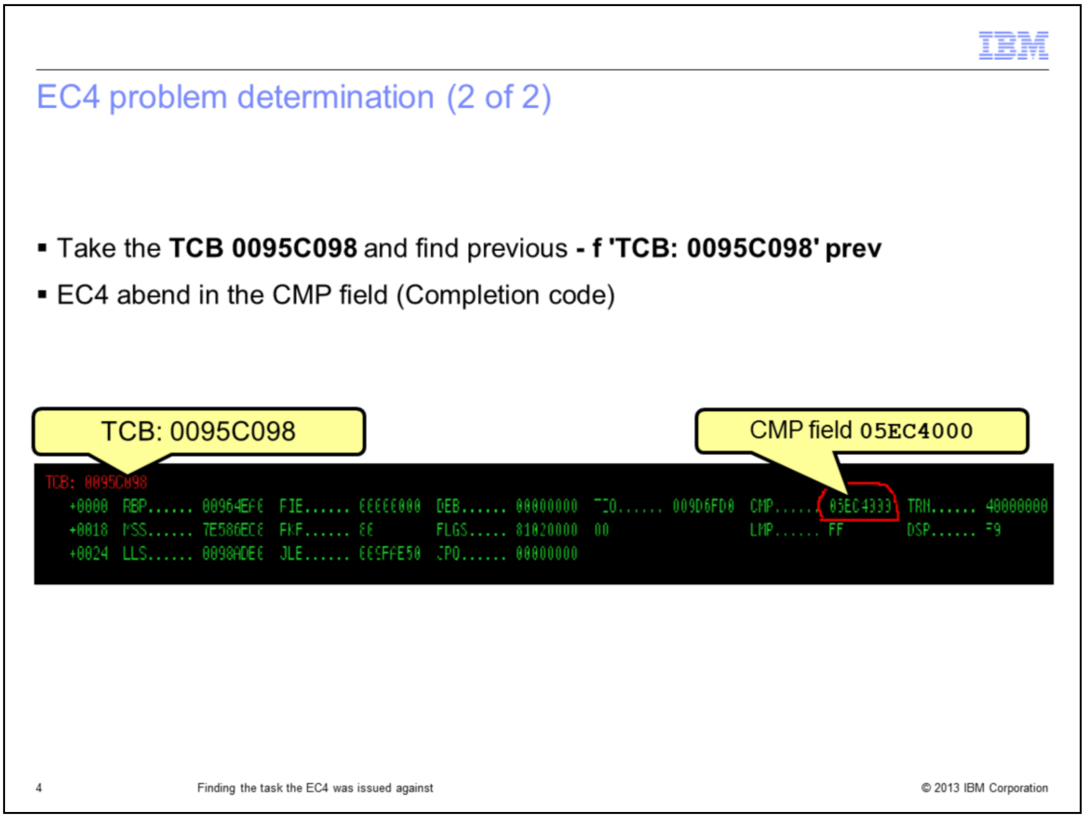
To start, run the issuing command **IP SUMM FORMAT** in the dump. Max to the bottom and start looking for the TCB that shows an **EC4 abend**.

An example of this JCL can be found in the *Tivoli Information Management for z/OS Planning and Install Guide* GC31-8751-00.

The **&SYSUID** symbol resolves to the submitting user ID.

Include any user-specific libraries in the relative DD statements.

The **SYSTSIN** is an example of running a **Search**; then the **Quit** command to exit InfoMan.



EC4 problem determination (2 of 2)

- Take the **TCB 0095C098** and find previous **- f 'TCB: 0095C098' prev**
- EC4 abend in the CMP field (Completion code)

TCB: 0095C098

CMP field **05EC4000**

```
TCB: 0095C098
   +0000  RBP...... 00964EFE  FIE...... EEEEE000  DEB...... 00000000  TIO...... 009D6FD0  CMP...... 05EC4333  TRN...... 40000000
   +0018  MSS...... 7E586ECE  FKF...... EE        FLGS..... 81020000  00                  LMP...... FF        DSP...... F9
   +0024  LLS...... 0098ADEE  JLE...... EE9FFE50  JPO...... 00000000
```

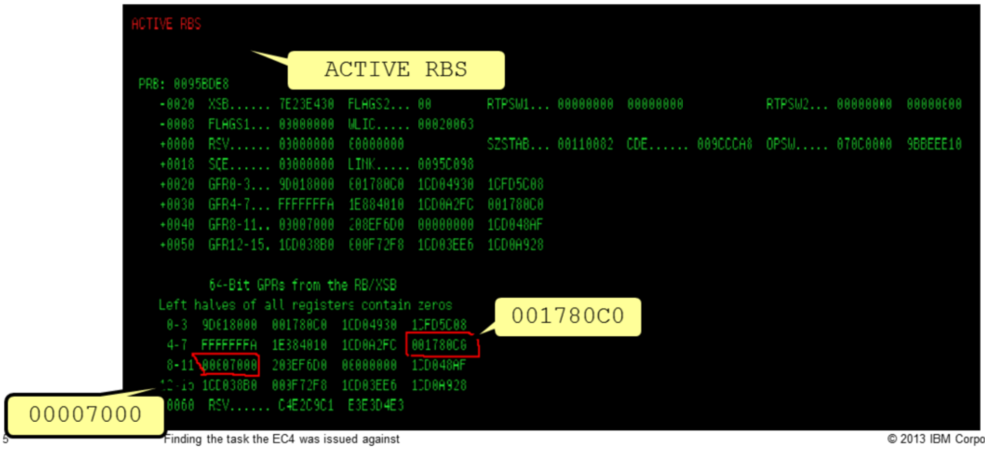
4 Finding the task the EC4 was issued against © 2013 IBM Corporation

Run the **ISPSTART** command to invoke the **BLGINIT** program to initialize your InfoMan session that the batch job uses to log on to InfoMan and perform the task.

For more information, see *Tivoli Information Management for z/OS Planning and Install Guide* GC31-8751-00.

Correlating the TCB to a TVB

- Locate ACTIVE RBS
- Correlate the TCB to a TVB
- Look at the location in storage



```
ACTIVE RBS

 PRB: 0095BDE8
   -0020  XSB...... 7E23E430  FLAGS2... 00        RTPSW1... 00000000  00000000          RTPSW2... 00000000  00000E00
   -0008  FLAGS1... 00000000  WLIC..... 00020063
   +0000  RSV...... 00000000  00000000            SZSTAB... 00110082  CDE...... 009CCCA8  OPSW..... 070C0000  9BBEEE10
   +0018  SCE...... 00000000  LINK..... 0095C098
   +0020  GFR0-3... 9D018000  001780C0  1CD04930  1CFD5C08
   +0030  GFR4-7... FFFFFFFA  1E884010  1CD0A2FC  001780C0
   +0040  GFR8-11.. 00007000  208EF6D0  00000000  1CD048AF
   +0050  GFR12-15. 1CD038B0  000F72F8  1CD03EE6  1CD0A928

          64-Bit GPRs from the RB/XSB
   Left halves of all registers contain zeros
    0-3   9D018000  001780C0  1CD04930  1CFD5C08
    4-7   FFFFFFFA  1E884010  1CD0A2FC  001780C0
    8-11  00007000  208EF6D0  00000000  1CD048AF
   12-15  1CD038B0  000F72F8  1CD03EE6  1CD0A928
   +0060  RSV...... C4E2C9C1  E3E3D4E3
```

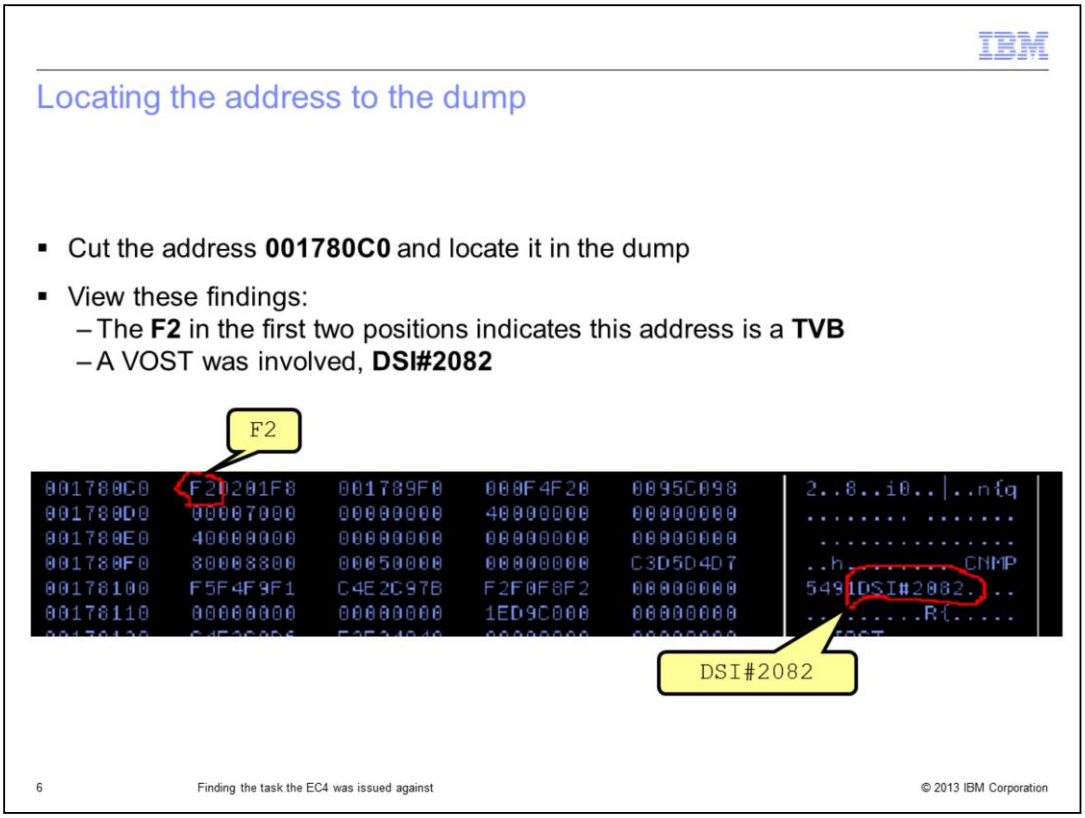
5 Finding the task the EC4 was issued against © 2013 IBM Corporation

In the output, scroll forward until you locate **ACTIVE RBS**.

You need to correlate the TCB to a TVB.

Look for low storage address, below 16m line. In this example, choose **001780C0** which might be the TVB address because it is the smallest address.

Address **00007000** is in MVS storage and points to the MVT. You can tell by looking at the location in storage. You can see that the first two bytes are F1, which indicate the MVT.



Locating the address to the dump

- Cut the address **001780C0** and locate it in the dump
- View these findings:
  - The **F2** in the first two positions indicates this address is a **TVB**
  - A VOST was involved, **DSI#2082**

```
001780C0   F20201F8   001789F0   000F4F20   0095C098  | 2..8..i0..|..n{q
001780D0   00007000   00000000   40000000   00000000  | ........ .......
001780E0   40000000   00000000   00000000   00000000  |  ...............
001780F0   80008800   00050000   00000000   C3D5D4D7  | ..h.........CNMP
00178100   F5F4F9F1   C4E2C97B   F2F0F8F2   00000000  | 5491DSI#2082....
00178110   00000000   00000000   1ED9C000   00000000  | ..........R{.....
```

6 Finding the task the EC4 was issued against © 2013 IBM Corporation

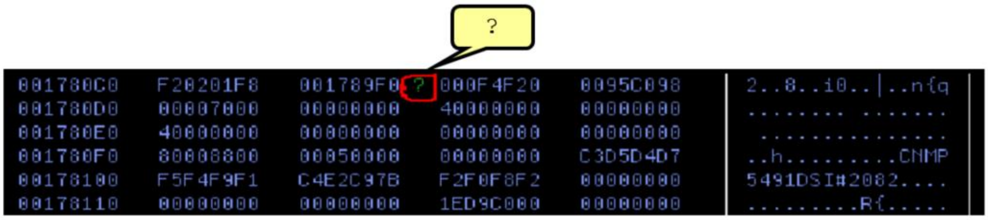
Cut the address **001780C0**, and locate it in the dump. The dump shows you two things.

First, the **F2** in the first two positions indicates this address is a **TVB**.

Second, in the right part of the image, you see that a VOST was involved, **DSI#2082**. The (**#**) in the name indicates that the item is a VOST.

Locating the LRCE chain

- Look for the LRCE chain to see what command the VOST issued
- To do this, start by going to the TVB +8 to get to the TIB



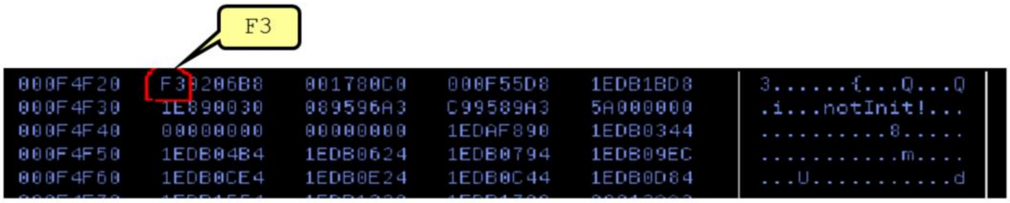
```
001780C0   F20201F8   001789F0 ? 000F4F20   0095C098   | 2..8..i0..|..n{q
001780D0   00007000   00000000   40000000   00000000   | ........ .......
001780E0   40000000   00000000   00000000   00000000   |  ...............
001780F0   80008800   00050000   00000000   C3D5D4D7   | ..h.........CNMP
00178100   F5F4F9F1   C4E2C97B   F2F0F8F2   00000000   | 5491DSI#2082....
00178110   00000000   00000000   1ED9C000   00000000   | ..........R{.....
```

Look for the LRCE chain to see what command the VOST issued.

To start the process, go to the TVB +8 to get to the TIB.

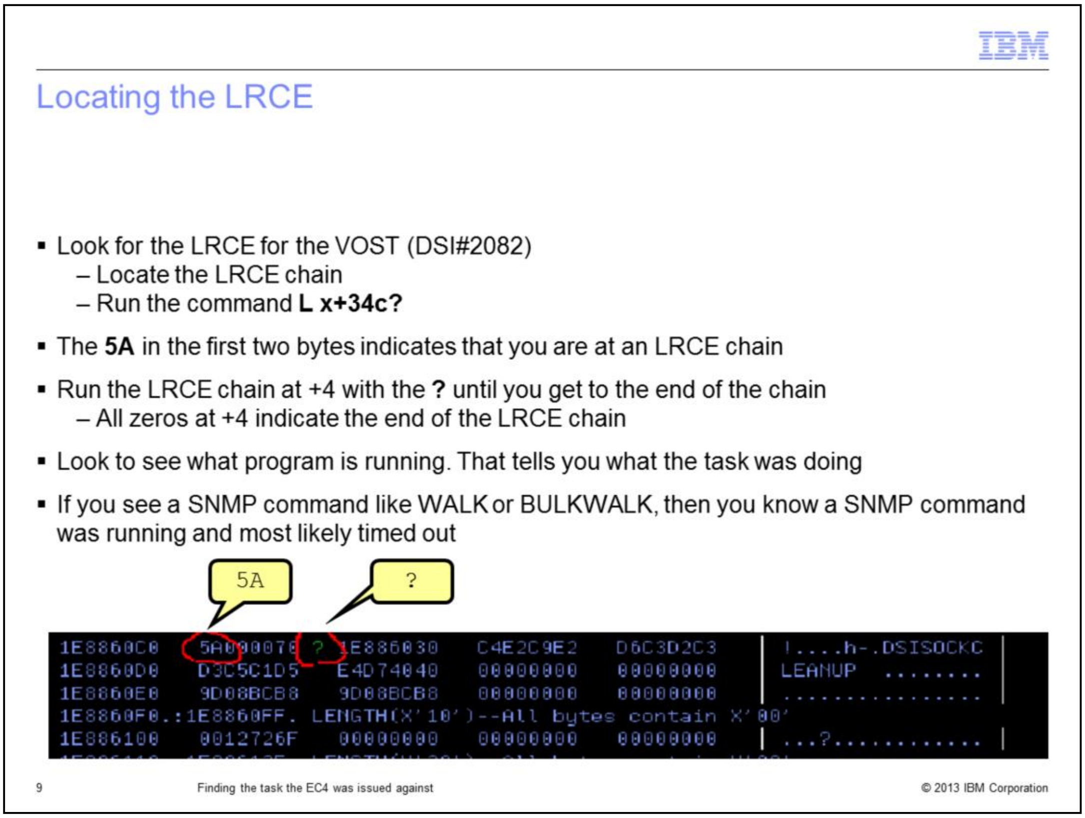
Viewing the TIB

F3 in the first two bytes indicates that you are looking at the TIB



```
000F4F20  F30206B8  001780C0  000F55D8  1EDB1BD8   3......{...Q...Q
000F4F30  1E890030  089596A3  C99589A3  5A000000   .i...notInit!...
000F4F40  00000000  00000000  1EDAF890  1EDB0344   ..........8.....
000F4F50  1EDB04B4  1EDB0624  1EDB0794  1EDB09EC   ...........m....
000F4F60  1EDB0CE4  1EDB0E24  1EDB0C44  1EDB0D84   ...U...........d
```

The F3 in the first two bytes indicates that you are looking at the TIB.



IBM

Locating the LRCE

- Look for the LRCE for the VOST (DSI#2082)
  - Locate the LRCE chain
  - Run the command **L x+34c?**
- The **5A** in the first two bytes indicates that you are at an LRCE chain
- Run the LRCE chain at +4 with the **?** until you get to the end of the chain
  - All zeros at +4 indicate the end of the LRCE chain
- Look to see what program is running. That tells you what the task was doing
- If you see a SNMP command like WALK or BULKWALK, then you know a SNMP command was running and most likely timed out

```
1E8860C0    5A000070 ? 1E886030    C4E2C9E2    D6C3D2C3   | !....h-.DSISOCKC |
1E8860D0    D3C5C1D5    E4D74040    00000000    00000000   | LEANUP  ........ |
1E8860E0    9D08BCB8    9D08BCB8    00000000    00000000   | ................ |
1E8860F0.:1E8860FF. LENGTH(X'10')--All bytes contain X'00'
1E886100    0012726F    00000000    00000000    00000000   | ...?............ |
```

9 Finding the task the EC4 was issued against © 2013 IBM Corporation

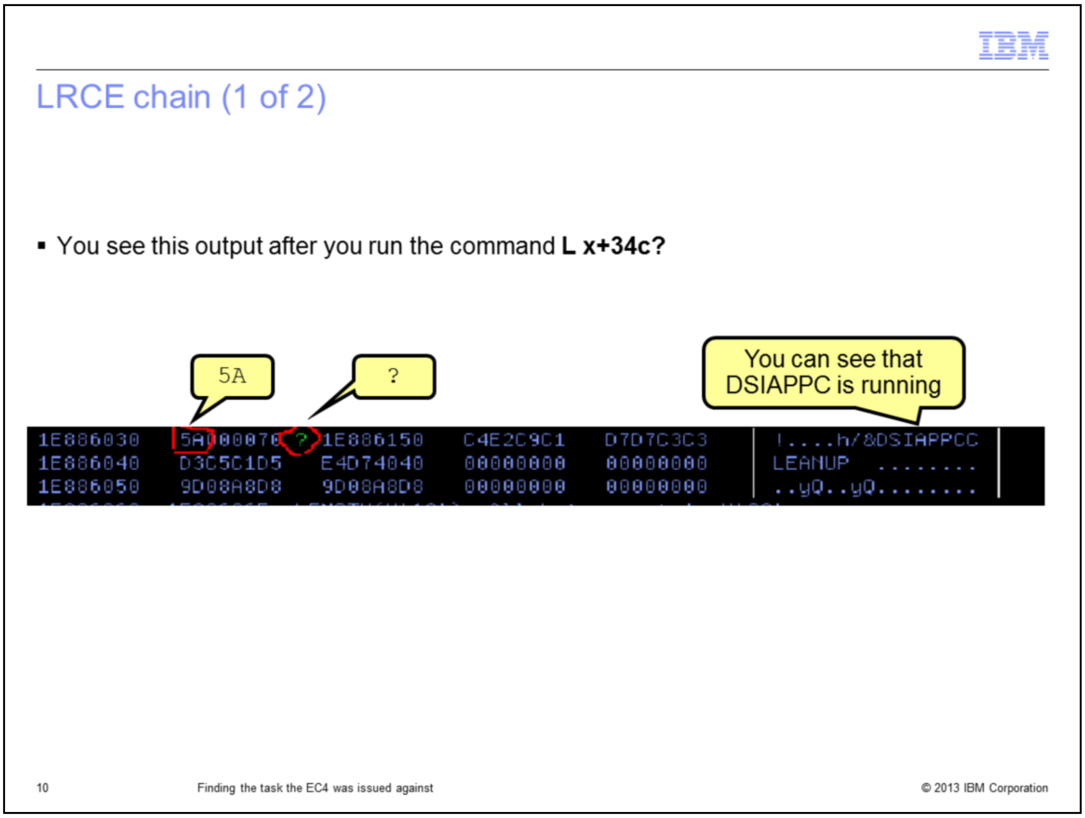
Now that you are at the TIB, you can start to look for the LRCE for the VOST (DSI#2082). To get to the LRCE chain, run the command **L x+34c?**.

You know that you are at an LRCE chain because of the **5A** in the first two bytes.

Next, you run the LRCE chain at +4 with the **?** until you get to the end of the chain. When you see all zeros at +4, you know that you are at the end of the LRCE chain.

Look to see what program is running, which tells you what the task was doing. If you see an SNMP command like WALK or BULKWALK, then you know that an SNMP command was running, and most likely timed out.

The next few slides take you through the process of searching for all zeros at +4.



LRCE chain (1 of 2)

- You see this output after you run the command **L x+34c?**

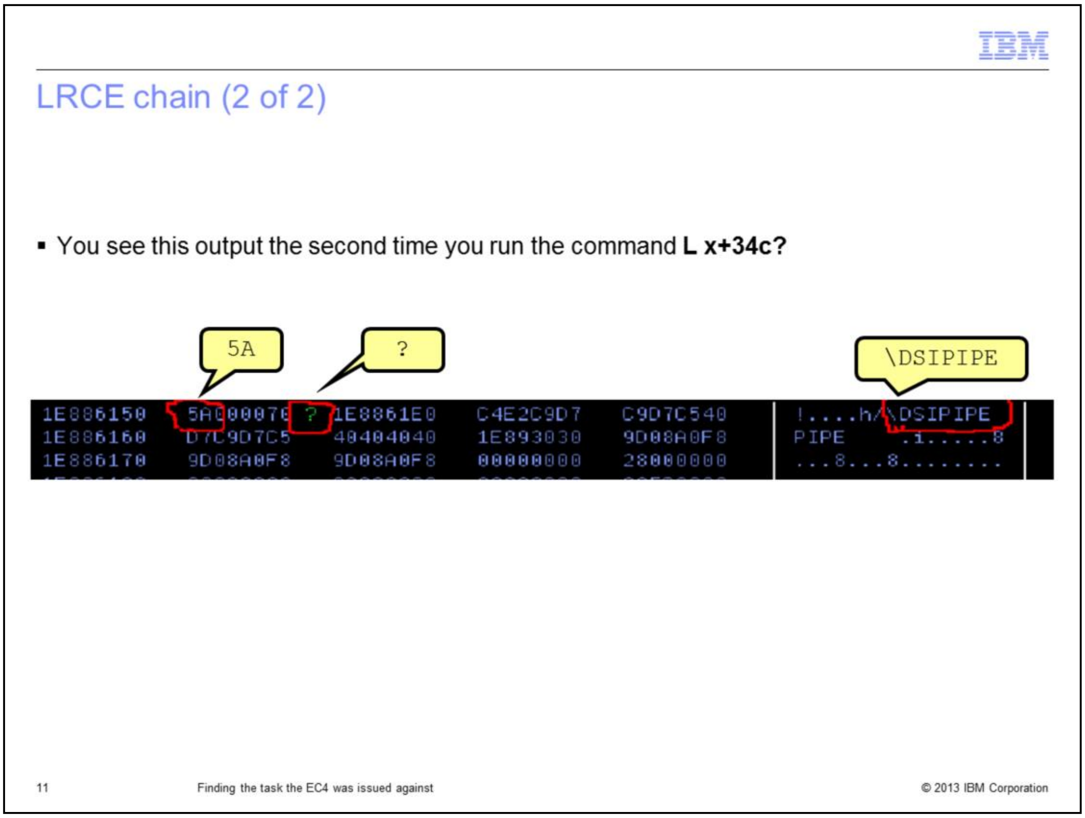
5A

?

You can see that DSIAPPC is running

```
1E886030    5A000070 ?1E886150    C4E2C9C1    D7D7C3C3    | !....h/&DSIAPPCC |
1E886040    D3C5C1D5    E4D74040    00000000    00000000    | LEANUP  ........ |
1E886050    9D08A8D8    9D08A8D8    00000000    00000000    | ..yQ..yQ........ |
```

After you run the command once, you generate the output that is shown here. You can see that you are still in an LRCE chain because the first two bytes are **5A**, and not at the end of the chain yet because **+4** is not zeros. You can see that **DSIAPPC** is running. Run the command again and keep looking.



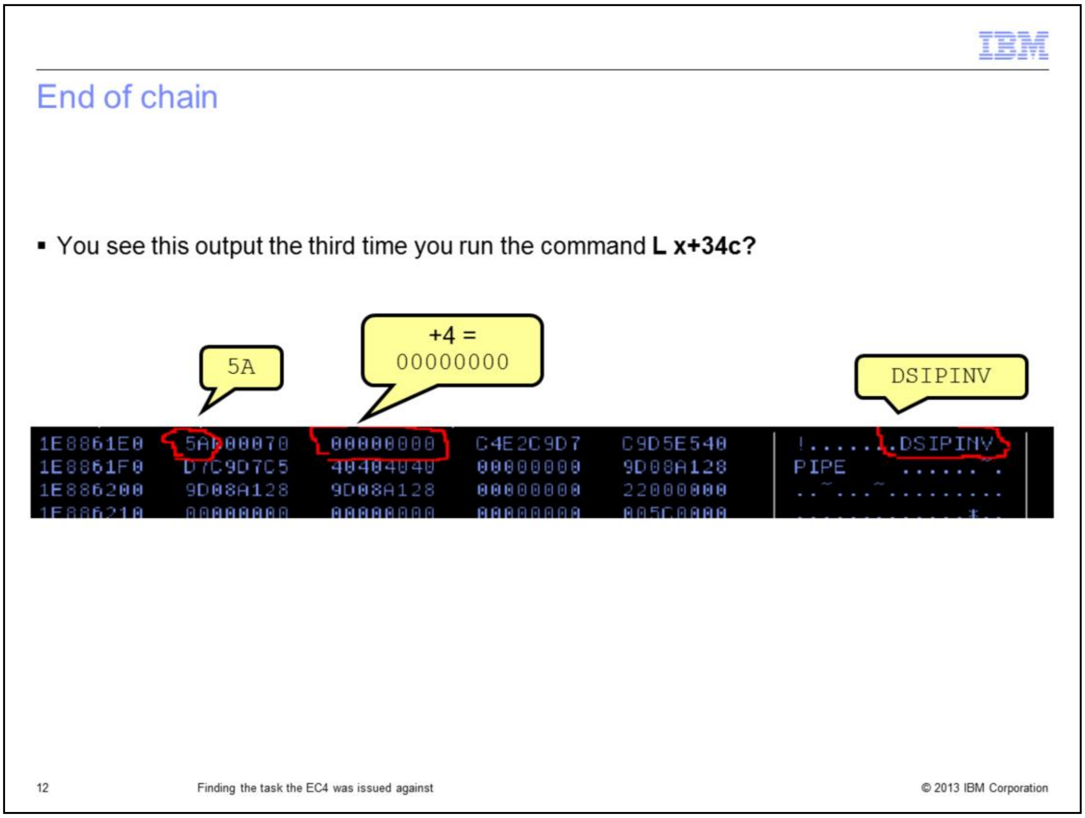
LRCE chain (2 of 2)

- You see this output the second time you run the command **L x+34c?**

```
1E886150   5A000070 ?  1E8861E0   C4E2C9D7   C9D7C540    !....h/\DSIPIPE
1E886160   D7C9D7C5    40404040   1E893030   9D08A0F8    PIPE    .i.....8
1E886170   9D08A0F8    9D08A0F8   00000000   28000000    ...8...8........
```

11   Finding the task the EC4 was issued against   © 2013 IBM Corporation

After you run the command a second time, you can see that first two bytes are **5A** and +4 is not zeros. You know that you are still in the LRCE chain. There is still more in the chain. You can see that DSIPIPE is running, which means the task was running a PIPE.



End of chain

- You see this output the third time you run the command **L x+34c?**

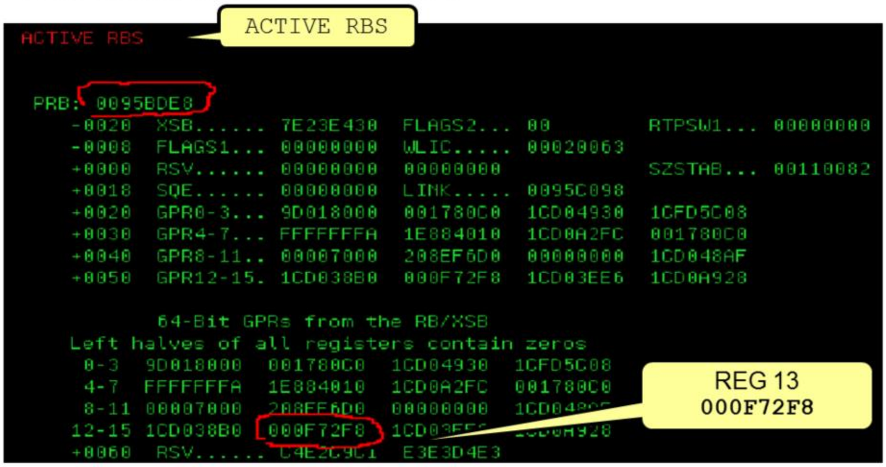
```
1E8861E0   5A000070   00000000   C4E2C9D7   C9D5E540   | !.......DSIPINV |
1E8861F0   D7C9D7C5   40404040   00000000   9D08A128   | PIPE    ....... |
1E886200   9D08A128   9D08A128   00000000   22000000   | ..~...~......... |
1E886210   00000000   00000000   00000000   005C0000   | .............*.. |
```

When the third command is completed, you can see you are still in the LRCE chain because the first two bytes are **5A**. Because +4 is all zeros, you know that you are at the end of the chain. You can see DSIPINV is running.

Because you are at the end of the chain, you need to look at the save area pointed to by REG 13 in the TCB with IP VERBX CNMIPCS.

Getting the register address

- Get the register **13** address from the TCB
  - Locate the TCB display from the top of this document
  - Find the **ACTIVE RBS**
  - Get the address from register **13**
- In the example, the address is **000F72F8**

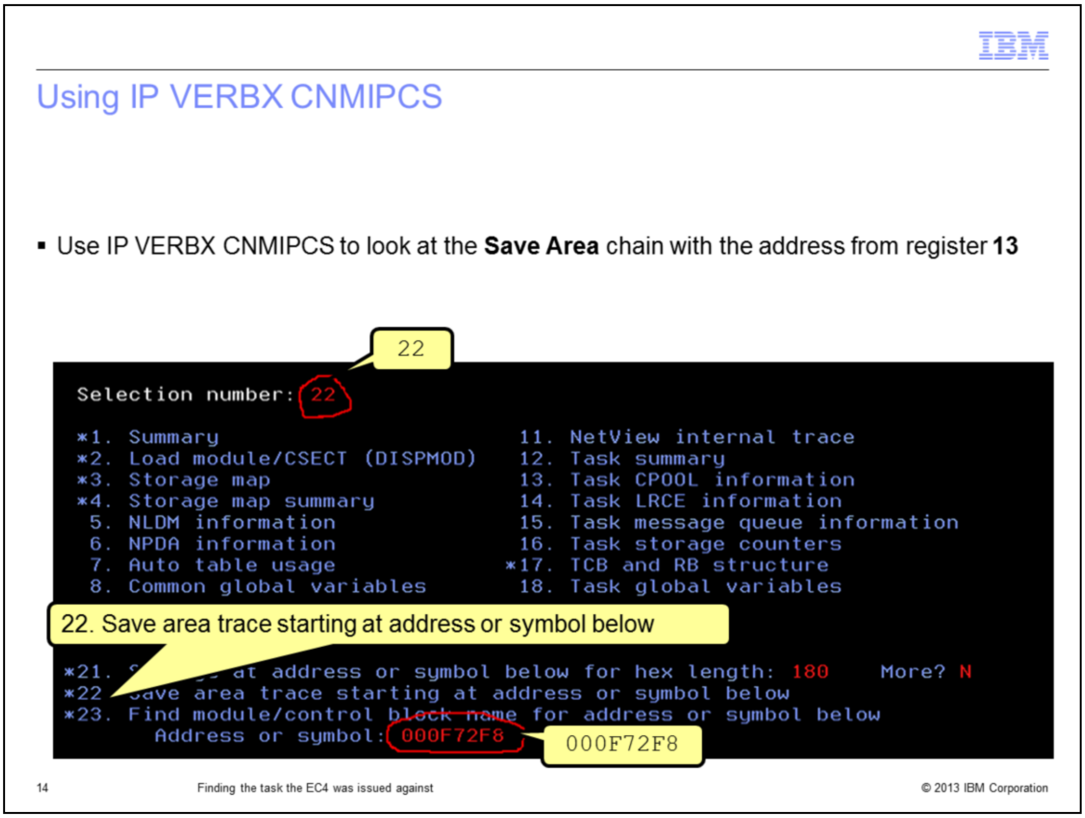


```
ACTIVE RBS

  PRB: 0095BDE8
     -0020  XSB...... 7E23E430  FLAGS2... 00        RTPSW1... 00000000
     -0008  FLAGS1... 00000000  WLIC..... 00020063
     +0000  RSV...... 00000000  00000000            SZSTAB... 00110082
     +0018  SQE...... 00000000  LINK..... 0095C098
     +0020  GPR0-3... 9D018000  001780C0  1CD04930  1CFD5C08
     +0030  GPR4-7... FFFFFFFA  1E884010  1CD0A2FC  001780C0
     +0040  GPR8-11.. 00007000  208EF6D0  00000000  1CD048AF
     +0050  GPR12-15. 1CD038B0  000F72F8  1CD03EE6  1CD0A928

            64-Bit GPRs from the RB/XSB
     Left halves of all registers contain zeros
      0-3   9D018000  001780C0  1CD04930  1CFD5C08
      4-7   FFFFFFFA  1E884010  1CD0A2FC  001780C0
      8-11  00007000  208EF6D0  00000000  1CD048AF
     12-15  1CD038B0  000F72F8  1CD03EE6  1CD0A928
     +0060  RSV...... C4E2C9C1  E3E3D4E3
```

13 Finding the task the EC4 was issued against © 2013 IBM Corporation

To get the register 13 address from the TCB, scroll up the TCB display from the top of this document and find the **ACTIVE RBS** as shown near the upper part of the image.

In the lower part of the image, you can see the row with registers 12 through 15. Identify the address from register 13. In this example, it is **000F72F8**.



Using IP VERBX CNMIPCS

- Use IP VERBX CNMIPCS to look at the **Save Area** chain with the address from register **13**

22

```
Selection number: 22

*1.  Summary                          11.  NetView internal trace
*2.  Load module/CSECT (DISPMOD)      12.  Task summary
*3.  Storage map                      13.  Task CPOOL information
*4.  Storage map summary              14.  Task LRCE information
 5.  NLDM information                 15.  Task message queue information
 6.  NPDA information                 16.  Task storage counters
 7.  Auto table usage                *17.  TCB and RB structure
 8.  Common global variables          18.  Task global variables

*21. S          at address or symbol below for hex length: 180    More? N
*22   ave area trace starting at address or symbol below
*23. Find module/control block name for address or symbol below
        Address or symbol: 000F72F8
```

22. Save area trace starting at address or symbol below

000F72F8

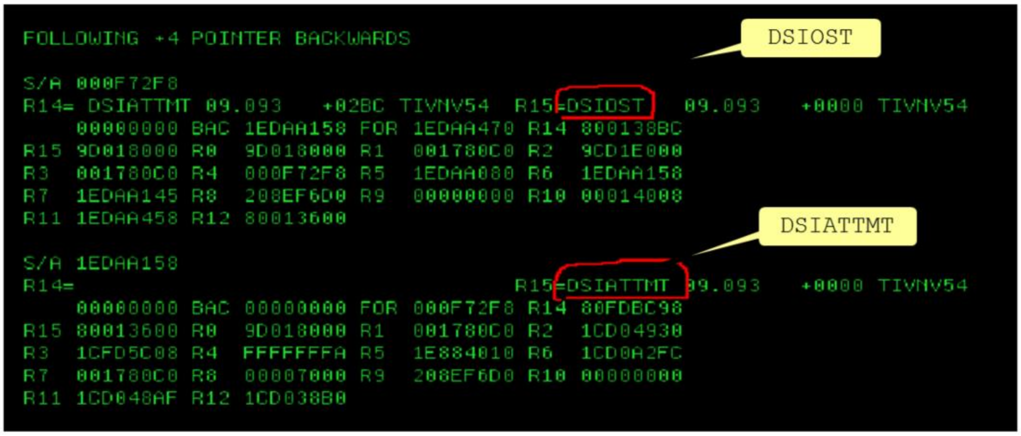
14 Finding the task the EC4 was issued against © 2013 IBM Corporation

Use IP VERBX CNMIPCS to look at the **Save Area** chain with the address from register 13.

See the selection **22. Save area trace starting at address or symbol below** and the address selected, **000F72F8**.

Save Area chain

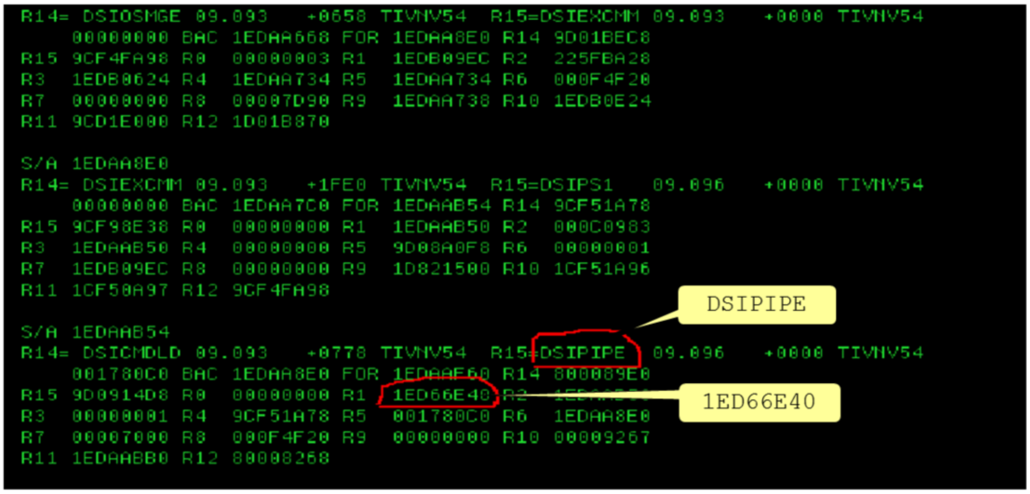
- Save Area chain



```
FOLLOWING +4 POINTER BACKWARDS

S/A 000F72F8
R14= DSIATTMT 09.093    +02BC TIVNV54  R15=DSIOST   09.093    +0000 TIVNV54
    00000000 BAC 1EDAA158 FOR 1EDAA470 R14 800138BC
R15 9D018000 R0  9D018000 R1  001780C0 R2  9CD1E000
R3  001780C0 R4  000F72F8 R5  1EDAA080 R6  1EDAA158
R7  1EDAA145 R8  208EF6D0 R9  00000000 R10 00014008
R11 1EDAA458 R12 80013600

S/A 1EDAA158
R14=                                   R15=DSIATTMT 09.093    +0000 TIVNV54
    00000000 BAC 00000000 FOR 000F72F8 R14 80FDBC98
R15 80013600 R0  9D018000 R1  001780C0 R2  1CD04930
R3  1CFD5C08 R4  FFFFFFFA R5  1E884010 R6  1CD0A2FC
R7  001780C0 R8  00007000 R9  208EF6D0 R10 00000000
R11 1CD048AF R12 1CD038B0
```

Here you see the Save Area chain. Look at the program names until you see a PIPE running. You look for a PIPE because DSIPIPE is in the LRCE chain. Scroll through the list.

PIPE



```
R14= DSIOSMGE 09.093   +0658 TIVNV54  R15=DSIEXCMM 09.093   +0000 TIVNV54
     00000000 BAC 1EDAA668 FOR 1EDAA8E0 R14 9D01BEC8
R15 9CF4FA98 R0  00000003 R1  1EDB09EC R2  225FBA28
R3  1EDB0624 R4  1EDAA734 R5  1EDAA734 R6  000F4F20
R7  00000000 R8  00007D90 R9  1EDAA738 R10 1EDB0E24
R11 9CD1E000 R12 1D01B870

S/A 1EDAA8E0
R14= DSIEXCMM 09.093   +1FE0 TIVNV54  R15=DSIPS1   09.096   +0000 TIVNV54
     00000000 BAC 1EDAA7C0 FOR 1EDAAB54 R14 9CF51A78
R15 9CF98E38 R0  00000000 R1  1EDAAB50 R2  000C0983
R3  1EDAAB50 R4  00000000 R5  9D08A0F8 R6  00000001
R7  1EDB09EC R8  00000000 R9  1D821500 R10 1CF51A96
R11 1CF50A97 R12 9CF4FA98

S/A 1EDAAB54
R14= DSICMDLD 09.093   +0778 TIVNV54  R15=DSIPIPE  09.096   +0000 TIVNV54
     001780C0 BAC 1EDAA8E0 FOR 1EDAAF60 R14 800089E0
R15 9D0914D8 R0  00000000 R1  1ED66E40 R2  1EDAA[illegible]
R3  00000001 R4  9CF51A78 R5  001780C0 R6  1EDAA8E0
R7  00007000 R8  000F4F20 R9  00000000 R10 00009267
R11 1EDAABB0 R12 80008268
```

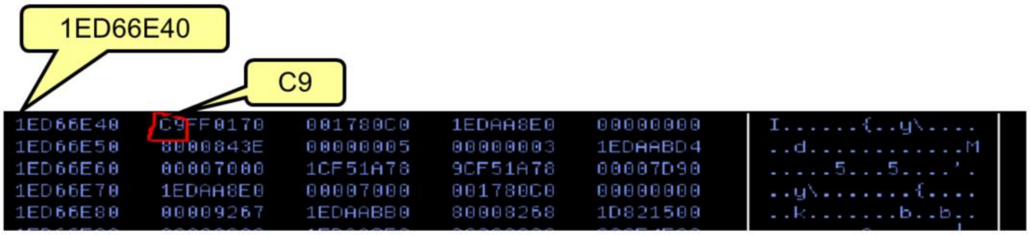
DSIPIPE

1ED66E40

Finally, you can see the PIPE in the output and can identify the register 1ED66E40.

Taking the address from register one

- Take the address from register 1, which is 1ED66E40 and go there in the dump
- C9 indicates the **Command Work Block**

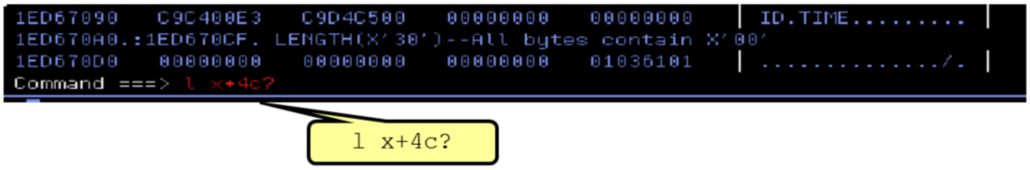


```
1ED66E40   C9FF0170   001780C0   1EDAA8E0   00000000   I......{..y\....
1ED66E50   8000843E   00000005   00000003   1EDAABD4   ..d............M
1ED66E60   00007000   1CF51A78   9CF51A78   00007D90   .....5...5....'.
1ED66E70   1EDAA8E0   00007000   001780C0   00000000   ..y\.......{....
1ED66E80   00009267   1EDAABB0   80008268   1D821500   ..k.......b..b..
```

Take the address from register 1. In the example, it is 1ED66E40. Go there in the dump. The C9 in register 2 indicates that you are at the *Command Work Block* (CWB).

Viewing the command buffer

- To see the **Command Buffer**, you must go to **4c** from this address
- Run the command **l x+4c?**



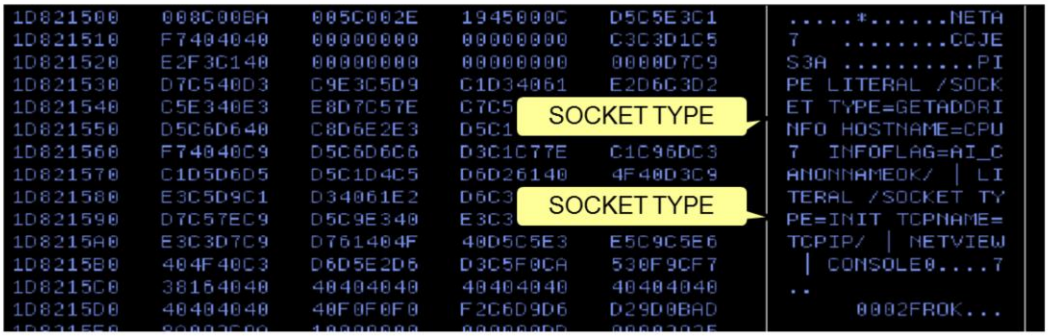
```
1ED67090   C9C400E3   C9D4C500   00000000   00000000  | ID.TIME......... |
1ED670A0.:1ED670CF. LENGTH(X'30')--All bytes contain X'00'
1ED670D0   00000000   00000000   00000000   01036101  | .............../. |
Command ===> l x+4c?
```

l x+4c?

To see the command buffer, you want to go to **4c** from this address. Run the command **l x+4c?** to see what command was running.

Socket command

- The command that was running when the EC4 occurred displays
- The command is a Socket command
- The EYECATHER shows the you which task or resource did not terminate within one minute, which caused NetView to issue the valid EC4 abend



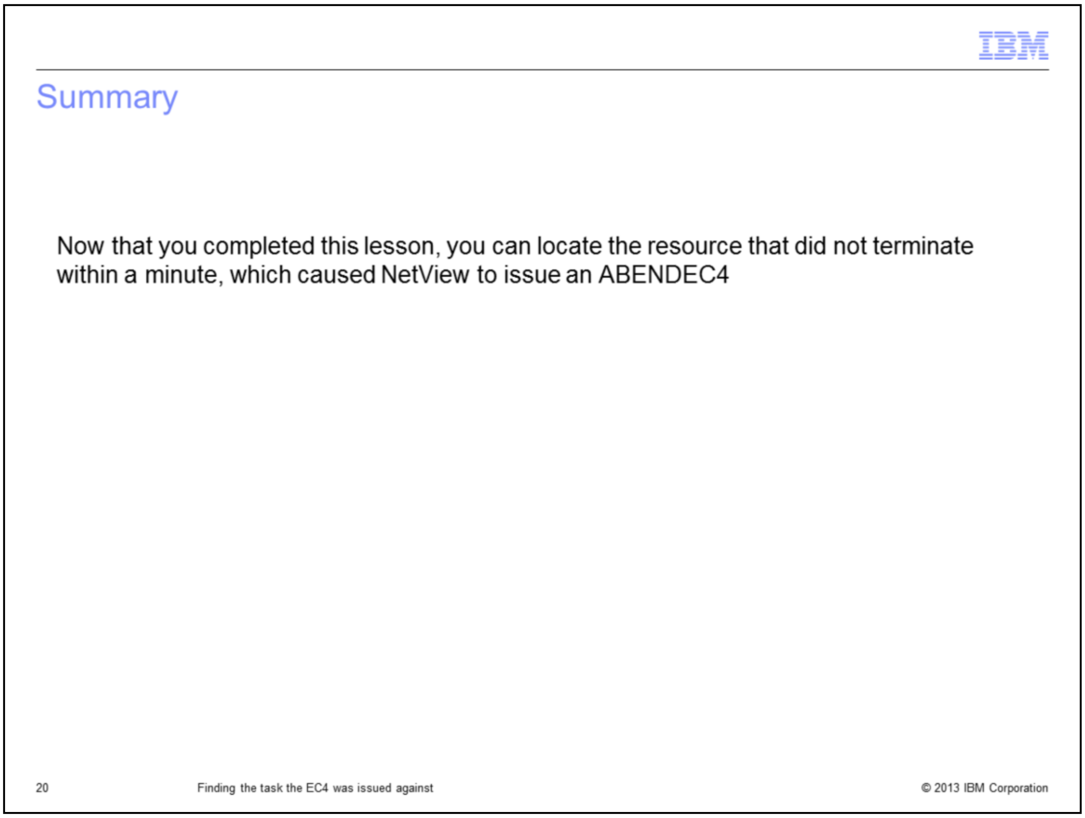
```
1D821500    008C00BA    005C002E    1945000C    D5C5E3C1    .....*......NETA
1D821510    F7404040    00000000    00000000    C3C3D1C5    7   ........CCJE
1D821520    E2F3C140    00000000    00000000    0000D7C9    S3A .........PI
1D821530    D7C540D3    C9E3C5D9    C1D34061    E2D6C3D2    PE LITERAL /SOCK
1D821540    C5E340E3    E8D7C57E    C7C5                    ET TYPE=GETADDRI
1D821550    D5C6D640    C8D6E2E3    D5C1                    NFO HOSTNAME=CPU
1D821560    F74040C9    D5C6D6C6    D3C1C77E    C1C96DC3    7  INFOFLAG=AI_C
1D821570    C1D5D6D5    D5C1D4C5    D6D26140    4F40D3C9    ANONNAMEOK/ | LI
1D821580    E3C5D9C1    D34061E2    D6C3                    TERAL /SOCKET TY
1D821590    D7C57EC9    D5C9E340    E3C3                    PE=INIT TCPNAME=
1D8215A0    E3C3D7C9    D761404F    40D5C5E3    E5C9C5E6    TCPIP/ | NETVIEW
1D8215B0    404F40C3    D6D5E2D6    D3C5F0CA    530F9CF7     | CONSOLE0....7
1D8215C0    38164040    40404040    40404040    40404040    ..
1D8215D0    40404040    40F0F0F0    F2C6D9D6    D29D0BAD         0002FROK...
```

SOCKET TYPE

SOCKET TYPE

This image shows the command buffer. You can see the command that was running when the EC4 occurred. It is a Socket command.

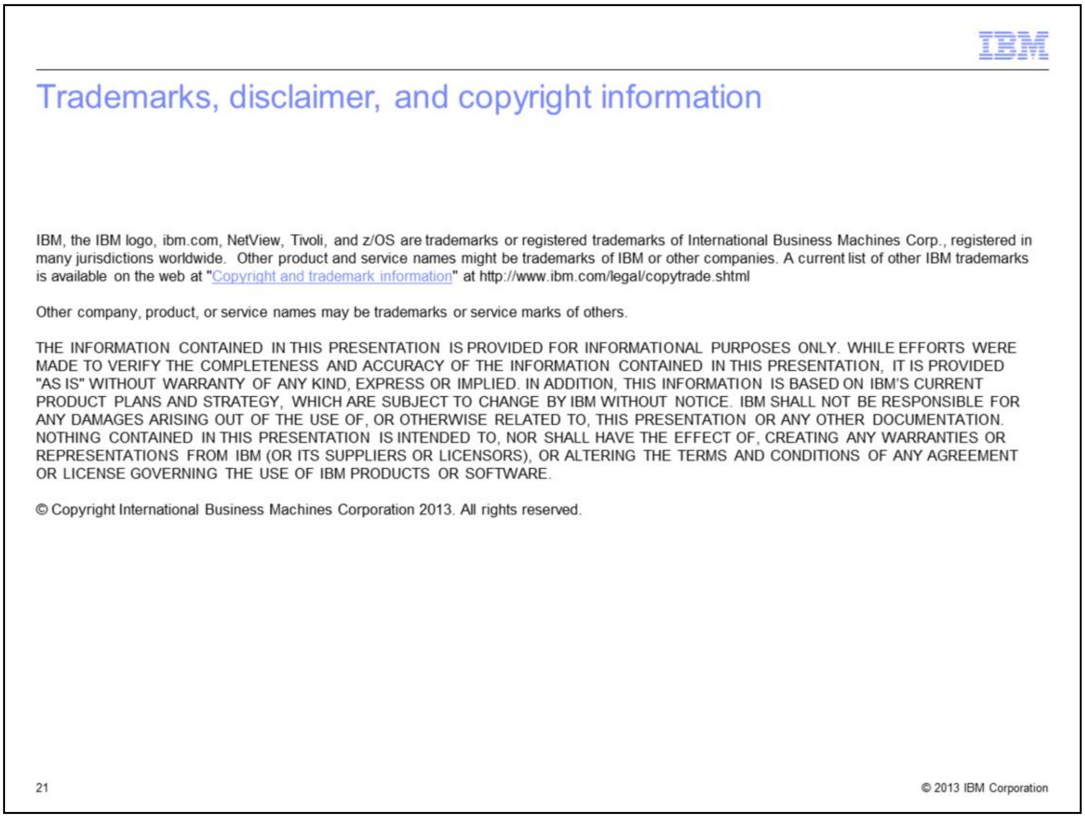
The EYECATHER shows you which task or resource did not terminate within one minute. The failure to terminate caused NetView to issue the valid EC4 abend.



Summary

Now that you completed this lesson, you can locate the resource that did not terminate within a minute, which caused NetView to issue an ABENDEC4

Now that you completed this lesson, you can locate the resource that did not terminate within a minute, which caused NetView to issue an ABENDEC4.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.