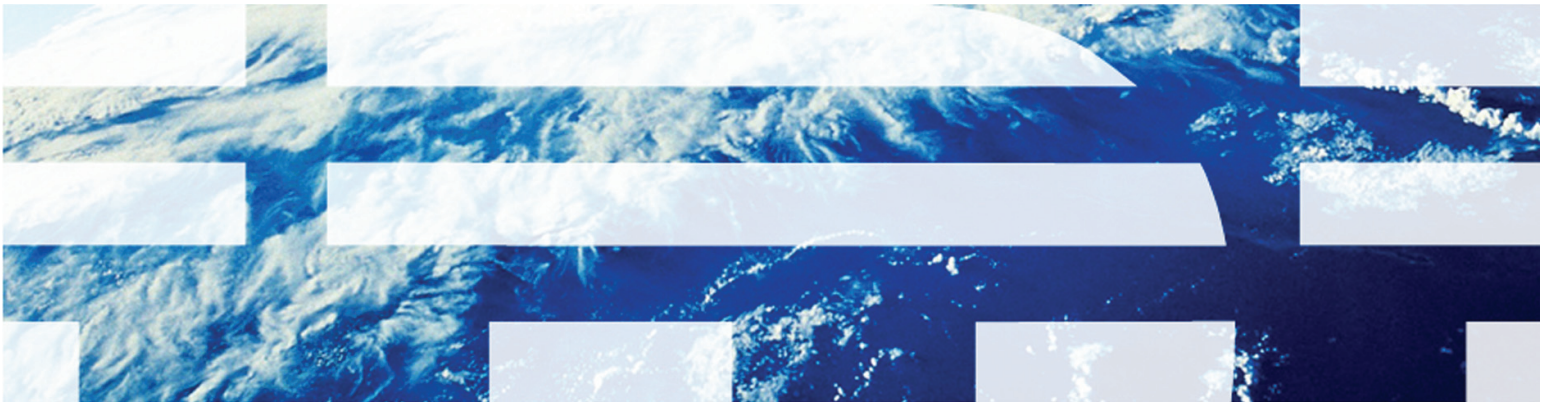


IBM Tivoli NetView for z/OS 6.1: Automation Techniques

Student Exercises

Course: TZ213 ERC: 1.0

August 2011

Printed in Ireland

Table of contents

Student exercises for Unit 1

No student exercises are provided for this unit. 1-1

Student exercises for Unit 2

Lab exercise overview 2-1
Data sets used in This lab 2-1
Exercise 2-1: Message Revision Table (MRT) 2-1
Lab exercise objectives 2-1
Lab exercise instructions 2-2
Lab exercise overview 2-2
Payroll start messages 2-3
Payroll stop messages 2-3
Payroll cancel messages 2-4
Additional messages issued by payroll 2-4
Connecting to TSO 2-5
Connecting to NetView (AOFDA) 2-5
Testing your MRT 2-6
Answers to Lab 2 Exercise 1 questions 2-9
Example MRT (VAPMRT1) 2-10
Exercise 2-2: ASSIGN Command 2-11
Lab exercise objectives 2-11
Lab exercise instructions 2-11
Answers to Lab 2 Exercise 2 questions 2-15
Exercise 2-3: NetView and z/OS consoles 2-16
Lab exercise objectives 2-16
Lab exercise instructions 2-16
Answers to Lab 2 Exercise 3 questions 2-19

Student exercises for Unit 3

Lab exercise overview 3-1
Exercise 3-2: Routing commands to other tasks and domains 3-1
Lab exercise objectives 3-1
Lab exercise instructions 3-1
Answers to Lab 3 Exercise 2 questions 3-3
Exercise 3-3: Timer commands 3-4
Lab exercise objectives 3-4
Lab exercise instructions 3-4
Answers to Lab 3 Exercise 3 questions 3-8
Exercise 3-4: Command Revision Table (CRT) 3-9
Lab exercise objectives 3-9
Lab exercise instructions 3-9
Answers to Lab 3 Exercise 4 questions 3-11

Student exercises for Unit 4

Lab exercise overview 4-1
Exercise 4-1: Managing the NetView automation tables 4-1

Lab exercise objectives 4-1
Lab exercise instructions 4-1
Answers to Lab 4 Exercise 1 questions 4-4

Student exercises for Unit 5

Lab exercise overview 5-1
Data sets used in this lab 5-1
Exercise 5-1: Coding automation table statements 5-1
Lab exercise objectives 5-1
Lab exercise instructions 5-1
Lab exercise instructions 5-2
Lab exercise instructions 5-4
Answers to Lab 5 Exercise 1 Questions 5-6

Student exercises for Unit 1

No student exercises are provided for this unit.

Student exercises for Unit 2

Lab exercise overview

The z/OS console is a major source of messages for automation. The messages can be modified or suppressed before they route to NetView for automation. The messages can also be suppressed from automation. In this exercise, you create message revision table (MRT) statements for processing messages that route to the MRT. During the lab exercise, you use the two following PCOMM sessions:

- NetView
- TSO

Data sets used in This lab

NV390.WORKSHOP.DSIPARM _____

Exercise 2-1: Message Revision Table (MRT)

Lab exercise objectives

This lab exercise uses the NetView Message Revision Table (MRT) to suppress and modify messages that are to route to the z/OS system console. At the end of the lab, you should be able to perform the following tasks:

- Define MRT statements to suppress messages.
- Define MRT statements to modify messages.
- Activate an MRT.

Lab exercise instructions

Use two PCOMM 3270 emulator sessions as follows:

- A TSO session for creating a Message Revision Table (MRT) member in DSIPARM. If the instructions refer to a panel ID, you can find it in the upper left corner of the panel
- A NetView session for activating the MRT and issuing commands that generate the messages for the MRT. Use the Canzlog data space to verify messages on the system console. If you want to access the system console, ask your instructor.

Note: Messages that are suppressed by MRT will show up in canzlog, by selecting the message and pressing Enter you see if message has been suppressed:

```
CNMKCZMD OUTPUT FOR $HASP395 Time: 07/13/11 05:27:44.523
CzID: 76807 00012C07x     AutoTime: 5 msec           DomTime: none
JobName: PAYROLL          DestConsole:               AutoToken:
Tags: MVS
Flags: MRT, Suppr, Auth
CHkey: PAYROLL           SystemID: MVSA            JobID: STC00578
SmsgID: 52001F4Fx         ASID: 003Dx                DomToken:
00000000x
AStype: S                AuthUser: SYSPROG         AuthGroup: SYS1
Mtype: E (C5x)

DescCodes: 0400 (6)
RouteCodes: 40000000000000000000000000000000
            (2)

$HASP395 PAYROLL  ENDED
```

Lab exercise overview

Use the TSO session to edit VAPMRT1 and create MRT statements for suppressing and revising (modifying) messages. For this lab exercise, you are the system programmer who is responsible for the *payroll* application. When the payroll application starts and stops, several messages are generated. Your job is to suppress some of the messages pertaining to the *payroll* application and revise other messages. You create an MRT to accomplish those tasks.

First, you need to know the messages that are issued for each action and also a description of how to process each message when it runs through the MRT. Some of the messages route to the system log (syslog) only. You code MRT statements to process the syslog messages and messages that go the system console.

Payroll start messages

When the payroll application starts, the following messages are displayed at the z/OS system console:

```
S RUNVAPL,JOBNAME=PAYROLL,NAME=VAPL20
$HASP100 PAYROLL  ON STCINRDR
IEF695I START RUNVAPL  WITH JOBNAME PAYROLL  IS ASSIGNED TO USER
SYSPROG
, GROUP SYS1
$HASP373 PAYROLL  STARTED
IEF403I PAYROLL - STARTED - TIME=02.11.00
@0023 VAPL20000A REPLY WARM OR COLD
R 23 SUPPRESSED
IEE600I REPLY TO 0023 IS;SUPPRESSED
+VAPL20010I VAPL20 WARM START COMPLETE
@0024 VAPL20999A REPLY END TO STOP
```

The VAPL29999A message indicates the payroll application is active.

During the lab exercise, you code MRT statements to accomplish the following tasks:

- Suppress all $HASP100, and $HASP373 messages.
- Do nothing to any IEF695I message.
- Revise IEF403I to send it to NetView for automation only.

  IEF403I is required by SA z/OS. You need to pass it to NetView.

Payroll stop messages

When the payroll application stops by responding to its outstanding WTOR, the following messages are displayed at the z/OS system console:

```
R 24 SUPPRESSED
IEE600I REPLY TO 0024 IS;SUPPRESSED
+VAPL20020I VAPL20 SHUTDOWN COMPLETE
-                                              --TIMINGS (MINS.)--
----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-PAYROLL          STEP1     3408      6      0 551177    .00     .4
637   0     0     0     0     0
IEF404I PAYROLL - ENDED - TIME=02.11.26
-PAYROLL  ENDED.  NAME-                  TOTAL TCB CPU TIME=   .00
TOTAL ELAPSED TIME=    .4
$HASP395 PAYROLL  ENDED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0040.
$HASP250 PAYROLL PURGED -- (JOB KEY WAS C6DA9EB2)
```

During the lab exercise, code MRT statements to perform the following tasks:

- Suppress all $HASP395 messages.
- Revise IEF404I to send it to NetView for automation only.

  IEF404I is required by SA z/OS. You need to pass it to NetView.
- Delete all IEA989I messages.
- Delete all IEE600I messages.
- Revise all $HASP250 messages so that they show as red and blinking

Payroll cancel messages

When the payroll application is canceled, the following messages appear at the z/OS system console:

```
C PAYROLL
IEA989I SLIP TRAP ID=X222 MATCHED.  JOBNAME=PAYROLL , ASID=0040.
IEE400I THESE MESSAGES CANCELLED - 0029.
IEF450I PAYROLL PAYROLL - ABEND=S222 U0000 REASON=00000000 550
        TIME=02.27.54
-                                          --TIMINGS (MINS.)--
        ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-PAYROLL          STEP1    *S222      6      0 551177    .00     .3
588   0     0     0     0     0
-PAYROLL  ENDED.  NAME-                  TOTAL TCB CPU TIME=   .00
TOTAL ELAPSED TIME=    .3
$HASP395 PAYROLL  ENDED
IEE301I PAYROLL           CANCEL COMMAND ACCEPTED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0040.
```

During the lab exercise, code MRT statements to perform the following tasks:

- Suppress all IEE400I and $HASP395 messages.
- Revise any IEF450I message to be displayed in yellow if issued for the payroll application. Otherwise, change the color to white for all other applications.
- Delete all IEE301I messages.

Additional messages issued by payroll

The payroll application issues messages that have a prefix of VAPL29. Examples are as follows:

- VAPL29999A indicates that the payroll application is active.
- VAPL29020I indicates that the payroll application stopped.

During the lab exercise, code MRT statements to perform the following tasks:

- Use an UPON(ALWAYS) condition to display all payroll messages in reverse video blue by using one of these methods:
  - Test for a jobname of PAYROLL for a length of eight characters.
  - Test a substring of the message ID for all VAPL29 messages. An example is as follows:

    ```
    (MSGID LEFT 6 = 'VAPL29')
    ```

Connecting to TSO

Perform the following steps:

___ 1. Access your system and log on to TSO, following the instructions that the instructor provides. Edit the VAPMRT1 member in NV390.WORKSHOP.DSIPARM.

Initially, VAPMRT1 has one line in it as follows:

```
* Empty MRT - ready for your input
```

Otherwise, ask your instructor for assistance.

___ 2. You can now begin to code your MRT statements in VAPMRT1. Use UPON, SELECT-WHEN, REVISE, NETVONLY, and so on.

Refer to the discussion of the messages that are issued when the payroll application is started, stopped, and canceled for properly coding your MRT statements.

Tip: For additional help, browse the sample MRT supplied by NetView, CNMSMRT1, to help you code your MRT statements.

___ 3. Press PF3 to save the MRT when you are ready to test. Do not end your session with TSO,. in case you need to make further changes to the VAPMRT1 member.

___ 4. Use the Canzlog data space to verify that your MRT logged messages appropriately.

You are now finished creating the MRT statements and need to load the MRT from NetView.

Connecting to NetView (AOFDA)

___ 5. Access your system and log on to NetView domain AOFDA following the instructions provided by your instructor.

Testing your MRT

___ 6. At this point, you are ready to test the MRT you defined. Type “REVISE STATUS” and verify no MRT loaded. You should see:

```
REVISE STATUS
DSI231I No 'revision MSG table' is active
DSI231I No 'revision CMD table' is active
```

If you do not see the DSI231I message, ask your instructor for assistance.

___ 7. Without an active MRT, start the payroll application by issuing the command:

```
PAYROLL START
```

Browse the canzlog and look for the messages pertaining to the start of the payroll application. Also, you may want to look at the system console. Which messages were issued? Note that suppressed messages show up in canzlog, but if you look at the message details it will say suppressed by MRT.

_____

_____

Note: The PAYROLL command is a REXX routine that is written for these lab exercises to simplify the start and stop of the payroll application.

___ 8. Stop the payroll application by issuing the following command:

```
PAYROLL STOP
```

Browse the canzlog and look for the messages pertaining to the stop of the payroll application. Also, look at the emulator session for your system console. Which messages were issued? _____

_____

___ 9. Restart the payroll application. When it becomes active, cancel it by issuing the following command:

```
PAYROLL CANCEL
```

Browse the canzlog and look for the messages pertaining to the cancelling of the payroll application. Also, look at the emulator session for your system console. Which messages were issued? _____

_____

___ 10. Test the MRT that you created. Issue the following command:

```
REVISE MSG,MEMBER=VAPMRT1,TEST
```

The CNM501I message is as follows:

```
CNM501I TEST OF NETVIEW AUTOMATION FILE "VAPMRT1" WAS SUCCESSFUL
```


If you have errors, switch to your TSO session to correct the errors and reissue the REVISE MSG command to test your corrections.

Tip: If you are not sure why you have an error, ask your instructor for assistance.

___ 11. When your testing of the MRT yields no errors, activate it by entering the following command:

```
REVISE MSG,MEMBER=VAPMRT1
```

The following BNH608I message is displayed:

```
BNH608I 'LOAD MSG VAPMRT1' REQUEST COMPLETED SUCCESSFULLY
```

___ 12. Your MRT is now active. Query the status of the MRT by issuing a REVISE MSG,STATUS command. A message similar to the following one is displayed:

```
CNM012I MSG Revision table VAPMRT1, loaded by TSCCW01, has
examined
 0 objects since being loaded, 11/09/10 05:03:19
```

This message indicates that the VAPMRT1 Message Revision Table is active but no messages run through the MRT yet.

Test your MRT. Perform all of the following steps before making any changes to your MRT.

Tip: If the expected results do not occur, you can issue the REVISE MSG,REPORT command to determine if the MRT statements were driven and matched.

___ 13. Start the payroll application with a PAYROLL START command. Verify that your MRT performed the following tasks:

- Suppressed the $HASP100 and $HASP373 messages. These messages should still appear in syslog.
- Took no action with the IEF695I message.
- Revised the IEF403I message to send it to NetView for automation only. This message is to be suppressed and not logged to syslog.

___ 14. Stop the payroll application with a PAYROLL STOP command. Verify that your MRT performed the following tasks:

- Suppressed the $HASP395 messages. This message is to still be displayed in syslog.
- Revised the IEF404I to send it to NetView for automation only. This message is to be suppressed and not logged to syslog.
- Deleted the IEA989I message. This message is be suppressed and not logged to syslog.to
- Deleted the IEE600I message.
- Revised the $HASP250 message to be displayed as red and blinking.

___ 15. Restart the payroll application. When it becomes active, cancel the payroll application with a PAYROLL CANCEL command. Verify that your MRT performed the following tasks:

- Suppressed the IEE400I and $HASP395 messages. These messages should still appear in syslog.
- Revised the IEF450I message to be displayed in yellow.
- Deleted the IEE301I message. This message is to be suppressed and not logged to syslog.
- Revised the $HASP250 message to be displayed as red and blinking.

___ 16. If your lab version of VAPMRT1 did not function as expected, perform the following steps:

___ a. Switch to your TSO emulator session and make the appropriate changes.

___ b. From your NetView emulator session, issue the REVISE MSG command to test and load the modified VAPMRT1.

___ c. Repeat the PAYROLL START, STOP, and CANCEL steps.

___ 17. When you complete the lab, turn off message revision by issuing the REVISE MSG,OFF command. A message similar to the following one is displayed:

```
DSI231I NO 'Message Revision Table' IS ACTIVE
```


Answers to Lab 2 Exercise 1 questions

Step 7

PAYROLL START should generate IEF695I, $HASP373, IEF403I, VAPL29000A, IEE600I, VAPL29010I, and VAPL29999A messages.

You might also see IEA630I message for the first time that you start the payroll application.

Step 8

PAYROLL STOP should generate IEE600I, VAPL29020I, IEF404I, $HASP395, and IEA989I messages.

Step 9

PAYROLL CANCEL should generate IEA989I, IEE400I, IEF450I, IEE301I, IEA989I, and $HASP395 messages.

Example MRT (VAPMRT1)

The following MRT is an example of VAPMRT1 that is coded for these lab exercises. Your MRT will most likely look different:

```
************************************************************
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*   Sample MRT for NetView for z/OS labs. *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
************************************************************

 UPON (MSGID='$HASP100'|MSGID='$HASP373'|MSGID='$HASP395'
      | MSGID='IEE400I')
    REVISE('N' DISPLAY)       ! suppress the message

 UPON (MSGID='IEF403I' | MSGID='IEF404I')
     NETVONLY                ! send to Netiew only


 UPON (MSGID='IEA989I' | MSGID='IEE301I' | MSGID='IEE600I')
    REVISE('Y' DELETE)       ! suppress, no log,no AUTO

UPON (MSGID='IEF450I')
 Select
  When(jobname='PAYROLL ') ! IEF450I from PAYROLL appl:
      REVISE('CY' COLOR) ! change to yellow
     REVISE(1.* 1 JOBNAME)! Append jobname to end of text
   Otherwise ! for all other appls:
      REVISE('CW' COLOR) ! change to white
     REVISE(1.* 1 JOBNAME) ! Append jobname to end of text
 End

UPON (MSGID='$HASP250')
   REVISE('Y' DISPLAY)        ! unsuppress the message
   REVISE('CR HB' COLOR)        ! make msgs red/blinking

UPON (ALWAYS)
   Select
     When(MSGID LEFT 6 = 'VAPL29')  ! Only VAPL29* messages
       REVISE('CB HR' COLOR)        ! make msgs blue/reverse
     Otherwise   ! All other messages
       EXIT      ! no change - exit
   End
```

Exercise 2-2: ASSIGN Command

Lab exercise objectives

This lab exercise uses the NetView **ASSIGN** command to route messages to NetView tasks, such as operators or automation tasks. At the end of the lab, you should be able to use the ASSIGN command to create operator groups and also define authorized receivers of messages.

Lab exercise instructions

Perform the following steps:

___ 1. Access your system and log on to NetView, following the instructions that your instructor provides.

___ 2. By default, just a few messages are assigned. Issue a LIST MSG=AUTH command to verify them. The following messages should be displayed:

```
LIST MSG=AUTH
DSI636I AUTH MESSAGE STRING: 'EZL*'
BNH647I     PRIORITY LEVEL: 3
DSI638I     PRI(1ST): AUTOAON
DSI642I END OF ASSIGN DISPLAY
```

The message assignment for EZL* exists if AON is or has been enabled. The assignment does not affect your lab exercise.

Unsolicited messages are processed by the default tasks within NetView: SSIR task for z/OS messages and PPT task for VTAM messages.

___ 3. Issue an AUTOTBL STATUS command. Verify that the LAB2TBL automation table is active. If not, activate it with the following command:

```
AUTOTBL MEMBER=LAB2TBL,INSERT LAST
```

___ 4. From TSO, submit the BR14 job from NV390.WORKSHOP.DSIPARM. This action creates unsolicited IEF403I and IEF404I messages.

When the job runs, browse the NetView log and shift to the left by pressing PF10. CNMCSSIR is the task that processes the messages.

___ 5. Assign IEF messages to your NetView operator ID by issuing the following command:

`ASSIGN MSG=IEF*,PRI=TSCCWxx` (where xx is your team number)

___ 6. Issue the LIST MSG=AUTH command to verify the assignment.

___ 7. Submit the BR14 job again from TSO. The IEF messages are displayed on your NCCF screen. The **%** next to the message ID indicates that the message was assigned to your operator ID. It also is displayed in the NetView log.

___ 8. Disassociate the message assignment by using the following command:

```
ASSIGN MSG=IEF*,DROP
```

___ 9. Remove the LAB2TBL automation table by using the following command:

```
AUTOTBL REMOVE,NAME=LAB2TBL
```

___ 10. Create an operator group called +TCPOPS and assign two automation operators to by using the ASSIGN command as follows:

```
ASSIGN GROUP=+OMEG OP=(OPER1,OPER2)
```

The following message should be displayed:

```
DSI633I ASSIGN COMMAND SUCCESSFULLY COMPLETED
```

___ 11. Use the LIST command as follows to display the operators in the +OMEG group:

```
LIST ASSIGN=GROUP,GROUP=+OMEG
```

The response should display OPER2 and OPER3 as members of the +OMEG group:

```
DSI180I GROUP ID: +OMEG
BNH647I     PRIORITY LEVEL: 3
DSI640I     OP(ALL):  OPER2 OPER3
DSI642I END OF ASSIGN DISPLAY
```

___ 12. Add the AUTO1 task to the +OMEG group as last and OPER1 to the +OMEG group as first. Issue the following commands:

```
ASSIGN GROUP=+OMEG OP=(AUTO1) ADDLAST
ASSIGN GROUP=+OMEG OP=(OPER1) ADDFIRST
```

Display the operators in the +OMEG group again by issuing the LIST command as follows:

```
LIST ASSIGN=GROUP,GROUP=+OMEG
```

The response should be displayed as follows:

```
DSI180I GROUP ID: +OMEG
BNH647I     PRIORITY LEVEL: 3
DSI640I     OP(ALL):  OPER1   OPER2   OPER3   AUTO1
DSI642I END OF ASSIGN DISPLAY
```

At this point, the +OMEG group contains OPER1 as first in the task list and AUTO1 as last.

Tip: If no active operator is in the primary group, the secondary group or operator does not receive any messages.

___ 13. Assign the +OMEG group as the authorized receiver for all CNDL* (OMEGAMON Subsystem messages) and your NetView operator ID (TSCCWxx) as the secondary receiver. Issue the following command:

```
ASSIGN MSG=CNDL*,PRI=+OMEG,SEC=TSCCWxx
```

Verify the message assignment by issuing the LIST command as follows:

```
LIST ASSIGN=AUTH,MSGID=CNDL*
```

The response should be displayed as follows:

```
DSI636I AUTH MESSAGE STRING: 'CNDL*'
BNH647I     PRIORITY LEVEL: 3
DSI638I     PRI(1ST): +OMEG
DSI639I     SEC(ALL): TSCCW01
DSI642I END OF ASSIGN DISPLAY
```

___ 14. Assign your NetView operator ID (TSCCWxx) as the authorized receiver for the following messages:

Message CNDL001I OMEGAMON SUBSYSTEM V620 INITIALIZED - SSID = CNDL

Issue the following command:

```
ASSIGN MSG=CNDL001I,PRI=TSCCWxx
```

Message CNDL002I OMEGAMON SUBSYSTEM V620 TERMINATED - SSID = CNDL

Issue the following command:

```
ASSIGN MSG=CNDL002I,PRI=TSCCWxx
```

At this point, TSCCWxx is the authorized receiver for CNDL001I and CNDL002I and is also the secondary receiver for all other CNDL* messages. The +OMEG group is the authorized receiver for all CNDL* messages.

Tip: As discussed in the lecture, more specific assignments, such as CNDL001I, take precedence over less specific assignments, such as CNDL*.

___ 15. Stop the OMEGAMON subsystem by issuing a P CANACN command from either NetView or the system console.

What did you see on your NetView operator screen? _____

_____

_____

___ 16. Restart the OMEGAMON subsystem by issuing an S CANACN command. What did you see on your NetView screen?

_____

_____

_____

___ 17. Remove all group and message assignments that you created during the lab exercise.

Delete the +OMEG operator group by issuing the following command:

```
ASSIGN GROUP=+OMEG,DROP
```

Delete the assignment for all CNDL* messages by issuing the following command:

```
ASSIGN MSG=CNDL*,DROP
```

Answers to Lab 2 Exercise 2 questions

Step 15

You should see one CNDL message on TSCCWxx with TSCCWxx as the authorized receiver for the CNDL002I message:

```
% CNDL002I  OMEGAMON SUBSYSTEM V620 TERMINATED - SSID = CNDL
```

The % indicates that this message is sent to the authorized receiver. Browse DSILOG to see that automation table was driven for the CNDL002I message on TSCCWxx. Look for these messages in DSILOG, and use PF10 to scroll left. TSCCWxx is the task.

```
* 12:39:00 E CNDL030I  FUNCTION KCNDLCF  STOPPED
* 12:39:00 E CNDL030I  FUNCTION DIOPINIT STOPPED
% 12:39:00 E CNDL002I  OMEGAMON SUBSYSTEM V620 TERMINATED - SSID =
CNDL
  12:45:45 * BR NETLOGA
```

Step 16

You should also here see several CNDL messages on TSCCWxx:

```
% CNDL001I  OMEGAMON SUBSYSTEM V620 INITIALIZED - SSID = CNDL
* CNDL190I  SUBSYSTEM ADDRESS SPACE INITIALIZATION ROUTINE
COMPLETED
  SUCCESSFULLY
* CNDL034I  SUBSYSTEM START MEMBER KCNSTR00
* CNDL027I  FUNCTION KCNDLCF  STARTED
* CNDL027I  FUNCTION DIOPINIT STARTED
```

The % indicates TSCCWxx is the authorized receiver for message CNDL001I. The * indicates TSCCWxx is a secondary receiver of the message. Each of the messages flagged as secondary copies also routed to an authorized receiver. For example, if you browse DSILOG, you see that CNDL001I appears twice. CNDL001I appears once for the secondary receiver and once for the authorized receiver.

Exercise 2-3: NetView and z/OS consoles

Lab exercise objectives

This lab exercise uses the NetView console management commands to manage EMCS consoles for NetView tasks. At the end of the lab, you should be able to perform the following tasks:

- Use the NetView **DISCONID**, **GETCONID**, and **RELCONID** commands to manage z/OS consoles assigned to NetView tasks.
- Explain how z/OS consoles get assigned to NetView tasks.

Lab exercise instructions

Perform the following tasks:

___ 1. From NetView AOFDA domain, issue a HELP GETCONID command and read how a console ID associates with a NetView task. What is the default console name if none is specified (GETCONID, CNMSTYLE definitions, and so on)?

_____

___ 2. Issue a DISCONID command to display all consoles for all NetView tasks. You should see messages similar to the following text:

```
* AOFDA    DISCONID
' AOFDA
CNM492I  OPERATOR ID   CONSOLE ID    CONSOLE NAME
CNM492I  -----------   ----------    ------------
CNM492I  TSCCW01         EXTENDED    TSC10WAD
CNM492I  AUTOAON         EXTENDED    AUTNOAAD
CNM492I  AUTO1                  0 *  *MASTER*
CNM492I  END DISPLAY
```

All tasks use the default console name assignment of the task name.

___ 3. If your operator ID, TSCCW*xx*, already has a console, issue the RELCONID command to remove any previously obtained consoles. You should see the following response:

```
CNM569I MVS CONSOLE ID RELEASED
```

___ 4. Issue a LIST * command for your NetView operator. Within the response, you should see a message similar to the following text:

```
DEFAULT MVS CONSOLE NAME: TSC10WAD
```

___ 5. Issue an MVS D A,L command. When that command completes, issue another DISCONID command. You should see a response similar to the following text:

```
* AOFDA    DISCONID
' AOFDA
CNM492I  OPERATOR ID   CONSOLE ID    CONSOLE NAME
CNM492I  -----------   ----------    ------------
CNM492I  TSCCW01         EXTENDED    TSC10WAD
CNM492I  AUTOAON         EXTENDED    AUTNOAAD
CNM492I  AUTO1                  0 *  *MASTER*
CNM492I  END DISPLAY
```

TSCCWxx now has a console associated with it. This was assigned automatically when you issued the MVS D A,L command.

___ 6. Route a MVS D A,L command to the AOFDB domain.

Tip: Issue the AOFDB: MVS D A,L command.

You see the result from the display. Issue the DISCONID command first on the AOFDA domain, and nothing new is displayed. However, if you issue the AOFDB: DISCONID command, a new console is allocated on the AOFDB domain for your TSCCWxx user as shown on the following text:

```
* AOFDA    AOFDB: DISCONID
* AOFDB    DISCONID
' AOFDB
CNM492I  OPERATOR ID   CONSOLE ID    CONSOLE NAME
CNM492I  -----------   ----------    ------------
CNM492I  AUTO1                  0 *  *MASTER*
CNM492I  TSCCW01         EXTENDED    TSC10WBD
CNM492I  END DISPLAY
```

___ 7. Switch to your z/OS system console. Issue a NetView command from the system console, using the default command designator of %. Issue %LIST * command at the system console. You should see the following response:

```
STATION: AUTO1       TERM: AUTO1
HCOPY: NOT ACTIVE    PROFILE: DSIPROFC
STATUS: ACTIVE       IDLE MINUTES: 0
ATTENDED: YES CONS   CURRENT COMMAND: LIST
AUTHRCVR: NO         CONTROL: GLOBAL
NGMFADMN: NO         DEFAULT MVS CONSOLE NAME: AUT1OTAD
NGMFVSPN: NNNN (NO SPAN CHECKING ON NMC VIEWS)
NGMFCMDS: YES        AUTOTASK: YES
IP ADDRESS:  N/A
OP CLASS LIST: NONE
DOMAIN LIST: AOFDA (I) AOFDB (I)
ACTIVE SPAN LIST: NONE
Task Serial:  604  REXX Environments: 2 (1%)
Messages  Pending:  0  Held:  0
WLM Service Class:  Not Available
END OF STATUS DISPLAY
```

Why did the LIST command run under AUTO1? _____

_____

Answers to Lab 2 Exercise 3 questions

Step 1

The default console name is the name of the task that issues the MVS command, for example, TSCCWxx.

Step 6

To control the creation of unique console names, you can perform one of the following tasks:

- Issue a RELCONID command before each MVS command to free up the console name.
- Issue a GETCONID command to specify a unique console name for each task.
- Edit CNMSTYLE and define a console mask with the ConsMask statement.

Step 7

Commands from the system console run under the AUTO1 task because it is associated with the *master* console as displayed by the following text:

```
AUTOTASK.?Master.Console = *ANY*
function.autotask.Master = AUTO1 // autotask with master console
```


Student exercises for Unit 3

Lab exercise overview

In this exercise, you learn to route commands to other NetView tasks within the same domain or in other NetView domains. You also learn to create and manage NetView timers.

Exercise 3-2: Routing commands to other tasks and domains

Lab exercise objectives

This lab introduces you to the NetView facilities to route commands to tasks within the same NetView domain or other NetView domains. At the end of the lab, you should be able to perform the following tasks:

- Use EXCMD to route commands to another task in the same NetView domain.
- Use RMTCMD to route commands to tasks in other NetView domains.
- Use labelled commands for both EXCMD and RMTCMD.

Lab exercise instructions

Perform the following steps:

___ 1. Access your system and log on to the NetView AOFDA domain, following the instructions that the instructor provides.

___ 2. Issue the LIST * command on task AUTO1 by entering the following command:

```
EXCMD AUTO1,LIST *
```

The output of the LIST command is not displayed, just a message that the EXCMD command ran. To see the actual output of the LIST command, browse the NetView log.

___ 3. You can send a command to another task and see the output of the command by using a labelled command as follows:

```
/AUTO1: LIST *
```

___ 4. The RMTCMD command routes commands to another NetView domain, creating a distributed autotask on the remote NetView that uses your operator ID (by default). Because you are starting an autotask, no password is required.

Use RMTCMD to establish a connection from AOFDA to AOFDB domains by issuing the following command:

```
RMTCMD LU=AOFDB,LIST *
```

The output of the LIST * command shows that you are a distributed autotask. How can you determine that? _____

_____

___ 5. The RMTCMD command supports communication over SNA and TCP/IP. Using the online help, can you determine whether your RMTCMD was using SNA or TCP/IP?

_____

___ 6. You can use a labelled command as follows to shorten the syntax of RMTCMD:

```
AOFDB: LIST *
```

___ 7. End your RMTCMD session by issuing the following command:

```
AOFDB: LOGOFF
```


Answers to Lab 3 Exercise 2 questions

Step 4

In the response to the LIST command, the STATION name and TERM name are the same:

```
STATION: TSCCWxx TERM: TSCCWxx
```

Step 5

When you specify the RMTCMD command, you must choose between using the LU or the DOMAIN parameters. If you use the LU parameter, you request an SNA session. If you use the DOMAIN parameter, you request a TCP/IP session. If your NetView domain supports both SNA and TCP/IP for RMTCMD, NetView attempts to start the TCP/IP session first if you specify a labelled RMTCMD.

Each NetView contains definitions in CNMSTYLE for controlling SNA and IP access. Browse CNMSTYLE and find the RMTINIT statement. Read the comments for further information.

You can also issue a RMTCMD QUERY RMTDOMS command to determine the active RMTCMD sessions and the communication method. An example follows:

```
* AOFDA    RMTCMD QUERY RMTDOMS
' AOFDA
BNH060I RMTCMD QUERY INFORMATION
BNH061I ------------------------------------
BNH068I REMOTE NETVIEW     VERSION  TRANSPORT
BNH061I -----------------  -------  ---------
BNH069I USIBMES.AOFDB      V6R1     TCP/IP
BNH690I   IP ADDRESS:  10.31.187.195  PORT:   4022
```

Exercise 3-3: Timer commands

Lab exercise objectives

This lab introduces you to the NetView facilities for scheduling and managing timers in a single NetView domain or multiple NetView domains. At the end of the lab, you should be able to perform the following tasks:

- Schedule AT, EVERY, and AFTER timers in NetView.
- Use the LIST TIMER command to display NetView timers.
- Use the TIMER panel interface to manage NetView timers, including timers in another NetView domain.

Lab exercise instructions

Perform the following steps:

___ 1. Issue the following AFTER command to schedule a NetView timer to run the MYCLIST command after one minute has passed:

```
AFTER 00:01:00,MYCLIST
```

___ 2. Issue the following LIST TIMER command to verify that the timer has been set:

```
LIST TIMER=ALL
```

The response should resemble the following text:

```
* AOFDA    LIST TIMER=ALL
' AOFDA
DISPLAY OF OUTSTANDING TIMER REQUESTS
TYPE: AFTER   TIME: 07/13/11 13:50:59
  COMMAND:  MYCLIST
  OP: TSCCW01  (TSCCW01 )   ID: SYS00013            TIMEFMSG: NO
GMT
1 TIMER ELEMENT(S) FOUND FOR TSCCW01
END OF DISPLAY
```

Wait one minute. When the AFTER timer opens, you should see messages similar to the following text:

```
- AOFDA    DSI208I TIME EXPIRATION - ID= 'SYS00013' - CMD=
'MYCLIST'
C AOFDA    Hello from MYCLIST running under task  TSCCW01
```

The DSI208I message is displayed because of the AFTER timer. The second message is displayed because of the MYCLIST command.

___ 3. Schedule an AFTER timer as follows to issue MYCLIST under AUTO1 after one minute has passed:

```
AFTER 00:01:00,ROUTE=AUTO1,MYCLIST
```

___ 4. Use the LIST TIMER command as follows to verify that the timer has been set:

```
LIST TIMER=ALL,OP=AUTO1
```

You need to specify the OP parameter to display the timer for AUTO1.

___ 5. When one minute has passed, browse the NetView log to verify that AUTO1 issued the MYCLIST command.

___ 6. Schedule an EVERY timer command as follows to issue MYCLIST under your operator ID every thirty seconds, giving the timer an ID of MYTIMER:

```
EVERY 00:00:30,ID=MYTIMER,MYCLIST
```

___ 7. When the timer has run twice, issue the PURGE TIMER command as follows to remove the timer:

```
PURGE TIMER=MYTIMER
```

___ 8. Schedule an AT timer command for MYCLIST to run under your operator ID at a time that is *30 minutes from now*. Be sure to save the timer to the Save/Restore database. An example follows:

```
AT 10:30:00,SAVE,MYCLIST
```

___ 9. Issue the LIST TIMER command as follows to verify that the timer has been set and saved:

```
LIST TIMER=ALL
```

___ 10. Issue the TIMER command to open a full-screen panel interface to the NetView timer commands. You should see the *Timer Management* panel (EZLK6000) with several timers scheduled by NetView. *Do not modify or delete any of the NetView timers.*

___ 11. The cursor should be positioned next to the first timer in the list. Select option one (add) to create a new timer.

___ 12. Change the Timer Type to be an AFTER timer and press Enter. You should see the Set AFTER Timer panel (EZLK6130).

___ 13. Fill in the appropriate fields to schedule an AFTER timer to run the MYCLIST command under the AUTO1 task in 30 seconds. Press Enter to create the timer.

You should see a message similar to the following text:

```
EZL973I REQUESTED TIMER SYS00098 ADDED ON AOFDA
```

Press PF3 to return to the Timer Management panel.

___ 14. After 30 seconds have elapsed, press PF5 to refresh the Timer Management panel. The AFTER timer should no longer be displayed.

Why were you not interrupted when the AFTER timer ran? _____

_____

___ 15. Select option one (add) to create another timer. Define an EVERY timer to run the MYCLIST command every 30 seconds on your operator ID. Specify a timer ID of MYTIMER.

Tip: If you are not on the Set EVERY timer panel (EZLK6110), change the Timer Type field for an EVERY timer and press Enter. Also, you must specify a day (1 through 9). Select option nine and keep the default of 000 days to schedule the timer today.

Press PF3 when you have defined the EVERY timer.

___ 16. Press PF6 to ROLL to the NCCF screen. Each time MYTIMER opens, a message similar to the following text is displayed:

```
Hello from MYCLIST running under task  TSCCWxx
```

___ 17. Press PF6 to ROLL back to the Timer Management panel. Tab over to MYTIMER and select option two (display/change) to edit the timer. You should see the Set EVERY Timer panel (EZLK6110).

___ 18. Change the timer type from EVERY to AT, and specify a time one minute from now. Press Enter to save the timer.

You should see message EZL974I REQUESTED TIMER MYTIMER CHANGED ON AOFDA.

Press PF3 to return to the Timer Management panel.

___ 19. Press PF6 to ROLL to the NCCF screen and wait for MYTIMER to open. A message similar to the following text is displayed:

```
Hello from MYCLIST running under task  TSCCWxx
```

___ 20. After MYTIMER runs, press PF6 to roll back to the Timer Management panel. Press PF5 to refresh the timers. MYTIMER should no longer be displayed.

___ 21. Type a question mark (**?**) in the Target field and press Enter. The Remote Target Selection panel (EZLK5500) should display the AOFDA and AOFDB domains, along with session information for each.

___ 22. **S**elect the AOFDB domain and press Enter. You should now see timers that are scheduled to run in the NetView AOFDB domain.

Several fields are now filled in as follows:

- The IP Addr and Port fields show that the TIMER function is using RMTCMD over TCP/IP to communicate with the AOFDB domain.
- The Remote Target Date and Time field is now filled in. In some cases, the target domain might not be in the same time zone.

___ 23. Press PF3 to close the Timer Management panel.

___ 24. Issue the TIMER command again. You should see the Timer Management panel with timers for your local NetView domain. By default, the TIMER command always displays timers in your local NetView domain. You can view timers in remote NetView domains if you specify a TARGET parameter on the TIMER command. For more information, see the help for the TIMER command.

Answers to Lab 3 Exercise 3 questions

Step 14

Because the AFTER timer was scheduled under AUTO1, you do not see any messages on your operator ID.

Exercise 3-4: Command Revision Table (CRT)

Lab exercise objectives

This lab introduces you to NetView Command Revision Table (CRT). At the end of the lab, you should be able to perform the following tasks:

- Install the necessary code for activating the CRT.
- Explain how to use the CRT to revise commands.
- Modify the CRT to achieve the required action for a specific command.

Lab exercise instructions

Perform the following steps:

___ 1. To enable the Command Revision Table, some items must be in place. Issue MVS D PROG,LNKLST to verify that NETV610.SCNMLNKN is in your Linklist. This library contains the DSIRCVEX module, which must be in the Linklist. This step has been done for you to prepare the system.

___ 2. Modify the MPFLST to include the user exit that handles the command revision. This step has been prepared for you also. Issue MVS D MPF and verify user exit DSIRVCEX is installed.

With these functions in place, you are now ready to start working with the CRT.

___ 3. Check that no CRT is active by issuing REVISE STATUS. Activate the default CRT CNMSCRT1 by issuing REVISE CMD,MEMBER=CNMSCRT1. You should receive the following response:

```
BNH608I 'LOAD CMD CNMSCRT1' request completed successfull
```

___ 4. Test your loaded CRT by issuing MVS SEND ‘THIS IS INTENDED FOR ALL’ to receive the following response:

```
* AOFDA    MVS SEND 'THIS IS INTENDED FOR ALL'
E AOFDA    TLH447E Please do not broadcast to all.
```

This is the CRT taking control, and the command is stopped. If you issue MVS SEND ‘ONLY FOR TSCCWxx’ USER=TSCCWxx instead, the command is passed for execution**.**

___ 5. Browse CNMSCRT1 and look at the sample logic.

___ 6. Look at some more logic with the CRT. The system comes with both a sample CRT CNMSCRT1, which we looked at previously. The same sample includes a REXX procedure as follows:

```
UPON(CMDVERB = 'V' ! CMDVERB = 'VARY')
   SELECT
     WHEN(WORD 2 = 'NETVIEW')                    ! V
NetView,(anytext)
        NETVONLY=CNMSRVMC                       ! handle in NetView
     OTHERWISE                ! all other VARY cmds untouched
   END
```

The NETVONLY=CNMSRVMC calls the REXX procedure CNMSRVMC, which is a provided sample. In such a REXX procedure, you can create your own logic to decide whether or not to issue the command.

___ 7. Copy the CNMSCRT1 to your NV390.WORKSHOP.DSIPARM and modify it so that a STOP or P command on procedure CANACN calls REXX CRT1. Browse CRT1 to see what occurs if you try to do STOP CANACN. CRT1 is modified from CNMSCRT1. Several REXX procedures can be called by different NETVONLY= statements.

___ 8. When you have changed your CNMSCRT1, activate it. Stop the active copy (REVISE CMD,OFF) then activate it with REVISE CMD,MEMBER=CNMSCRT1.

___ 9. Test your new CRT by issuing MVS P CANACN. The response should be as follows:

```
* AOFDA    MVS P CANACN
> AOFDA    12 TLH916W OMEGAMON Subsystem should not be stopped
           before the TEMS. Answer "Y" to continue

Reply to the WTOR with N
E AOFDA    IEE600I REPLY TO 12 IS;N
E AOFDA    TLH917I STOP CANACN CANCELED.
```

___ 10. Stop the active CRT by issuing REVISE CMD,OFF.

Answers to Lab 3 Exercise 4 questions

Step 7

The following entries should be incorporated into the CNMSCRT1 member:

```
*  UPON statements must follow control statements
UPON(CMDVERB = 'P' | CMDVERB = 'STOP')
   SELECT
     WHEN(WORD 2 = 'CANACN')                    ! P CANACN
        NETVONLY=CRT1                         ! handle in NetView
     OTHERWISE                 ! all other VARY cmds untouched
   END
```


Student exercises for Unit 4

Lab exercise overview

In this exercise, you log on to NetView to manage the automation tables with the AUTOTBL and AUTOMAN commands.

Exercise 4-1: Managing the NetView automation tables

Lab exercise objectives

This lab introduces you to managing the NetView automation table. At the end of the lab, you should be able to perform the following tasks:

- Use the AUTOTBL command to display the active automation table and load additional automation tables.
- Use the AUTOMAN panel interface to manage the automation tables.

Lab exercise instructions

Perform the following steps:

___ 1. Display the currently active automation table by issuing the AUTOTBL STATUS command. You should see a response similar to the following text:

```
* AOFDA    AUTOTBL STATUS
' AOFDA
BNH361I THE AUTOMATION TABLE CONSISTS OF THE FOLLOWING LIST OF
MEMBERS:
AOFDAPPT COMPLETED INSERT  FOR TABLE Ä1: DSITBL01 AT 11/10/10
08:44:14 (FIRST)
AOFDAPPT COMPLETED INSERT  FOR TABLE Ä2: VAPLAT   AT 11/10/10
08:44:14
```

Two automation tables are active: DSITBL01 and VAPLAT. Automation tables can be loaded by using the AUTOTBL command or, as in this case, when NetView initializes. Automation tables that load during initialization are specified in CNMSTYLE with the AUTOCMD.statement.

___ 2. Browse the CNMSTYLE member. On the command line (CMD==>), specify ALL 1.8 AUTOCMD**.** and press Enter. The trailing period is necessary. Running this command displays all of the AUTOCMD.statements in CNMSTYLE and its included members.

You should see several AUTOCMD. statements similar to the following text:

```
AUTOCMD.DSITBL01.list = ListName
AUTOCMD.DSITBL01.marker = (AON)
AUTOCMD.DSITBL01.order  = *FIRST*
AUTOCMD.VAPLAT.MARKER = VPL
AUTOCMD.VAPLAT.ORDER  = B
```

Automation tables are loaded based on the *order* statement in CNMSTYLE. DSITBL01 loads first by default (*FIRST*). All other automation tables load based on an EBCDIC sort of the *order* statement. VAPLAT is a user-defined automation table that loads second. In this case, there are no other automation tables.

Suppose you were asked to define AUTOCMD.statements for inserting an automation table between DSITBL01 and VAPLAT. What would your AUTOCMD.statements look like?

_____

_____

___ 3. Issue an AUTOTBL command to insert the LAB4TBL as the first automation table by issuing the following command:

```
AUTOTBL MEMBER=LAB4TBL,INSERT FIRST
```

The AUTOTBL request should fail, and respond with the following message:

```
BNH367E UNABLE TO COMPLETE AUTOTBL INSERT REQUEST.  REASON CODE:
104
```

Why does the insert initially fail? _____

_____

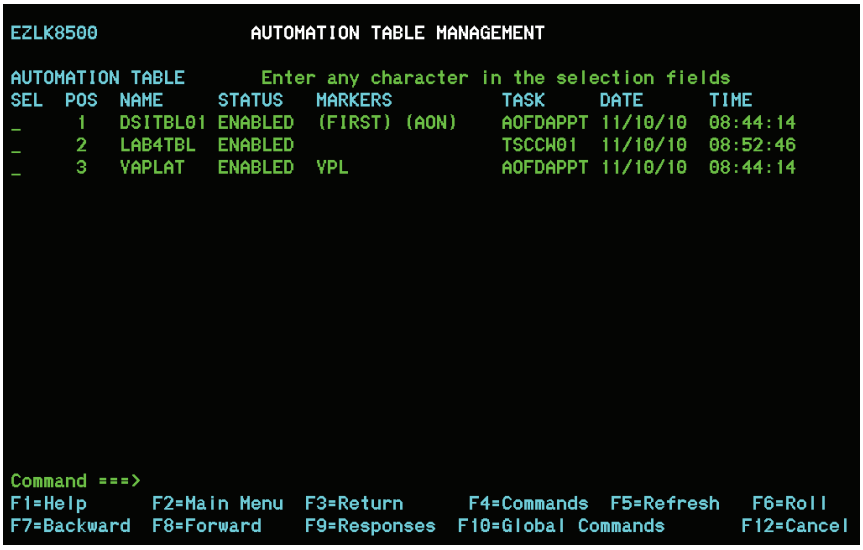
___ 4. Insert LAB4TBL as the second automation table by issuing the following command:

```
AUTOTBL MEMBER=LAB4TBL,INSERT AT=2
```

The following BNH360I message is displayed:

```
BNH360I INSERT REQUEST COMPLETED FOR DSIPARM MEMBER LAB4TBL AT
LOCATION 2 WITHIN THE LIST OF ACTIVE AUTOMATION TABLES
```


___ 5. Issue the AUTOMAN command to display the active automation tables in a panel interface. You should see the Automation Table Management panel (EZLK8500) as follows:



This panel displays the three automation tables that are active with LAB4TBL inserted as the second table.

___ 6. Type any non-blank character in the SEL column next to LAB4TBL and press PF4 or ENTER. This action displays a list of commands that can be issued against LAB4TBL. A window opens on the panel with a list of nine commands. From here, you can enable or disable, test the table, and so on.

___ 7. Select option nine (Display options) to view the list of display options.

___ 8. Select option one (Browse table with includes). The NetView browse function displays the source of the LAB4TBL automation table. The comments indicate one that LAB4TBL is currently *under construction*. Press PF3 to return to AUTOMAN.

___ 9. If LAB4TBL does not contain any automation logic, delete it from the list of active automation tables. Press PF3 until you return to the command window. Issue Unload LAB4TBL, option seven. The Automation Table Management panel (EZLK8500) displays a message as follows, indicating the unload was successful:

```
EZL919I ALL ACTIONS SUCCESSFULLY COMPLETED
```

___ 10. Press PF3 to end AUTOMAN. You use AUTOMAN again to load automation tables in the lab exercises for Unit 5.

Answers to Lab 4 Exercise 1 questions

Step 2

Suppose you were asked to insert an automation table named MYTABLE between DSITBL01 and VAPLAT when NetView initializes. The CNMSTYLE statements should look similar to the following text:

```
AUTOCMD.mytable.MARKER = MYT
AUTOCMD.mytable.ORDER  = A
```

Step 3

The insert fails because DSITBL01 is locked as the first automation table with an AUTOCMD.DSITBL01.*order* statement in CNMSTYLE, as the following text shows:

AUTOCMD.DSITBL01.order = *FIRST*

The help for message BNH367E return code 104 states as follows:

```
A request to establish a table as the first table
within the list of automation tables failed because
another table is already established as the locked
first table. The table must be removed before similar
attempts are successful. See the description of the
AUTOTBL command for more information about the FIRST
keyword.
```


Student exercises for Unit 5

Lab exercise overview

In this exercise, you code automation table statements to trap and automate messages that indicate the status of resources. You define automation to run commands on automation tasks.

Data sets used in this lab

NV390.WORKSHOP.DSIPARM _____

Exercise 5-1: Coding automation table statements

Lab exercise objectives

This lab provides an in-depth look at coding the NetView automation table statements. At the end of the lab, you should be able to code automation table entries to perform the following tasks:

- Trap messages.
- Take actions when the messages occur.

Lab exercise instructions

This exercise is about defining and starting automation operators, Perform the following steps:

___ 1. Log on to TSO and edit the **LABOPFU** member in NV390.WORKSHOP.DSIPARM. Create NetView operator definitions for several automation tasks: MVSAUTO, TSOAUTO, RMFAUTO, and MISCAUTO

For each operator definition, use definitions similar to *TSCCW01*, replacing it with the automation task name:

```
TSCCW01 OPERATOR    PASSWORD=TSCCW01
        PROFILEN    DSIPROFA
```

___ 2. From NetView AOFDA domain, issue the following REFRESH command to dynamically enable the new operator definitions:

```
REFRESH OPERS
```

You should see the following messages:

```
DWO831I OPERATOR MVSAUTO IS ADDED TO THE GROUP OF VALID NETVIEW
OPERATORS
DWO831I OPERATOR TSOAUTO IS ADDED TO THE GROUP OF VALID NETVIEW
OPERATORS
DWO831I OPERATOR RMFAUTO IS ADDED TO THE GROUP OF VALID NETVIEW
OPERATORS
DWO831I OPERATOR MISCAUTO IS ADDED TO THE GROUP OF VALID NETVIEW
OPERATORS
DSI633I REFRESH COMMAND SUCCESSFULLY COMPLETED
```

If you do not see any messages from the REFRESH OPERS command, LABOPFU loaded because of the memStore function. Unload LABOPFU with the MEMSTOUT command and run REFRESH OPERS again as follows:

```
MEMSTOUT UNLOAD DSIPARM.LABOPFU
REFRESH OPERS
```

___ 3. Issue an AUTOTASK to log MVSAUTO on as an automation task. An example follows:

```
AUTOTASK OPID=MVSAUTO
```

You should see the following response:

```
CNM570I STARTING AUTOMATION TASK MVSAUTO
DSI530I 'MVSAUTO' : 'OST' IS READY AND WAITING FOR WORK
```

Issue an AUTOTASK for the three remaining automation tasks that you just defined.

Lab exercise instructions

This exercise is about coding automation table statements, Perform the following steps:

___ 4. Switch to your TSO session and edit the LAB5TBL member in NV390.WORKSHOP.DSIPARM. You create automation table statements to trap specific messages and run commands to restart applications, such as TSO, RMF, and so on.

Note: Type “CAPS OFF” on the TSO command line before you begin creating your automation table statements. With this setting, you can use mixed-case text in the automation table; improving its readability.

Create automation statements as follows:

- IEF404I automation: (generic message as follows, indicating application has ended)

  `IEF404I jobname - ENDED - TIME=14.17.41`

  ___ a. Create automation to trap the IEF404I and restart the failed job (second token, *jobname*) automatically. Route the command to MVSAUTO.

  ___ b. Log the message in NetView.

  ___ c. Suppress the message from the system log.

- IEF403I automation: (generic message as follows, indicating application now active)

  `IEF403I jobname - STARTED - TIME=14.27.36`

  ___ a. Create automation to trap the IEF403I and call the APPLUP EXEC with the jobname (second token). Route the command to MVSAUTO.

  ___ b. Log the message in NetView.

  ___ c. Suppress the message from the system log.

- IKT010D automation: (TSO users active message with reply, as follows)

  `nn IKT010D 00001 USER(S) ACTIVE, REPLY 'SIC' OR 'FSTOP'`

  ___ a. Create automation to trap the IKT010D message and reply automatically with FSTOP. Route the reply to the TSOAUTO task.

  ___ b. Log the message in NetView.

  ___ c. Suppress the message from the system log.

  TSO restarts when the IEF404I message is issued.

- CNDL001I automation: (OMEGAMON Subsystem Initialized message as follows)

  `CNDL001I  OMEGAMON SUBSYSTEM V620 INITIALIZED - SSID = CNDL`

  ___ a. Create automation to trap the CNDL001I message and call the APPLUP EXEC with the SSID name from the message.

  ___ b. Log the message in NetView.

  ___ c. Suppress the message from the system log.

___ 5. When finished creating the automation table statements in LAB5TBL, press PF3 to save your work.

Lab exercise instructions

This exercise is about testing your automation table statements, Perform the following steps:

___ 6. In NetView AOFDA, use the following AUTOTBL command to test the syntax of your LAB5TBL:

```
AUTOTBL MEMBER=LAB5TBL,TEST
```

You should seetheCNM501I message. If not, correct the errors and issue the AUTOTBL command to test LAB5TBL again. The response should be similar to the following text:

```
CNM501I TEST OF NETVIEW AUTOMATION FILE "LAB5TBL" WAS SUCCESSFUL
```

___ 7. When your automation table test is successful, load it as the second table in the list, using the AUTOMAN panel interface.

___ 8. When LAB5TBL loads correctly, use your system console session to stop several of the applications as follows to test your automation:

```
P RMF
P CANACN
P AUTOSSI
```

___ 9. Browse DSILOG and look for CNM493I messages pertaining to each application that stopped. For example, for the RMF application, you should see messages similar to the following text:

```
CNM493I LAB5TBL  : #0000023 : MVS START RMF
IEF404I RMF - ENDED - TIME=15.49.09
```

RMF has ended. Automation for IEF404I has parsed the jobname, RMF, from the failure message and issued the MVS START RMF command as follows.

```
CNM493I LAB5TBL  : #0000036 : APPLUP RMF
IEF403I RMF - STARTED - TIME=15.49.09
```

RMF has started. Automation for IEF403I has parsed the jobname, RMF, and passed it as a parameter to the APPLUP EXEC as follows.

```
DSI039I MSG FROM MVSAUTO  : JOB NAME RMF IS NOW ACTIVE.
```

The APPLUP EXEC issues the NetView MSG command to inform operators that RMF is now active.

___ 10. When all other automation works successfully, issue P TSO to bring TSO down while you are still logged on. Running the command tests the automation table statements that you created for TSO.

You should see messages similar to these in DSILOG. Your automation should parse the reply number (84) and respond with FSTOP.

```
- CNM493I LAB5TBL  : #0000049 : MVS REPLY 84,FSTOP
> 84 IKT010D 00001 USER(S) ACTIVE, REPLY 'SIC' OR 'FSTOP'
E IEE600I REPLY TO 84 IS;FSTOP
```

What does the #00000049 in the example CNM493I mean? _____

You can use the CNM493I message as an audit trail of the automated actions taken when an event is received. This can be very helpful when you initially create your automation table statements.

Tip: After your automation is in place, you can turn off the logging of CNM493I by issuing the NetView OVERRIDE or DEFAULTS commands. Or you can modify CNMSTYLE with a DEFAULTS.CNM493I={YES | NO} statement.

___ 11. Issue the AUTOMAN command again. **S**elect your automation table, LAB5TBL, and press PF4 (or ENTER) to display the command window.

Select option nine (DISPLAY options), which opens another window with a list of display selections. Select option seven (display AUTOCNT statistics).

You should see a panel similar to the following graphic, which is an example of the AUTOCNT STATISTICS display for LAB5TBL:

```
CNMKWIND OUTPUT FROM  AUTOCNT STATISTICS                  LINE 0 OF 46
*------------------------------ Top of Data --------------------------------*
DWO800I AUTOMATION TABLE MSG DETAIL REPORT BY TSCCW01
-----------( LAB5TBL /AUTOM728 MESSAGE DETAILS  11/11/10 03:30:15 )------------
TSCCW01  COMPLETED INSERT  FOR TABLE Ä2: LAB5TBL  AT 11/11/10 03:20:58
                                                        !-- PERCENTAGES --!
STMT L SEQUENCE NUMBER/ MEMBER    COMPARE    MATCH  E C A D MATCH/ COMP/ MATCH/
NUMB I LABEL NAME       NAME      COUNT      COUNT  C I I I COMP   TOTAL TOTAL
--------------------------------------------------------------------------------
MEMB                    LAB5TBL
0003 S Ä0000012         LAB5TBL       147        4  0          2.7  100.0   2.7
0004 S Ä0000014         LAB5TBL         4        0  0          0.0    2.7   0.0
0005 S Ä0000015         LAB5TBL         4        0  0          0.0    2.7   0.0
0006 S Ä0000016         LAB5TBL         4        0  0          0.0    2.7   0.0
0007 S Ä0000017         LAB5TBL         4        0  0          0.0    2.7   0.0
0008 S Ä0000018         LAB5TBL         4        0  0          0.0    2.7   0.0
0009 S Ä0000019         LAB5TBL         4        0  0          0.0    2.7   0.0
0010 S Ä0000020         LAB5TBL         4        0  0          0.0    2.7   0.0
0011 S Ä0000021         LAB5TBL         4        0  0          0.0    2.7   0.0
0012 S Ä0000022         LAB5TBL         4        1  0         25.0    2.7   0.7
0013 S Ä0000023         LAB5TBL         3        2  2         66.7    2.0   1.4
0014 S Ä0000029         LAB5TBL         1        1  1        100.0    0.7   0.7
0016 S Ä0000036         LAB5TBL       143        4  1          2.8   97.3   2.7
0017 S Ä0000042         LAB5TBL       139        0  1          0.0   94.6   0.0
0018 S Ä0000049         LAB5TBL       139        1  1          0.7   94.6   0.7
0019 S Ä0000057         LAB5TBL       138        0  1          0.0   93.9   0.0
--------------------------------------------------------------------------------
DWO800I AUTOMATION TABLE MSU DETAIL REPORT BY TSCCW01
------------( LAB5TBL /AUTOM728 MSU DETAILS  11/11/10 03:30:15 )---------------
NO MSU STATEMENTS IN CURRENT AUTOMATION TABLE
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>
```

For example, line number twelve was compared 147 times and resulted in 4 matches. Line number twelve is the following text:

```
IF MSGID = 'IEF404I' then
```

Four IEF404I messages matched in LAB5TBL.

Answers to Lab 5 Exercise 1 Questions

Step 10

Within the example CNM493I message, the #00000049 indicates the statement that matches in the automation table. The statement can be a line number or a sequence number. In this case, it is a relative line number (because of the # as the first character) within the automation table member, LAB5TBL. The online help for CNM493I describes this field as follows:

```
seqnum: The 8-character sequence number field of the first record
of the automation statement that generated this command.

Note: If this message is displayed for a statement that was added
without sequence numbers, then seqnum will be the line number where
the statement was found in the automation table member. The number
will be in the format #nnnnnnn.
```


Printed in Ireland

CT6EAEN