Trademarks

The following terms are registered trademarks of International Business Machines Corporation in the United States and/or other countries: AIX, AS/400, DB2, IBM, Micro Channel, MQSeries, Netfinity, NUMA-Q, OS/390, OS/400, Parallel Sysplex, PartnerLink, POWERparallel, RS/6000, S/390, Scalable POWERparallel Systems, Sequent, SP2, System/390, ThinkPad, WebSphere.

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries: DB2 Universal Database, DEEP BLUE, e-business (logo), ~, GigaProcessor, HACMP/6000, Intelligent Miner, iSeries, Network Station, NUMACenter, POWER2 Architecture, PowerPC 604, pSeries, Sequent (logo), SmoothStart, SP, xSeries, zSeries. A full list of U.S. trademarks owned by IBM may be found at http://iplswww.nas.ibm.com/wpts/trademarks/trademar.htm. NetView, Tivoli and TME are registered trademarks and TME Enterprise is a trademark of Tivoli Systems, Inc. in the United States and/or other countries.
Oracle, MetaLink are registered trademarks of Oracle Corporation in the USA and/or other countries.
Microsoft, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.
LINUS is a registered trademark of Linus Torvalds.
Intel and Pentium are registered trademarks and MMX, Pentium II Xeon and Pentium III Xeon are trademarks of Intel Corporation in the United States and/or other countries.
Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Other company, product and service names may be trademarks or service marks of others.
Introduction

Both IBM Tivoli NetView for z/OS and IBM Tivoli AF/Operator provide system automation capabilities. Certain functions such as message driven automation are common to both products. Other functions are unique – for example, NetView can detect and drive automation based on SNA alerts or TCP/IP traps, while AF/Operator can drive automation based on OMEGAMON performance thresholds.

If both products are installed, it can be useful to integrate the products and to allow them to exchange information. An environment may be using one of the products as the primary automation vehicle, but still wants to use the other in certain situations to gather additional information. This is the second of two papers covering the integration of the two products:

- Part 1 describes the architecture for, and provides an example of, NetView accessing AF/Operator to gather information (in this case, from OMEGAMON).
- Part 2 (this paper) will describe the architecture for, and provide an example of, AF/Operator accessing NetView to gather information.

Each of these accesses can be implemented independently of each other.

NOTE: The IBM Tivoli OMEGACENTER Gateway product provides a subset of AF/Operator functions; the information described in this document pertaining to NetView accessing AF/Operator also applies to the OMEGACENTER Gateway.

Integration Architecture

The following figure depicts the NetView and AF/Operator components that are used to integrate the product:
The NetView and AF/Operator address spaces must reside on the same z/OS image to use these functions. For both types of integration, NetView and AF/Operator must have access to each other’s load library. For example:

- The AF/Operator TKANMOD library, from the Run Time Environment (RTE) within which AF/Operator is configured, is added to the STEPLIB concatenation of the NetView started tasks JCL.
- The NetView CNMLINK library is added to the STEPLIB concatenation of the AF/Operator started task JCL.

The components within the “AF Operator Access to NetView” dashed box are discussed in this paper. The desired information to be captured from NetView will determine which of these components are necessary:

- For AF/Operator to issue commands to NetView, the Subsystem Interface (SSI) is used to send the command to NetView and receive a response.
- For AF/Operator to detect and trap VTAM messages or NetView console commands, one or more of the DSI exits will be required.

**Example: AF/Operator issuing commands to NetView**

AF/Operator can invoke NetView commands, CLISTs, or REXX programs via its OPER command. This command gives access to the z/OS console SSI. The NetView Subsystem Interface address space defines a Command Prefix character NetView will use to determine if a command on the SSI is meant for NetView.
The command prefix character is specified in the DSIG parameter of the address space JCL. For example if the JCL parameter specifies `DSIG='%'`, then NetView will receive all command strings that start with ‘%’.

For NetView to correctly execute the command, the following must be defined:

- If operator commands are protected by a SAF product, AF/Operator must have the authority to issue operator commands.
- There must be a match between the console name or number defined on the OPER command (via the CONSOLE parameter) and a console name or number assigned to a NetView autotask:
  - If any NetView autotask has a CONSOLE name of *ANY* assigned, the CONSOLE parameter does not have to be used on the OPER command – any value will match, and this autotask will execute the command.
  - If the CONSOLE parameter of OPER specifics a console number, and no NetView autotask has a CONSOLE name of *ANY*, there must be a NetView autotask with the same console number defined as its console.
  - If the CONSOLE parameter of OPER specifics a console name, and no NetView autotask has a CONSOLE name of *ANY*, the console name must match a name that can be generated from the AF/Operator CONSOLE startup parameter, using the PREFIX and LIMIT sub-parameters, and a NetView autotask must also have the same console name defined as its console.

The general flow of execution is:

1. An AF/Operator REXX program uses the OPER command to issue a command string with the NetView prefix character. The RESP keyword must be used to have the response returned into a set of variables.
2. The command string goes to the SSI, where subsystems listening on the SSI check the prefix character.
3. Since the prefix character matches the NetView prefix character, NetView takes the command string off of the SSI.
4. If there is an autotask with the matching console name or number, or with *ANY* specified if no match is found, the autotask executes the command.
5. The command response is returned to the invoking AF/Operator program, which can analyze the command response.

Matching the AF/Operator and NetView Console Definitions

In the example environment AF/Operator has the following CONSOLE definition in its startup parameters:

```
CONSOLE(TYPE(EMCS),LIMIT(6),
PREFIX(AFOP),KEY(AFOP)),
```

With a prefix value of AFOP and a limit value of 6, default EMCS console names AFOP0001 – AFOP0006 will be generated.

In NetView autotask AUTOAFO will be used to execute commands sent from AF/Operator. The console name for AUTOAFO can be defined in either of the following 2 ways:
1. *ANY* (AUTOTASK OPID=AUTOAFO, CONSOLE=*ANY*).
2. AFOPxxxx (AUTOTASK OPID=AUTOAFO, CONSOLE=AFOPxxxx), where xxxx is in the range 0001 through 0006. If this option is used the OPER command must use the same name in its CONSOLE parameter.

Creating an AF/OPERATOR REXX Program

The following REXX program queries NetView for its five highest CPU consuming tasks:

```rexx
/* REXX */
/* SEND PIPE COMMAND TO NETVIEW - RETRIEVE TOP 5 TASKS */
"OPER '%'PIPE NETVIEW TASKUTIL |",
"SEPARATE | TAKE 8 | CONSOLE' RESP CONSOLE(AFOP0001)"
RC=GLBVGET("LINE#")
DO I=1 TO LINE#
   RC=GLBVGET('LINE'I)
   SAY "LINE" I": " VALUE('LINE'I)
END
EXIT
```

The OPER command invokes the NetView PIPE command to retrieve the top 5 NetView CPU users as reported by the NETVIEW TASKUTIL command. The RESP parameter causes the results to be returned into the LINE# variable. The CONSOLE parameter is set to AFOP0001, so the AUTOAFO autotask console is set to AFOP0001. After getting the response the results are displayed in the AF/Operator job log.

The REXX program is invoked using the AF/Operator EX command from a z/OS console. The AF/Operator job log shows the results:

```
EX AFO2NV2
OPER '%'PIPE NETVIEW TASKUTIL | SEPARATE | TAKE 8 | CONSOLE' RESP CONSOLE(AFOP0001)
LINE 1: %PIPE NETVIEW TASKUTIL | SEPARATE | TAKE 8 | CONSOLE
LINE 2: DWO022I
LINE 3: TASKNAME TYPE DPR CPU-TIME N-CPU% S-CPU% MESSAGEQ STORAGE-K CMD
LINE 4: -------- ---- --- ----------- ------ ------ -------- --------- --------
LINE 5: AUTO1 AUTO 250 20618.61 79.96 0.13 0 140 **NONE**
LINE 6: DSITIMMT OPT 255 1305.94 3.64 0.01 N/A 37 N/A
LINE 7: DSIMONIT OPT 255 832.25 3.50 0.01 N/A 4 N/A
LINE 8: AUTOAFO AUTO 250 0.11 1.90 0.00 0 110 PIPE
LINE 9: DSILog DST 254 182.51 1.25 0.00 0 73 N/A
```

In the NetView, log, AUTOAFO is shown as the autotask executing the PIPE request:

```
AUTOAFO CNM19 14:19:56 | PIPE NETVIEW TASKUTIL | SEPARATE | TAKE 8
AUTOAFO CNM19 14:19:58 ' DWO022I
AUTOAFO CNM19 14:19:58 ' TASKNAME TYPE DPR CPU-TIME N-CPU% S-CPU
AUTOAFO CNM19 14:19:58 ' --------------------- ----- ----------- ------- -------
AUTOAFO CNM19 14:19:58 ' AUTO1 AUTO 250 20618.61 79.96 0.1
AUTOAFO CNM19 14:19:58 ' DSITIMMT OPT 255 1305.94 3.64 0.0
AUTOAFO CNM19 14:19:58 ' DSIMONIT OPT 255 832.25 3.50 0.0
AUTOAFO CNM19 14:19:58 ' AUTOAFO AUTO 250 0.11 1.90 0.0
```
This example is summarized in this picture:

```
EX AFO2NV2
/* REXX */
"* SEND PIPE COMMAND TO NETVIEW */
"OPER PIPE NETVIEW TASKUTIL |*,
"SEPARATE | TAKE 8 | CONSOLE' RESP "
RC=GLBVGET("LINE#")
DO I=1 TO LINE#
  RC=GLBVGET('LINE'I)
  SAY "LINE" I": " VALUE('LINE'I)
END
EXIT
```

**Operational Considerations**

- Only NetView command line commands (i.e., non-interactive or not using WINDOW) can be invoked in this manner.
- Only one NetView autotask can have a Console of %ANY% assigned. This means that requests received by it are processed serially. If multiple AF/Operator requests to NetView are to run in parallel, multiple autotasks each with a specific console name as describe above will be required.

**Trapping VTAM Messages and NetView Commands**

AF/Operator provides exits for use with these NetView exit points:

- The provided DSIEX01 exit will capture NetView NCCF commands and allow AF/Operator CMD traps to act upon them.
- The provided DSIEX06 exit will capture VTAM solicited messages (responses to commands issued by a user or the NetView PPT task) and allow AF/Operator WTO traps to act upon them.
- The provided DSIEX11 exit will capture VTAM unsolicited messages and allow AF/Operator WTO traps to act upon them.
During AF/Operation configuration, these exits can be enabled using the Installation and Configuration Assistance Tool. After selecting AF/Operator as the product to configure, select **Install Optional Features->NetView Interface->Install NetView Exits**, and select the target system name to display the following panel:

```
V340 --------------------- INSTALL NETVIEW EXITS (MVS1) ---------------------
COMMAND ===>

DSIEX01 - NetView exit for NetView console commands
DSIEX06 - NetView exit for solicited VTAM messages
DSIEX11 - NetView exit for unsolicited VTAM messages

Specify your NetView exit installation choices.
Include user exit along with AO/MVS NetView exit AODSIX01 ===> N
Include user exit along with AO/MVS NetView exit AODSIX06 ===> N
Include user exit along with AO/MVS NetView exit AODSIX11 ===> N

NETVIEW LIBRARIES NEEDED:
MACRO LIBRARIES        === NV390.V5R1.SCNMAG1____________
                         === ________________
NETVIEW LOAD LIBRARY    === NV390.V5R1.CNMLINK________
USER EXIT LIBRARY       === ________________

Press Enter to generate a batch job that installs the NetView exits.
F1=Help   F3=End
```

If an existing DSIEX exit is already in place, change the “Include user exit” option to Y for the corresponding exit, and add the name of the user exit library where the exits reside. This will cause the existing exit to be included with the AF/Operator provided exit, so that there is no loss of function.

From this panel a job is generated to compile the exits and place them in a Load Library. The exits are enabled by placing them in a NetView STEPLIB library, and use the LOADEXIT command in CNMSTYLE (for NetView V1R4 and later) to load them when NetView starts. Once loaded, CMD and WTO traps can be used for these commands and messages just as they are used for other commands and messages.

**Summary**

The combination of NetView and AF/Operator can be very beneficial due to their complimentary functions. The available interfaces allow a bi-directional exchange of information, which provides added flexibility when designing automation solutions requiring this level of information integration.

**References**

- **Tivoli NetView for z/OS Installation: Getting Started (SC31-8872)** describes the base installation and configuration of NetView.
- **Tivoli NetView for z/OS Users Guide (GC31-8849)** contains information on defining autotasks.
- **Tivoli AF/Operator Configuration and Customization Guide (GC32-9138)** - provides a planning worksheet and describes the system level configuration changes for NetView and AF/Operator to enable the interface.
- **Tivoli AF/Operator Users Guide (GC32-9142)** – provides information on how to control the interface and how to manage the information flow between NetView and AF/Operator