EView/390z Management for HP Operations Manager for MS Windows

Installation Guide

Software Version: 6.4

August 2013

Legal Notices

Warranty

EView Technology makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. EView Technology shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

EView Technology, Inc.
4909 Green Road
Raleigh, North Carolina 27616
United States of America

Copyright Notices

Trademark Notices

EView/390® is a registered trademark of EView Technology, Inc.

S/390, OS/390, z/OS, and zSeries are trademarks of International Business Machines Corporation.

Microsoft®, Windows®, and Windows NT® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of the Open Group.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Table of Contents

1

Conventions

Introduction

The following typographical conventions are used in this manual.

Table 1-1: Typographical Conventions

Font	Meaning	Example
Italic	Book or manual titles, and man page names	See the *EView/390z Discovery for HP UCMDB Installation Guide* for more information.
	Provides emphasis	You *must* follow these steps.
	Specifies a variable that you must supply when entering a command	At the prompt, enter rlogin *your_name* where you supply your login name.
	Parameters to a function	The *oper_name* parameter returns an integer response.
Bold	New terms	The monitor **agent** observes...
`Computer`	Text and items on the computer screen	The system replies: `Press Enter`
	Command names	Use the `dir` command ...
	Function names	Use the `opc_connect()` function to connect...
	File and directory names	`C:\windows\`
	Process names	Check to see if `opcmona` is running.
	Window/dialog box names	In the `Add Logfile` window...
`Computer Bold`	Text that you must enter	At the prompt, enter **`dir`**
Keycap	Keyboard keys	Press **Return** .

Font	Meaning	Example
[Button]	Buttons on the user interface.	Click [Operator] . Click the [Apply] button.
Menu Items	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows.	Select Actions:Utilities ->Reports ...

2

Documentation Map

EView/390z Management for z/OS (EView/390) provides a set of manuals that help you use the product and understand the concepts underlying the product. This section describes what information is available and where you can find it.

➤ In addition to EView/390 documentation, related HP Operations Manager products provide a comprehensive set of manuals that help you use the products and improve your understanding of the underlying OMW concepts.

EView/390z Printed Manuals

This section provides an overview of the printed manuals and their contents.

EView/390z Management for OM Windows Concepts Guide

Explains EView/390z features, functions, architecture, and data flow. Describes EView/390z agent and server components, process management, Operations Manager Windows (OMW) integration and message windows.

EView/390z Management for HP Operations Manager for MS Windows Installation Guide

Explains how to install, de-install, and configure EView/390z. Also includes how to upload installation files from the OM management server, update EView/390z, and start and stop EView/390z processes.

EView/390z Management for HP Operations Manager for MS Windows Administrator's Reference

Explains how to customize and use EView/390z. Also includes troubleshooting procedures and explanations of EView/390z system messages.

EView/390z Online Information

The following information is available online:

- *EView/390z Management for OM Windows Product Brief*
- *EView/390z Management for HP Operations Manager for MS Windows Installation Guide*
- *EView/390z Management for HP Operations Manager for MS Windows Administrator's Reference*

3

Installing and De-installing EView/390z

This chapter describes how to install and de-install EView/390 Management for OM Windows (EView/390).

EView/390z consists of two components. The “Server” component is installed on an HP OM server that also has the OM Agent. The “Agent” component is installed on each z/OS operating system partition (LPAR).

The EView/390z software is installed first on the OM server and includes the Agent software files which are then transferred to the z/OS partitions for installation.

Installation Requirements

This section describes the operating system, hardware, and software requirements for installing EView/390z software. To avoid problems during installation, read this section before you start the installation process.

Hardware Requirements

- OMW Management Server

EView/390z requires appropriate Ethernet hardware on the OMW Management Server and OMW Console to communicate via TCP/IP.

All other hardware requirements are the same as the requirements for HP Operations Manager for Windows.

- S/390 Managed Node

EView/390z requires the appropriate Ethernet hardware on the S/390 mainframe to allow for TCP/IP communication with the OMW server.

In addition, make sure that OMW management server, OMW console, and S/390 managed nodes meet the disk space requirements described in Table 3-1.

Table 3-1: Additional Disk-Space Requirements

Machine	Disk Space
OMW Management Server	25MB
OMW Console	5 MB
S/390- Mainframe	60 tracks of 3390 DASD

Software Requirements

- On the OMW Management Server and Console:
  - One of the following:
    - ○ Microsoft Windows 2003 Server or higher AND
      HP Operations Manager for Windows Version 8.x.
    - ○ Microsoft Windows 2008 R2 (64 bit) or higher AND
      HP Operations Manager for Windows Version A.09.00.
  - The TCP/IP network protocol stack must be active.
  - For EView/390z tools which run on the OMW Console, Java Plug-in Version 1.4 or higher must be installed (available from http://java.sun.com/products/plugin).

All other software requirements are the same as the requirements for HP Operations Manager for Windows.

- On the z/OS managed node:
  - z/OS V1R3 or higher.
  - The TCP/IP network protocol stack (V3R1 or higher) must be active.

Obtaining License Keys

EView/390z requires a license key to be applied to the configuration of each z/OS system that is to be managed by the OMW server. A license is required for each physical mainframe machine; the same license key may be used for multiple LPARs on the same physical system. Contact EView Technology at +1-919-878-5199 or e-mail support@eview-tech.com to get the necessary license keys. Be prepared to give the serial number of the S/390 system (or first logical serial number in a multi-CPU LPAR system). The serial number can be found by issuing a `DISPLAY M=CPU` command from a z/OS console and reading the last five characters of the CPC ND line of the output.

For example, the following output shows that the serial number of this particular S/390 system is 70571:

DISPLAY M=CPU

```
IEE174I 06.00.00 DISPLAY M
PROCESSOR STATUS
ID  CPU                  SERIAL
0    +                   0105717060
1    +                   1105717060
2    +                   2105717060
3    +                   3105717060

CPC ND = 007060.H30.IBM.02.000000070571
CPC SI = 7060.H30.IBM.02.0000000000070571
CPC ID = 00
```

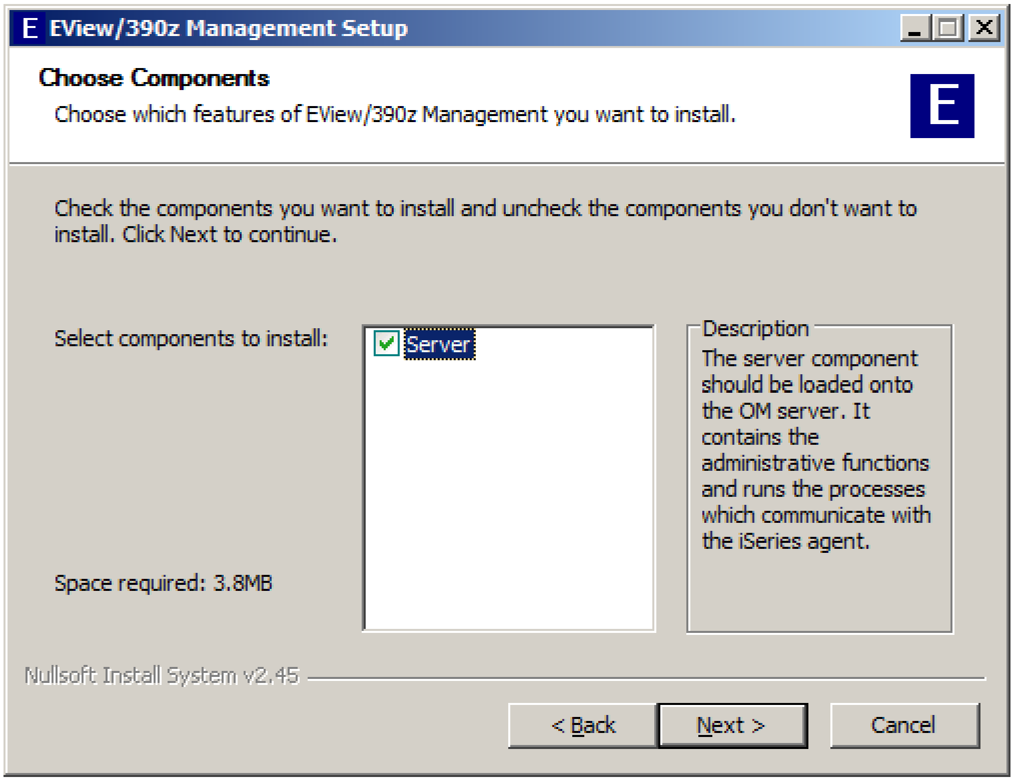
Installing EView/390z on an OMW Management Server or Console

The EView/390z installation program can be run to install the EView/390z server component, the EView/390z client component, or both.

Installation Steps

1. Insert the EView/390z installation CD into the CD drive of the OMW management server or OMW console.
2. If the setup wizard does not automatically start, go to the top level directory on the installation CD and double-click `Eview390ZInstall.exe`.
3. Select the Server checkbox in the setup wizard. The software should only be installed on the OMW server.

Figure 3-1: EView/390z Installation



Choose Components

Choose which features of EView/390z Management you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install: ☑ Server

Description

The server component should be loaded onto the OM server. It contains the administrative functions and runs the processes which communicate with the iSeries agent.

Space required: 3.8MB

Installed File Locations on the Management Server

The installation process copies the necessary files to the OMW management server in the directory path you specified. The default path for EView/390 files is:

```
\Program Files\EView Technology\EView 390\
```

The installation process also defines an “EV390” virtual directory pointer in IIS to point to the installed Java class files.

Installing EView/390z on the Managed Nodes

To install the EView/390z agent on the managed nodes, use the File Transfer Protocol (FTP) to upload the EView/390z datasets to all z/OS mainframe LPARs that are to be set up as OMW managed nodes.

What to Upload

After installing EView/390z on the management server, the mainframe datasets are located in the following directory:

```
\Program Files\EView Technology\EView 390\mf
```

This directory contains the EView/390z files for uploading to the z/OS mainframe, as shown in Table 3-2. Identify which files are needed, based on the listed requirements.

Table 3-2: EView/390z Files to Upload to the z/OS Mainframe

File Name	Description	3390 DASD Tracks	Required?
EV390.V64.LOAD.SEQ	EView/390 agent executables	35	Yes
EV390.V64.SAMP.SEQ	Sample JCL, startup parameter cards, and VTAM definitions	8	Yes
EV390.V64.CLIST.SEQ	Command lists used by NetView/390 and SOLVE:NETMASTER for executing mainframe commands	5	Only if NetView/390 or SOLVE: NETMASTER is installed
EV390.V64.ASM.SEQ	Assembler source code for onsite assembly of NetView/390 and CICS exits	8	Only if NetView/390 or CICS is installed

Transferring Files to the Mainframe

Use FTP to send the several files from the OMW server to the z/OS system. Use binary mode when transmitting the files, and use the SITE or LOCSITE command to force ftp to create the target datasets with attributes DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) and a primary allocation of 35 DASD tracks. Consult your mainframe systems programmer for the appropriate dataset high-level qualifier (hlq) name for the files as they are transferred to the S/390:

```
C:> cd Program Files\EView Technology\EView 390\mf
C:> ftp s390name
User: username
Password: ****
ftp> bin
ftp> quote site blksize=3120
ftp> quote site lrecl=80
ftp> quote site recfm=fb
ftp> quote site primary=35
ftp> put EV390.V64.LOAD.SEQ hlq.EV390.V64.LOAD.SEQ
ftp> put EV390.V64.SAMP.SEQ hlq.EV390.V64.SAMP.SEQ
ftp> put EV390.V64.CLIST.SEQ hlq.EV390.V64.CLIST.SEQ
ftp> put EV390.V64.ASM.SEQ hlq.EV390.V64.ASM.SEQ
ftp> quit
```

If you receive a B37 or D37 "out of space" error from any of the `put` commands, you may need to pre-allocate the sequential file on the mainframe using the sizes given in Table 3-2.

Extracting Partitioned Datasets from Sequential Datasets

After uploading the files to sequential datasets on the mainframe, use the TSO RECEIVE command to extract a partitioned dataset (PDS) from each of the sequential datasets.

From a TSO command line, enter the following command for each of the uploaded datasets:

```
RECEIVE INDS('hlq.dataset.SEQ')
```

The RECEIVE command will prompt you for additional restore parameters. The output PDS name can be modified at this time by entering the DA parameter. For example, to change the SAMP dataset HLQ to “EV390” enter the following:

```
DA('EV390.V64.SAMP')
```

It is recommended that the EView/390z datasets contain the version number.

De-installing EView/390z

This section describes how to remove EView/390z software from the following:

- OMW GUI
- OMW management server
- z/OS managed nodes

To Remove EView/390z Components from the OMW GUI

Follow these steps to manually remove EView/390z components from the OMW GUI:

1. Open the EView/390z Task Manager and stop any running EView/390z processes, then click each defined node and select the "Delete Node" button.
2. From the OMW server, delete any z/OS managed nodes from the Nodes subtree, and the "Mainframe" node group.
3. Delete the "Mainframe Tools" group from the Tools subtree.
4. Delete the "Mainframe" policy group from the Policy Management subtree, and re-deploy the remaining policies to the OMW server node.
5. Remove any log files created during the EView/390z run:

   ```
   cd <EV390_install_directory>\log
   ```

   ```
   del *.*
   ```

6. Remove any node configuration files created during the EView/390z run:

   ```
   cd <EV390_install_directory>\conf
   ```

   ```
   del ev390_config*
   ```

To Remove EView/390z Components from the OMW Server

Use the “Add/Remove Programs” utility from Windows Control Panel to remove EView/390 files, registry entries, and the EView/390 virtual directory in IIS.

To Remove EView/390z from the OMW Managed Nodes

To remove EView/390z from the managed nodes, follow these steps:

1. Stop the EView/390z task on the z/OS managed node.

   To find out how to stop the EView/390z job on the managed nodes, see “*Running EView/390z as a Started Task”* in Chapter 8.

2. Delete the EView/390z datasets installed on the z/OS managed nodes.

4

Updating Mainframe Software

This chapter contains instructions for updating z/OS resources on the mainframe, and updating the EView/390z input parameter cards to customize the mainframe task for the particular needs of your site.

Phase 1: Updating TCP/IP Connectivity

The following modifications need to be made to the IBM TCP/IP:

- Reserve port numbers in `PROFILE.TCPIP`
- Identify TCP/IP high-level quantifier

Reserving Port Numbers in PROFILE.TCPIP

Choose two available port numbers for use by EView/390 and identify them in the list of `PORT` values in the `PROFILE.TCPIP` dataset:

```
6106 TCP VP390
6107 TCP VP390
```

(The default ports used by EView/390 are 6106 and 6107. The default job name for EView/390 is "VP390".)

➤ This step is optional. If specific port numbers are not reserved for EView/390 use, the EView/390z client connection will still succeed, but this reservation will flag the chosen port numbers for exclusive use by EView/390 so they are not used by other products on the mainframe.

Identifying the active TCPIP.DATA file

Make note of the dataset/member location of the active TCPIP.DATA file for the TCP/IP stack that EView/390 will be connecting to. This dataset name will be needed in Chapter 6 when defining the `SYSTCPD` DD card in the startup JCL job.

Phase 2: Updating z/OS and VTAM

To run EView/390z, modify the z/OS datasets as follows:

- Authorize the *`hlq`*`.LOAD` dataset
- Set the performance group or add an entry to Workload Manager
- Add an entry to the Program Properties Table
- Add an entry to the RACF class
- Add `PPOLOG` to VTAM startup options
- Add an application major node to VTAM

Inform the mainframe system programmer of changes needed to the `SYS1.PARMLIB` members.

Authorizing the *hlq*.LOAD Dataset

Add the EView/390z `hlq.LOAD` dataset and its DASD volume name to the list of APF authorized datasets in one of the following:

- `SYS1.PARMLIB(IEAAPFxx)`
- `SYS1.PARMLIB(PROGxx)`

This addition is required to allow EView/390z to process certain authorized commands and perform security checks.

The authorization added to `SYS1.PARMLIB` takes effect after the next IPL. To dynamically authorize the `hlq.LOAD` dataset on DASD volume `volser` without an IPL, enter the following z/OS command:

```
SETPROG APF,ADD,DSNAME=hlq.LOAD,VOLUME=volser
```

Setting the Performance Group or Service Class

If Workload Manager (WLM) is present on the S/390 system, add an entry for the VP390 job to the SYSTEM or SYSSTC service class, assigning it a priority slightly lower than VTAM.

If WLM is not used, set the performance group by adding a `TRXNAME` parameter for EView/390 to the STC subsystem definition of `SYS1.PARMLIB(IEAICSxx).`

In the `TRXNAME` line, specify one of the following:

- Same performance group used by NetView/390 (if present)
- Performance group that is one level below the VTAM performance group

This addition ensures that EView/390z receives enough CPU time to avoid a backlog of network information processing. The default name for the EView/390 startup job is `VP390`.

For example, if NetView/390 is running in performance group `8`, specify the addition for VP390 with the following:

```
TRXNAME=VP390,PGN=8
```

To dynamically reload the ICS file after a new entry is added, enter the following z/OS command:

```
SET ICS=xx
```

where *xx* is the two-digit suffix of the edited member.

Adding an Entry to the Program Properties Table

Add a PPT entry to the `SYS1.PARMLIB(SCHEDxx)` for VP390, identifying the started task as a non-swappable, non-timed system task. This addition ensures that the VP390 address space is not swapped and that the job is not terminated when no network activity occurs.

The syntax for the PPT entry is as follow:

```
PPT PGMNAME(VP390)
NOSWAP
SYST
```

To dynamically reload the PPT after a new entry is added, enter the following z/OS command:

```
SET SCH=xx
```

where *xx* is the two-digit suffix of the edited member.

Adding an Entry to the RACF Class

The VP390 task requires a z/OS UNIX System Services (USS) segment. Because USS segments are associated with RACF-defined user IDs, you should add an identifying entry for VP390 to a RACF class to meet the USS requirement.

This addition allows the VP390 to run as a started task. If the VP390 is to be run as a submitted job, enter the user ID on the `JOB` card of the startup job.

To add an entry to the RACF class, follow these steps:

1. Verify that the `STARTED` class is defined by entering the following command:

   ```
   RLIST STARTED *
   ```

   This command displays a list of entries for the `STARTED` class.

2. Determine whether a RACF user (for example, `IBMUSER`) has an OMVS segment by entering the following command:

   ```
   LU IBMUSER OMVS
   ```

3. If the `STARTED` class is activated, add the VP390 task to the defined user (for example, `IBMUSER`) by entering the following:

   ```
   RDEFINE STARTED VP390.VP390 STDATA(USER(IBMUSER) GROUP(SYS1))
   ```

   Then refresh the class by entering the following:

   ```
   SETROPTS RACLIST(STARTED) REFRESH
   ```

4. If the `STARTED` class is not activated, assign RACF identities to the started procedures.

   Incorporate the following sample into the ICHRIN03 job of `SYS1.SAMPLIB(RACTABLE)`

   Example:

   ```
   ICHRINO3 CSECT
   COUNT    DC    AL2(((ENDRINO3-COUNT-2)/32)+32768)

   *-------New VP390 Entry-----------------

   ENTRY1   EQU   *
   PROC1    DC    CL8'VP390   '
   USERID   DC    CL8'IBMUSER '
   GROUP1   DC    CL8'SYS1    '
   FLAGS1   DC    XLI'00'
            DC    XL7'00'

   *-------Last Entry----------------------
   ```

```
ENTRY2   EQU   *
PROC2    DC    CL8'*       '
USERID2  DC    CL8'IBMUSER '
GROUP2   DC    CL8'=       '
FLAG2    DC    XLI'00'
ENDRINO3 EQU   *
END
```

RACF allows the started procedures table to contain a generic entry, indicated by an asterisk (*) in the procedure-name field. When searching the table for a procedure-name match, if RACF finds a procedure name of "*" as the last entry in the table and the procedure name was not specifically matched by any other entry in the table, RACF uses the "*" entry as a match for the procedure. This procedure is documented in *the IBM Security Server (RACF) System Programmer's Guide*.

Allow Viewing of All JES2 Jobs

If you intend to use the EView/390z Mainframe Visual Management Interface (MVMI) Java application to access the list of all jobs on the JES2 Input, Output, and Held queues (regardless of job owner), follow one of the following two options, depending on your mainframe configuration:

a. If RACF is being used for SDSF security, use the following commands to allow the user ID that VP390 is running under to access the JES2 queues:

```
PERMIT ISFCMD.DSP.ACTIVE.JES2 CLASS(SDSF) ID(userid) ACC(READ)
PERMIT ISFCMD.DSP.HELD.JES2 CLASS(SDSF) ID(userid) ACC(READ)
PERMIT ISFCMD.DSP.INPUT.JES2 CLASS(SDSF) ID(userid) ACC(READ)
PERMIT ISFCMD.DSP.OUTPUT.JES2 CLASS(SDSF) ID(userid) ACC(READ)
PERMIT ISFCMD.FILTER.* CLASS(SDSF) ID(userid) ACC(READ)
```

replacing *`userid`* with the RACF user ID that the VP390 job is started under (which was named in the `RDEFINE` statement above). The ISFCMD class must be active.

b. If the SDSF server address space is running, edit SYS1.PARMLIB(ISFPRMxx) and add another GROUP ahead of the default ISFUSER group. The new GROUP should have the same attributes as the default ISFUSER group, with the following exceptions:

- A new NAME
- Authorize the functions I, O, H, DA, ST, SE, and PREF
- Allow ALL browse authority
- An IUID parameter to restrict the group to only be used by a specific user ID

The following sample shows which fields are changed from the ISFUSER profile:

```
/*******************************************************************/
/*GROUP ISFVP390 - VP390 Group with expanded job viewing capability*/
/*******************************************************************/
GROUP NAME(ISFVP390),       /* Group name                <-CHANGED */
TSOAUTH(JCL),               /* User must have JCL                  */
ACTION(11,12,USER),         /* Default route codes in log          */
ACTIONBAR(YES),             /* Display action bar on panels        */
APPC(ON),                   /* Include APPC sysout                 */
AUPDT(10),                  /* Default auto update interval        */
AUTH(I,O,H,DA,ST,SE,PREF),  /* Authorized functions      <-CHANGED */
CMDAUTH(USERID,NOTIFY),     /* Command authority                   */
CMDLEV(2),                  /* Command level                       */
CONFIRM(ON),                /* Enable cancel confirmation          */
CURSOR(ON),                 /* Leave cursor on last row processed  */
DADFLT(IN,OUT,TRANS,STC,TSU,JOB),  /* Default rows on DA           */
DATE(MMDDYYYY),             /* Default date format                 */
DATESEP('/'),               /* Default datesep format              */
DISPLAY(OFF),               /* Do not display current values       */
DSPAUTH(ALL),               /* Browse authority          <-CHANGED */
ILOGCOL(1),                 /* Initial display column in log       */
LANG(ENGLISH),              /* Default language                    */
LOGOPT(OPERACT),            /* Default log option                  */
OWNER(NONE),                /* Default owner             <-CHANGED */
/*PREFIX(USERID),              Default prefix            <-REMOVED */
UPCTAB(TRTAB2),             /* Upper case translate table name     */
VALTAB(TRTAB),              /* Valid character translate table     */
IUID(VP390USR),     /* Only for userids in the VP390USR list <-NEW */
VIO(SYSALLDA)               /* Unit name for page mode output      */
```

Following all of the GROUP entries, create a new Name Table using the same name used in the IUID field above:

```
NTBL NAME(VP390USR)
  NTBLENT STRING(userid)
```

where `userid` is the RACF user ID that VP390 is started under (from the RDEFINE command in the previous step).

After making these additions, save the `ISFPRMxx` member and refresh the SDSF server with the console command:

```
MODIFY SDSF,REFRESH
```

Allow MQ Commands

If commands are to be issued to IBM WebSphere MQ (see command option 50 in the *EView/390z Administrator's Reference*), and RACF security is active for WebSphere MQ, the following steps must be done depending on the RACF MQ classes that are active:

1. If the MQCONN class is active, the EView agent user must have access to the ***mqssid***.BATCH profile (where ***mqssid*** is the MQ subsystem name):

   ```
   PERMIT mqssid.BATCH CLASS(MQCONN) ID(agent-user) ACCESS(READ)
   ```

2. If the MQQUEUE class is active for WebSphere MQ, the agent user must have update access to the SYSTEM.COMMAND.REPLY.MODEL queue, for example:

   ```
   PERMIT mqssid.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(agent-
   user) ACCESS(UPDATE)
   ```

   and the command input queue:

   ```
   PERMIT mqssid.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(agent-user)
   ACCESS(UPDATE)
   ```

   A profile for the EView/390z command reply queue must be created, for example:

   ```
   RDEFINE mqssid.EVIEW.COMMAND.REPLY.Q UACC(NONE)
   ```

   and the agent user given Alter access to create and delete this queue:

   ```
   PERMIT mqssid.EVIEW.COMMAND.REPLY.Q CLASS(MQQUEUE) ID(agent-user)
   ACCESS(ALTER)
   ```

   (These queue names can be customized using SYSIN cards described on page 35.)

3. If the MQCMDS class is active, the agent user must be given access to the desired MQ commands, for example:

   ```
   PERMIT mqssid.DISPLAY.SYSTEM  CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   PERMIT mqssid.DISPLAY.GROUP   CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   PERMIT mqssid.DISPLAY.QUEUE   CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   PERMIT mqssid.DISPLAY.QMGR    CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   PERMIT mqssid.DISPLAY.CHANNEL CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   PERMIT mqssid.DISPLAY.CHINIT  CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   PERMIT mqssid.DISPLAY.GROUP   CLASS(MQCMDS) ID(agent-user) ACCESS(READ)
   ```

4. The EView/390z startup JCL must be updated to include the MQ SCSQAUTH dataset in the STEPLIB concatenation.

Adding PPOLOG to VTAM Startup Options

To ensure that VTAM messages are sent to the Primary Program Operator (PPO) in response to console commands, add the following to your VTAM startup options in `SYS1.VTAMLST(ATCSTRxx)`:

```
PPOLOG=YES
```

If the `PPOLOG` parameter is not set in the currently running VTAM, add it dynamically with the following z/OS command:

```
MODIFY vtamproc,VTAMOPTS,PPOLOG=YES
```

where ***vtamproc*** is the name of the VTAM started task.

Adding the *hlq*.SAMP(EVAPPL) Definition

To add the `hlq.SAMP(EVAPPL)` application major node definition, follow these steps:

1. Copy the following application major node definition from the EView/390z SAMP dataset into your VTAMLST library:

   `hlq.SAMP(EVAPPL)`

2. If NetView/390 or SOLVE:NETMASTER is running in this z/OS LPAR, comment out the PPO definition in the `EVAPPL`.

3. Add `EVAPPL` to the list of major nodes in the following:

   `SYS1.VTAMLST(ATCCONxx)`

4. Activate the major node and verify that the APPL definitions are active:

   ```
   VARY NET,ACT,ID=EVAPPL,SCOPE=ALL
   DISPLAY NET,ID=EVAPPL,E
   ```

   You can modify the APPL resource names in the definition to accommodate your network resources naming conventions. Changes made to these default names must also be made to the parameter cards, defined in the next section.

Phase 3: Updating EView/390z Parameter Cards

Modify initialization parameter cards to match the resources you have configured for EView/390z and OMW. The parameter card dataset member(s) are pointed to by the SYSIN DD of the started task.

The EView/390z parameter cards are located in:

`hlq.SAMP(VPOPARM)`

Parameter cards may contain system symbols to ensure uniqueness if the same dataset member is being used for multiple LPARs. Symbols must start with an ampersand (&) and must end with a period if the symbol substitution occurs in the middle of a string. For example, to incorporate the value of &SYSCLONE. into the name of the console defined for the CMD subtask, the parameter card would look like:

```
CMD &SYSCLONE.CON
```

➤ Strings resulting from symbol substitution must conform to the length and syntax requirements of the parameter.

A detailed description of the parameter cards follows.

CMD Parameter Card

Provides EView/390z with the ability to issue z/OS (MVS) commands.

Valid Values

```
consname [LOG|NOLOG] [HC={YES|NO}]
```

Sample Syntax

```
CMD EVOCONS2 NOLOG HC=YES
```

Description

This card will initialize the `CMD` subtask and define an extended MCS software console, which is used to send z/OS (MVS) commands from EView/390z to the mainframe OS.

Parameters

consname	Required. Specify a 1-8 character name for the extended MCS console you wish to define for issuing z/OS (MVS) commands. If this name is defined in RACF, the OPERPARM values in the RACF entry for this name are used for the console definition. Otherwise, a console is defined with default parameters AUTH=MASTER and ROUTCDE=NONE. (See Appendix A for information on defining an extended console to RACF.)
`LOG|NOLOG`	Optional. Specify `LOG` to force a system log message to be written for all z/OS commands entered from an EView/390z server. The commands are recorded in the system log with an EVO033 message. Specifying `NOLOG` here suppresses the writing of the EVO033 message. `NOLOG` is the default.

`HC={YES|NO}`	Optional. Specify `HC=YES` to record all commands and responses from this extended console in the mainframe hardcopy log. `HC=NO` will prevent any hardcopy logging of the commands that are sent in from the EView/390z client. `HC=NO` is the default

DELAY Parameter Card

Number of seconds to wait until the next attempt to restart a subtask.

Valid Values

`1 to 86400` (seconds)

Sample Syntax

```
DELAY 45
```

Description

Specifies the amount of time (in seconds) before a subtask attempts restarting itself following a termination. The maximum delay time allowed is 86,400 seconds (one day). Each subtask parameter card can be coded with its own unique delay time. Customize any subtask by entering in the desired `DELAY` card immediately before the subtask card. Any `DELAY` value entered becomes the default for all subsequent subtask cards.

The delay time reflects how quickly a needed resource can be recovered. A `TCP` subtask may require time to reset the port through which the OMW server is connected. The default `DELAY` value is 30 seconds.

DROP_AUTOMATION_FLAGGED Parameter Card

Indicate whether to drop MVS messages with the “Automation Requested” flag set.

Valid Parameters

`{YES|NO} [LOG]`

Sample Syntax

```
DROP_AUTOMATION_FLAGGED YES LOG
```

Description

The DROP_AUTOMATION_FLAGGED parameter card is used to capture or drop z/OS messages with the “Automation Requested” bit flag set.

Parameters

`YES	NO`	"YES" indicates that a message with the Automation Requested flag will be suppressed. "NO" indicates that the message will not be automatically suppressed, but the message ID must still be present in the message filter table to be passed on to the OMW server. "NO" is the default.
`LOG`	Optional. Add the LOG parameter to force any suppressed messages to be written to the NOMATCH log. (This parameter is only relevant if the NOMATCH logging subtask has been defined and is active.)

DROP_HARDCOPY_ONLY Parameter Card

Indicate whether to drop MVS messages with the "Hardcopy Only" flag set.

Valid Values

```
{YES|NO} [LOG]
```

Sample Syntax

```
DROP_HARDCOPY_ONLY NO LOG
```

Description

The DROP_HARDCOPY_ONLY parameter card is used to capture or drop MVS messages with the "Hardcopy Only" bit flag set. This parameter is only relevant when the HC=YES option is used on the MVS parameter card.)

Parameters

`YES	NO`	"YES" indicates that a message with the Hardcopy Only flag will be suppressed. "NO" indicates that the message will not be automatically suppressed, but the message ID must still be present in the message filter table to be passed on to the OMW server. "YES" is the default.
`LOG`	Optional. Add the `LOG` parameter to force any suppressed messages to be written to the NOMATCH log. (This parameter is only relevant if the `NOMATCH` logging subtask has been defined and is active.)

DROP_MPF_SUPPRESSED Parameter Card

Indicate whether to drop MVS messages with the "MPF Suppressed" flag set.

Valid Values

```
{YES|NO} [LOG]
```

Sample Syntax

```
DROP_MPF_SUPPRESSED YES LOG
```

Description

The DROP_MPF_SUPPRESSED parameter card is used to capture or drop MVS messages with the "Message Suppressed by MPF" bit flag set.

Parameters

`YES	NO`	“YES” indicates that a message with the "MPF Suppressed" flag will be suppressed. (This can also be accomplished by setting the "`AUTO=NO`" option on the MVS subtask definition card.) “NO” indicates that the message will not be automatically suppressed, but the message ID must still be present in the message filter table to be passed on to the OMW server. “NO” is the default.
`LOG`	Optional. Add the `LOG` parameter to force any suppressed messages to be written to the NOMATCH log. (This parameter is only relevant if the `NOMATCH` logging subtask has been defined and is active.)

DROP_SSI_SUPPRESSED Parameter Card

Indicate whether to honor the SSI suppression flag on MVS messages.

Valid Values

```
{YES|NO} [LOG]
```

Sample Syntax

```
DROP_SSI_SUPPRESSED YES LOG
```

Description

The DROP_SSI_SUPPRESSED parameter card is used when another application on the mainframe sets the Subsystem Interface (SSI) suppression flag on MVS messages (usually for the purpose of message filtering). Use this card if messages with the SSI flag should not be forwarded on to the OMW server.

Parameters

`YES	NO`	“YES” indicates that SSI suppression should be honored, and those MVS messages with the SSI flag set will not be forwarded to the OMW server. “NO” indicates that messages will not be automatically suppressed, but the message ID must still be present in the message filter table for it to be passed on to the OMW server. “NO” is the default.
`LOG`	Optional. Add the `LOG` parameter to force any suppressed messages to be written to the NOMATCH log. (This parameter is only relevant if the `NOMATCH` logging subtask has been defined and is active.)

DROP_WTO_USER_EXIT_SUPPRESSED Parameter Card

Indicate whether to honor the WTO User Exit suppression flag on MVS messages.

Valid Values

```
{YES|NO} [LOG]
```

Sample Syntax

```
DROP_WTO_USER_EXIT_SUPPRESSED YES LOG
```

Description

The DROP_WTO_USER_EXIT_SUPPRESSED parameter card is used when MVS messages have been acted on by a WTO user exit (usually for the purpose of message filtering). Use this card if messages with the WTO user exit flag should not be forwarded on to the OMW server.

Parameters

`YES	NO`	"YES" indicates that WTO user exit suppression should be honored, and those MVS messages with the WTO user exit flag set will not be forwarded to the OMW server. "NO" indicates that the WTO user exit suppression flag will be ignored, but a message ID must still be present in the message filter table for it to be passed on to the OMW server. "NO" is the default.
`LOG`	Optional. Add the `LOG` parameter to force any suppressed messages to be written to the NOMATCH log. (This parameter is only relevant if the `NOMATCH` logging subtask has been defined and is active.)

FILTER Parameter Card

Identify the messages that are to be sent to the OMW server and the commands that can be accepted from the OMW server.

Valid Parameters

```
MSG {[+|-]msgid [JOBNAME=(jobname[,jobname]...)]

  [JOBID=(jobid[,jobid]...)] [REXX=script]} ...

CMD regexpression
```

Sample Syntax

```
FILTER MSG ABC123I XYZ1234I -IST663I +IEE114I
FILTER MSG DEF* QRS...I
FILTER MSG JKL* JOBNAME=(MYJOB*)
FILTER MSG JKL.... JOBID=(STC*,TSU*)
FILTER MSG IEE362A REXX=CLEARSMF

FILTER CMD ^D TCPIP,.*,NETSTAT,ROUTE$
FILTER CMD ^D NET,MAJNODES$
```

Description

The FILTER card adds message IDs to the message filter table or commands to the command filter table.

FILTER MSG

z/OS messages must have a match in the message filter table to be forwarded to the OMW management server. The message table holds up to 2000 message IDs by default. If a successful match is found for a message ID, the job name and/or job ID of the job which generated the message can also be checked, and the message will only pass if it also matches the given job name or job ID. If a REXX script is specified and the RMA subtask is running, the named script will be called.

Parameters

msgid	The message ID of the message to pass to the OMW server. Typically, this is the first token of the message (up to the first space). *msgid* can contain the special period placeholder character (.) to indicate that any character in that position should match. If the *msgid* is terminated with an asterisk (*), matching will occur only on characters preceding the asterisk. Multi-line MVS messages will be forwarded to the OMW browser as one continuous message unless a plus sign "+" is placed in front of *msgid*, in which case the multiple lines will be sent as separate messages, each with *msgid* prepended to the message text, or a minus sign "-" in front of *msgid*, in which case the multiple lines will be sent as separate messages, but without the *msgid* prepended to the message text. *msgid* can be up to 15 characters in length.
jobname	The job name of the job which issued the message. The same wildcarding rules used in `msgid` may be applied to this parameter. *jobname* may be up to 8 characters in length.
jobid	The job ID of the job which issued the message. The same wildcarding rules used in msgid may be applied to this parameter. *jobid* may be up to 8 characters in length.
script	A REXX script to be called if this message is encountered. The script will be called with one parameter: the message text, the job name, and the job ID, separated by carriage return characters (x'0A'). (If there is no job name or job ID, these fields are filled in with "N/A"). See the sample programs JOBSUB, PARSEMSG, and SMFDUMP in the EV390.SAMP dataset for examples on how to set up automation programs. The REXX script must be available in the dataset specified by the DD card named on the RMA subtask card (see page 44).

FILTER CMD

Identify z/OS (MVS) commands that may be issued through the EView/390z extended console (initialized by the CMD parameter card). Specify only one command expression per FILTER CMD line. The command table holds up to 200 commands by default. If a command is issued which has no match in the command filter table, an EVO161 message will be returned to inform the caller that the command is unauthorized. Note that if no FILTER CMD cards are specified, then all commands will be forwarded to the console with no restrictions.

Parameters

regexpression	A Unix-style regular expression. An incoming command must match one of the regular expressions in the table of command filters before it will be sent to the console for execution. Be sure to use correct characters for the caret and square brackets if those characters are used in the *regexpression*. In the default IBM-1047 codeset, the values are: Character EBCDIC Hex Value Caret ^ x'54' Left Square bracket [x'AD' Right Square Bracket] x'BD' (Depending on the character mapping of the terminal emulator, the caret symbol may be displayed as a "not" ¬ symbol.)

FILTERTABLE Parameter Card

Specify the size of the message filter table and/or the command filter table.

Valid Values

```
[MSGSIZE=msgentries] [CMDSIZE=cmdentries]
```

Sample Syntax

```
FILTERTABLE MSGSIZE=3000 CMDSIZE=1000
```

Description

Use the FILTERTABLE card to set the number of message ID entries allowed in the message filter table and the number of command entries allowed in the command filter table. If this card is not present, the message filter table will default to allow 2000 message IDs and the command filter table will default to allow 200 commands. The FILTERTABLE card should only be used once, and it **must** be specified in the SYSIN deck **ahead of** any FILTER cards.

Parameters

`MSGSIZE=`*`msgentries`*	Specify the maximum number of entries in the message filter table. *`msgentries`* must be an integer between 1 and 500000.
`CMDSIZE=`*`cmdentries`*	Specify the maximum number of entries in the command filter table. *`cmdentries`* must be an integer between 1 and 500000.

MQMODEL Parameter Card
MQCOMMAND Parameter Card
MQDYNAMIC Parameter Card

Overrides to the default WebSphere MQ queue names.

Valid Values

Valid WebSphere MQ queue names (up to 48 characters)

Sample Syntax

```
MQMODEL EVIEW.COMMAND.REPLY.MODEL
MQCOMMAND EVIEW.COMMAND.INPUT
MQDYNAMIC EVIEW.COMMAND.REPLY.Q
```

Description

These three parameter cards define overrides to the default MQ queue names when sending a command to the WebSphere MQ. (See "Using OSINFO System Information API Commands" option 50 on page 49 of the *Administrator's Reference*.) The default queue names are:

Reply Model	SYSTEM.COMMAND.REPLY.MODEL
Command Queue	SYSTEM.COMMAND.INPUT
Dynamic Output Queue	EVIEW.COMMAND.REPLY.Q

If any of these default names are changed, the associated RACF permission must be changed to match the new names. (See "Allow MQ Commands" on page 25.)

If used, these MQ parameter cards must be placed in the SYSIN deck ahead of the OSINFO card.

MVS Parameter Card

Provides EView/390z with z/OS console message support.

Valid Values

```
MVS consname  [AUTO={YES|NO}]
              [DOM[=ALL] [DOMLIST[=size]]]
              [HC={YES|NO}]
              [HILITE={IGNORE|ONLY|ALL}]
              [IST]
              [MSCOPE={*|*ALL|sysname[,sysname...]}]
              [MON={[NAME[,]][SESS]}|NONE]
              [QL=limit] [QLP=percentage]
              [ROUT={ALL|NONE|rtcode[,rtcode]...}]
              [UD={YES|NO}]
```

Sample Syntax

```
MVS EVOCONS1 DOM ROUT=22,116-128 MON=NAME,SESS AUTO=NO UD=NO
      QL=15000 QLP=70 HILITE=ALL MSCOPE=SYS1,SYS2
```

Description

This card will initialize the MVS subtask and define an extended MCS software console to receive z/OS (MVS) messages for forwarding to the OMW server.

Parameters

consname	Required. Specify a 1-8 character name for the extended MCS console you wish to define for receiving z/OS (MVS) messages. If this name is defined in RACF, the OPERPARM values in the RACF entry for this name are used for the console definition. Otherwise, a console is defined with default parameter AUTH=INFO.
AUTO	Optional. Specify that the console will or will not receive messages that have been automated by the MVS Message Processing Facility (MPF). The default for this parameter is YES. That is, messages will be captured by this console even if they were previously processed by MPF automation.
DOM	Optional. Instruct the console to capture MVS "Delete Operator Messages" and forward them to the OMW server. Specify DOM to capture standard messages, or specify DOM=ALL to capture all DOMs (such as in a sysplex environment). The DOM information will be sent in an EVO211 message, giving the MVS message key or token of the deleted message(s). This information may be used for OMW message correlation and auto-acknowledgement. (The EVO211 message does not need to be added to the mainframe message filter table.)
DOMLIST	Optional. This option maintains a list of message IDs or tokens for all MVS messages that have been forwarded to the OMW server, and then only forwards DOMs which match an entry in this list. *size* is the number of message IDs and tokens to be saved in the list. If no size is specified, the DOMLIST defaults to 1000 messages. Use this parameter if the OMW server is receiving a large number of DOM messages which do not match any relevant message on the OMW browser. The DOMLIST parameter is valid only if the DOM parameter is also specified.
HC	Optional. Specifies whether this console should receive all messages destined for the hardcopy console. The default setting is NO.

`HILITE`

Optional. Allows special checking of highlighted MVS messages (messages preceded with an asterisk or at-sign) against the message filter table before forwarding to the OMW server. Set the `HILITE` parameter to one of these options:

- `IGNORE` This is the default behavior. That is, the message ID of highlighted messages must appear in the message filter table to allow the message to be passed on to the OMW server.
- `ONLY` Only highlighted messages with their message ID in the message filter table will be forwarded to the OMW server. (To send all highlighted messages, add an asterisk entry to the message filter table.)
- `ALL` All highlighted messages will be forwarded to the OMW server regardless of their message ID. Non-highlighted messages will still be forwarded if their message ID is in the message filter table.

Regardless of this option's setting, all messages are still subject to other restrictions, such as the MPF or SSI suppression parameter cards listed above.

`IST`

Optional. VTAM IST messages are normally filtered out by the MVS subtask. Use the IST parameter to remove this MVS subtask filter, allowing IST messages to pass. However, the IST parameter is not recommended unless both the PPO and the PPI interfaces are unavailable. (If this IST parameter is used in conjunction with either the PPO or PPI subtasks, the same VTAM message will be sent to the OMW server twice)

`MON`

Optional. Allows the console to monitor messages regarding the starting or stopping of jobs and user sessions. The `NAME` parameter is used to capture messages regarding the start and end of jobs (such as IEF403I and IEF404I). The `SESS` parameter is used to capture messages regarding the start and end of user sessions (such as IEF125I and IEF126I). The default setting is to activate the capture of both NAME and SESS messages.

`MSCOPE`

Optional. Specify the scope of which LPARs in a sysplex should send messages to this console. Set the `MSCOPE` parameter to one of these options:

- `*` This is the default behavior. The console will only receive MVS messages originating from the local LPAR on which the EView/390z job is running.
- `*ALL` The console will receive MVS messages from all LPARs in the sysplex.
- *`sysname`* The console will receive MVS messages from the list of systems named. Each *`sysname`* in the list must be no more than eight characters. The list may have up to eight LPAR names. Separate the names in the list with commas.

`QL`

Optional. Specifies the maximum number of messages that can be queued to this console. The queue limit may be any positive number up to 2147483647 (2 gigabytes). If not specified, *`limit`* is set to 5000.

`QLP`	Optional. Specifies the percentage of the console's internal queue that must be used before a warning message is sent to the OMW server `percentage` may be between 1 and 100. Message EVO703 is sent to the OMW server when the console's queue has met or exceeded this level. When the queue drops below the specified *`percentage`*, message EVO704 is sent to the OMW server to inform that the backlog has been relieved. If `QLP` is not specified, EView/390 will not issue the EVO703 and EVO704 messages.
	`EVO703  Console` *`consname`* `is utilizing` *`n`*`% of message queue`
	`EVO704  Console` *`consname`* `queue backlog has been relieved`

`ROUT`	Optional. Initialize the extended console with specific routing codes. Only messages with the specified routing codes will be captured by the console. Valid routing codes are in the range of 1-128 and can be specified as a single number (`ROUT=5`), a range of numbers (`ROUT=7-10`), or multiple numbers and ranges separated by commas (`ROUT=5,7-10,20,128)`. The default for this parameter is `ALL`. That is, the console will receive all messages regardless of routing code.
`UD`	Optional. Specifies whether this console is to receive undelivered messages. If set to `YES` and another console is taken offline, any messages that would have been routed exclusively to that offline console will instead be sent this console. The default setting is `YES`.

NLS Parameter Card

Set the National Language Support (NLS) codeset value.

Valid Parameters

```
[CODESET=value] [LANG=langcode]
```

Sample Syntax

```
NLS CODESET=IBM-1047
NLS CODESET=IBM-939 LANG=JPN
```

Description

This card is used to identify the character set used on the mainframe and language used for supported message translations. The *`value`* must be a codeset provided by the z/OS Language Environment. A list of codeset values is provided in Appendix D of the *IBM C/C++ Programming Guide* (IBM publication SC09-4765). The *`langcode`* must be a supported language code that is available on this mainframe.

Parameters

`value`	The name of the codeset for the locale of the mainframe. The default is the EBCDIC "IBM-1047" codepage.
`langcode`	The 3-character language code used for message translations. The default is "ENG" (English).

NOMATCHLOG Parameter Card

Identify the logging dataset(s) which record the messages that are not forwarded to the OMW server.

Valid Values

`log1 [log2 ...]`

Sample Syntax

```
NOMATCHLOG NMLOG1 NMLOG2 NMLOG3
```

Description

This card will initialize a `NOMATCH` subtask, which is responsible for writing any mainframe messages which were not passed on to the OMW server due to a defined filtering restriction. Each logging dataset will be written to until its space capacity is met. Writing will then switch to the next dataset in this list. When the last dataset is filled, the subtask will wrap back to the first dataset in the list and purge its contents before continuing its logging. The logging datasets will be written in the following format:

Column	Data
1	Reason for the filtering (in this order of precedence): M MVS message had the MPF suppression flag set, and the DROP_MPF_SUPPRESSED parameter card indicated that these messages are to be suppressed and logged. H MVS message had the Hardcopy Only flag set, and the DROP_HARDCOPY_ONLY parameter card indicated that these messages are to be suppressed and logged. E MVS message had the WTO User Exit suppression flag set, and the DROP_WTO_USER_EXIT_SUPPRESSED parameter card indicated that these messages are to be suppressed and logged. A MVS message had the Automation Requested flag set, and the DROP_AUTOMATION_FLAGGED parameter card indicated that these messages are to be suppressed and logged. S MVS message had the SSI flag set, and the DROP_SSI_SUPPRESSED parameter card indicated that these messages are to be suppressed and logged. V Message ID was not in the EView/390z message filter table.
3-9	Date of message in Julian format: YYYYDDD
11-18	Time of message in format: HH.MM.SS
20-27	Job name which produced message, if any
29-36	Originating system name of message, if available
38-45	Job ID which produced message, if any
47	MVS high-intensity character (*or @), if any
48-	Message text

Parameters

`logn`	The DD name(s) of the logging datasets. Up to ten DD names may be specified. Each name given must match a DD card definition in the VP390 started task JCL, and each logging dataset should be predefined with `DCB=(DSORG=PS,RECFM=V,LRECL=1651).`

OSINFO Parameter Card

Initializes the OSI subtask to respond to various requests for z/OS operating system information and statistics.

Valid Values

```
SDSFMAX=n
```

Sample Syntax

```
OSINFO SDSFMAX=400
```

Description

Use the OSINFO card to initialize a subtask which will accept command type 46 requests from the OMW server and return information about z/OS jobs and performance statistics. See the *EView/390z Administrator's Reference* for syntax of type 46 requests and the available options. Some of the options require SDSF to be active on the z/OS system and will also require two DD cards in the VP390 startup JCL: `ISFIN` and `ISFOUT`.

Parameters

n	An integer value indicating the maximum number of lines of information that will be returned from the queries to SDSF. Each line will contain information about one job. The default is 1000.

OPC Parameter Card

Start a subtask to manage incoming OVOPCMSG calls and passes them on to the OMW server.

Valid Values

```
[PORT=portnum]
```

Sample Syntax

```
OPC PORT=6200
```

Description

Use the OPC card to initialize a subtask which open a TCP/IP port and accept OVOPCMSG calls from other mainframe scripts to be forwarded to the OMW server.

Multiple OPC cards are allowed if OVOPCMSG calls may be coming from several sources. If multiple OPC cards are defined, the PORT option must be used to define a unique port for each of the several OPC subtasks. See page 53 of the *Administrator's Reference* for information on sending OpenView OPCMSG commands from the mainframe agent to the OMW server.

Parameters

portnum	Optional. An integer value between 1-65535 indicating the TCP/IP port number to listen to for OVOPCMSG data, but the preferred place to specify the port number is in the /etc/vp390info file. If no port number is specified, the port number defaults to 6108.

PERF Parameter Card

Initializes the PERF subtask to send RMF data to the OMW server at specific intervals.

Valid Values

```
[INTERVAL=n]
```

Sample Syntax

```
PERF INTERVAL=15
```

Description

Use the PERF card to initialize a subtask which will send RMF data to the OMW server at a defined interval. On the OMW server, the RMF data will be directed a message server log file. These data can be used in an OMW policy to be forwarded to the CODA subagent or the OV Performance Agent, if available.

Parameters

n	An integer value indicating how often, in minutes, the RMF data is sent to the OMW server. The default is 15 minutes.

PPI Parameter Card

Request setup of the PPI to NetView/390 or SOLVE:NETMASTER.

Valid Values

`[BUFLEN=`*`n`*`]`

Sample Syntax

```
PPI
```

Description

This card will initialize a PPI subtask. Add this card to connect EView/390z to the NetView/390 or SOLVE:NETMASTER PPI for the receipt of VTAM messages. The PPI must be active in accordance with the NetView/390 or SOLVE:NETMASTER documentation. This executable is responsible for communicating with the NTIPPI executable which runs as a user exit in the NetView/390 or SOLVE:NETMASTER address space. The NetView/390 or SOLVE:NETMASTER application will control the PPO connection to VTAM information. The PPI subtask should only be used when NetView/390 or SOLVE:NETMASTER is present on the system.

➤ Do not include this card if neither NetView/390 nor SOLVE:NETMASTER is present on the system. Use the `PPO` parameter card instead.

Parameters

n	Optional. Add this parameter to the PPI card to specify a non-standard buffer length size. The default size is 104. Use `BUFLEN=40` if you receive an EVO096 error message when attempting to connect to an older version of NetView or SOLVE:NETMASTER.

PPO Parameter Card

PPO APPL defined to allow EView/390z to act as the VTAM Primary Program Operator.

Valid Values

<VTAM APPL Definition> `[ECHO]`

Sample Syntax

```
PPO EVOPPO1 ECHO
```

Description

This card will initialize a `PPO` subtask that initializes a Primary Program Operator ACB for the purpose of receiving unsolicited messages from VTAM, most importantly those messages regarding status changes of VTAM resources.

➤ Do not include this card if EView/390z if is running in combination with other management software such as NetView/390 or SOLVE:NETMASTER, as only one application in a domain can be the PPO. Use the `PPI` parameter card instead.

Parameters

VTAM APPL Definition	Required. Specify the name of the VTAM APPL definition coded with `AUTH=PPO`. The default APPL provided in the EView/390z application major node is "EVOPPO1". (See "Adding the *hlq*.SAMP(EVAPPL) Definition" on page 27.)
`ECHO`	Optional. If used, it directs EView/390z to forward a copy of the VTAM messages received to the operator console.

PRINTCARDS Parameter Card

Write the SYSIN cards to the SYSPRINT.

Valid Values

None

Sample Syntax

```
PRINTCARDS
```

Description

The PRINTCARDS card instructs the VP390 job to print each of the SYSIN lines that it reads to the SYSPRINT, excluding blank lines and comment lines. This is usually only used to document the cards that have been read when sending the SYSPRINT output to support for problem analysis. Enter PRINTCARDS as the first line of the SYSIN dataset member to document all the lines of the SYSIN input.

The PRINTCARDS card can be used multiple times in the SYSIN as a toggle control. The second occurrence of PRINTCARDS will stop the writing to SYSPRINT, the third occurrence will resume the writing, and so forth.

Parameters

None

RESTART Parameter Card

Number of restart attempts to allow a subtask before giving up.

Valid Values

`1` to `65535`, or `UNLIMITED`

Sample Syntax

```
RESTART 100
RESTART UNLIMITED
```

Description

Specifies the number of times a subtask attempts to automatically restart. After this limit is reached, the subtask remains in a "Down" state until it is manually reactivated using the `INIT` command. (See the description of the `INIT` command in Appendix A of the *EView/390z Administrator's Reference*.) Specify `UNLIMITED` instead of a number to allow a subtask to make an unlimited number of restart attempts. Each subtask can have a unique restart count by specifying another `RESTART` card immediately before the card which defines the subtask. The default `RESTART` value is `5`.

RMA Parameter Card

Start a subtask to manage REXX Message Automation.

Valid Values

`[DD=`*`ddname`*`]`

Sample Syntax

```
RMA DD=OVEXECS
```

Description

The RMA subtask processes any messages that are matched in the message filter table and have a REXX script associated with it (see the FILTER card definition on page 32). The named script will be called with one parameter, composed of the message text, the job name, and the job ID, separated by carriage return characters (x'0A'). If there is no job name or job ID, the field will be filled in with "N/A". Any text returned from the REXX script will be forwarded to the OMW server. See the *Administrator's Reference* for samples of REXX automation scripts. Multiple `RMA` cards may be defined to distribute the workload in case a long-running REXX program ties up one RMA for an extended period of time. If multiple cards are defined, all must have the same *`ddname`* specified (if the default is not used).

Parameters

ddname	Optional. The DD name of the dataset that holds the REXX execs. This name must match a DD name in the VP390 startup JCL. The default DD name is SYSEXEC, which points to the VP390 CLIST dataset.

RMFCYCLE Parameter Card

Provide the agent with the RMF measurement interval.

Valid Values

n	Integer value between 50 and 9999 (milliseconds)

Sample Syntax

```
RMFCYCLE 500
```

Description

This parameter is only necessary if you are using the PERF subtask, or are using the RMF options of the OSINFO subtask. This parameter should be set to match the CYCLE parameter in the RMF initialization (member ERBRMFxx in PARMLIB). The default RMF sampling period is 1000 milliseconds. You do not need to add the RMFCYCLE card if RMF uses the default cycle time.

Parameters

n	An integer value between 50 and 9999 representing the number of milliseconds in the RMF sampling cycle. 1000 is the default.

SEC Parameter Card

Load module to be called for security calls.

Valid Values

<Load module name>

Sample Syntax

```
SEC EVRACF
```

Description

This card will initialize a `SEC` subtask and will use the specified load module in *`hlq`*.LOAD to process calls to the mainframe security software. Currently, the only valid load module is EVRACF. EView/390z has been tested successfully with IBM RACF and Computer Associates ACF2 software. The `SEC` subtask accepts user IDs, passwords, and (optionally) new passwords to be sent to RACF or ACF2 for verification, and will send one of several return codes back to the requesting OMW application based on the response from RACF or ACF2.

SMFBUFFER Parameter Card

Identify the size of the work buffer when calling SMF functions.

Valid Values

n	Integer value between 260 and 2000000000.

Sample Syntax

```
SMFBUFFER 1200000
```

Description

This parameter is only necessary if you are using the PERF subtask, or are using the RMF options of the OSINFO subtask. The calls to collect SMF information require a predefined block of memory which is used as a work area by SMF. The size of this work area is dependent on how many processes are running on the mainframe. By default, EView/390z allocates a buffer of 1000000 bytes. If you are consistently receiving error message EVO131 with an error code "-104" when issuing EView commands to the mainframe, use this parameter card to increase the size of the allocated buffer. (The buffer can also be increased dynamically while the EView/390z job is running by using the Modify command. See "About PERF Commands" in Appendix A of the *EView/390z Administrator's Reference*.)

Increasing the SMF buffer size by a substantial amount may require increasing the REGION size of the EView/390z address space.

Parameters

`n`	An integer value between 260 and 2000000000 representing the number of bytes to be allocated for the SMF work buffer.

SPO Parameter Card

SPO APPL to allow EView/390z to send commands to VTAM through a Secondary Program Operator.

Valid Values

<VTAM APPL Definition>

Sample Syntax

```
SPO EVOSPO1
```

Description

This card will initialize a SPO subtask with the ID of an APPL definition card coded with `AUTH=SPO`. This identifies a Secondary Program Operator (SPO) application, which receives solicited messages generated by commands issued from the OMW server. This subtask executable is responsible for initializing a SPO ACB to VTAM, then receiving VTAM commands (for example, `Vary` or `Display`) from OMW management servers, sending the commands to VTAM over the SPO, and sending the VTAM responses to the OMW server that initiated the command. Multiple `SPO` subtasks are allowed under EView/390z to distribute the work if several commands come in at nearly the same time from different OMW operators.

Parameters

VTAM APPL Definition	Required. Specify the name of the VTAM APPL definition coded with `AUTH=SPO`. The default APPL provided in the EView/390z application major node is "EVOSPO1". (See "Adding the *hlq*.SAMP(EVAPPL) Definition" on page 27.)

TCP Parameter Card

Identify port numbers and parameters for the TCP/IP connection to the OMW server.

Valid Values

```
mmsport cmdport [hlq] [BUFDD=dd1,dd2 [ACK=ack] [LIMIT=limit]] [HB=hb]
              [BINDIP=bindaddr] SERVERIP=servaddr[/{maskaddr|maskprefix}]]
```

Sample Syntax

```
TCP 6106 6107 BUFDD=BFR1,BFR2 ACK=5 LIMIT=20 HB=30 BINDIP=10.1.1.8 SERVERIP=10.1.1.0/24
```

Description

This card will initialize a `TCP` subtask, which is responsible for opening two TCP/IP ports on the mainframe, then waiting for an EView/390z server component to start communication with the mainframe agent via these ports. While it waits for a connection, the `TCP` subtask can optionally write new mainframe messages to a set of buffering files, and then send the buffered messages after a connection is established. Multiple TCP cards are allowed. (If you are also using the same EView/390z job for uCMDB Discovery and Operations Management, you will need to define a TCP card for each connection.) The *`mmsport`* and *`cmdport`* parameters must be unique for each TCP card defined.

Parameters

`mmsport`	Required. Port number opened on the mainframe for establishing a socket connection with the Master Message Server task on the OMW management server. This number must match the EVOMF_HCI_AGENT_PORT value entered when adding the z/OS node through the EView/390z Task Manager.
`cmdport`	Required. Port number opened on the mainframe for establishing a socket connection with the Command Server task on the OMW management server. This number must match the EVOMF_CMDS_AGENT_PORT value entered when adding the z/OS node through the EView/390z Task Manager.
`hlq`	Optional, deprecated. High-level qualifier (hlq) for the mainframe TCP/IP datasets. This parameter is used to find the TCP/IP profile datasets, and is needed only if the default hlq is not used during TCP/IP installation. The *`hlq`* must be identified on each card. The TCP subtask will not be initialized if the *`hlq`* is misstated.
`dd1,dd2`	Optional. The DD names of the two buffering datasets. These DD names must be listed in the VP390 startup job, and they must point to predefined datasets with DCB=(DSORG=PS,RECFM=V,LRECL=1663).

`ack`	Optional. The number of unsolicited mainframe messages that will be passed on to the OMW server before an acknowledgment is expected from the OMW server. By default, EView/390z will expect an acknowledgment after every 5 messages. If an acknowledgment is not received, the mainframe agent will resend all of the messages back to the last successful acknowledgment. Then, if the OMW server acknowledgment is still not received, the mainframe agent will close the TCP/IP connection and wait for a reconnect request. Upon reconnection, all unacknowledged messages back to the last successful acknowledgment will be resent to the OMW server, and these messages may appear duplicated in the OMW browser. If this causes a problem, set the `ACK` parameter to 1 to ensure an acknowledgment from the OMW server after every message is sent. The `ACK` parameter is only valid if `BUFDD` is specified
`limit`	Optional. The age limit (in minutes) of buffered messages that the mainframe agent will send to the OMW server. By default, messages read from the buffering files that are over 20 minutes old will not be forwarded to the EView/390z client. Set this value to 0 to receive all buffered messages regardless of their age. The `LIMIT` parameter is only valid if `BUFDD` is specified
`hb`	Optional. Length of time (in seconds) between heartbeat tests to verify the TCP/IP connection. By default, a short heartbeat message will be sent between the OMW server and the mainframe agent every 30 seconds.
`bindaddr`	Optional. An IPV4 dotted decimal address that names a specific mainframe IP address that the listening ports should bind to (useful at sites with multiple TCP/IP addresses defined in the same mainframe LPAR).
`servaddr`	Optional. An IPV4 dotted decimal address that names a specific OMW server IP address that the listening ports will accept a connection from. *`servaddr`* must be specified in the nnn.nnn.nnn.nnn IP address format. Additionally, the *`servaddr`* can be masked to allow a range of IP addresses to connect to the ports. To specify a mask, add a slash (/) after *`servaddr`* followed by either a dotted decimal *`maskaddr`* (e.g., 255.255.255.0) or a numerical *`maskprefix`* between 0-32 to represent the number of bits to be included in the mask starting from the leftmost bit (e.g.: "24" would be equivalent to 255.255.255.0).

5

Updating NetView/390

This chapter explains the updates required for EView/390z to work in conjunction with IBM NetView/390. NetView/390 must be restarted for the changes to take effect.

Phase 1: Verifying the Subsystem Interface Installation

As a first step in updating NetView/390, verify that the NetView/390 subsystem address space is active, as defined in the IBM *NetView/390 Installation and Administration Guide*. The NetView/390 subsystem interface is necessary for cross-memory communications between NetView/390 and EView/390z. The subsystem address space is usually started when NetView/390 and the job name begins with the same four characters as the NetView/390 job name.

Phase 2: Assembling and Linking NetView/390 Exits

To run in the NetView/390 address space, EView/390z uses two exits, a DST, and a command processor.

Modify and submit the JCL in `hlq.SAMP(ASMJCL)` according to instructions in that member to create one or more of the following load modules, based on your needs:

Table 5-1: NetView Exits

Load Module Name	Description
NTIPPI	This program runs as a NetView DST task. It collects the messages and command responses from the other three programs and sends it out of the NetView address space, via the SSI, to the VP390 address space. This program must be assembled and running if any of the other three programs below are to be used.
DSIEX06	This exit captures solicited VTAM message responses from commands that NetView operators issued through the NetView Primary Operator Interface (POI). Use this exit if you expect NetView operators to be issuing VTAM Vary commands against SNA resources, and you want OMW to see the effects of such commands on the SNA network.
DSIEX11	This exit captures unsolicited VTAM messages that come in through the NetView POI. DSIEX11 is generally more important than DSIEX06, since DSIEX06 messages are user-initiated, while DSIEX11 messages will be generated unexpectedly when there is trouble in the SNA network.
NTIMQS	This program is used by CLISTs running under a NetView autotask. The CLISTs execute NetView commands sent to the mainframe from the OMW server, either automatic actions or individual operator commands, so by restricting the command security level of the NetView autotask, you can restrict what NetView commands may be executed from the OMW server. When the command responses arrive, the CLIST will use this NTIMQS program to forward the response message(s) to the NTIPPI task.

Phase 3: Updating NetView/390 Datasets

Next, update the NetView/390 datasets and the initial command list

➤ If you use the CNMSTYLE member of DSIPARM for your customization of NetView, use the "Updating CNMSTYLE" section, and skip the "Updating DSIDMN" and "Updating the Initial Command List" sections.

Updating CNMSTYLE

Update the CNMSTYLE member of DSIPARM as follows:

1. In the section defining autotasks, add the following line:

```
AUTOTASK.EVOAUTO1.Console = *NONE*
```

2. In the section defining optional tasks, add the following lines:

```
TASK.NTIPPI.MOD=NTIPPI
TASK.NTIPPI.PRI=8
TASK.NTIPPI.INIT=Y
```

3. If the DSIEX06 or DSIEX11 exits were assembled in Phase 2 above and are expected to be used, then change the LOADEXIT parameter for the exit(s) from "No" to "Yes".

Updating DSIDMN

Update the DSIDMN as follows:

1. Define the EView/390z Mainframe Collector optional task.

   Add the following definition to a DSIDMN member of your NetView/390 DSIPARM dataset:

   ```
   TASK MOD=NTIPPI,TSKID=NTIPPI,PRI=8,INT=Y
   ```

2. Verify that the two NetView/390 tasks, CNMCALRT and CNMCSSIR, are defined:

   - Define CNMCALRT with INIT=Y.
   - Define CNMCSSIR with INIT=N. Start CNMCSSIR in the command list CNME1035 during NetView/390 initialization.

   These tasks provide command and message forwarding services for EView/390z.

Updating DSICMD or CNMCMD

Define a command model for the `NTIMQS` load module. If you are running a NetView prior to V5R2, add the following definition line to a `DSICMD` member of your NetView/390 `DSIPARM` dataset:

```
NTIMQS CMDMDL MOD=NTIMQS,RES=N
```

If you are running NetView V5R2 or later, add the following lines to a CNMCMD member of the NetView/390 DSIPARM dataset:

```
CMDDEF.NTIMQS.MOD=NTIMQS
CMDDEF.NTIMQS.RES=N
```

Updating DSIOPF

Define an additional NetView/390 autotask by adding the following definition to the DSIOPF member of your NetView/390 DSIPARM dataset:

```
EVOAUTO1 OPERATOR PASSWORD=PASSWORD
PROFILEN EVOPROF
```

Although you may change the operator ID (EVOAUTO1) to conform to your site requirements, it must match the EVOCMD_OPERATOR configuration parameter on the OMW management server. For details about EView/390z configuration parameters on the OMW management server, see the *EView/390z Administrator's Reference*.

You may also change the PROFILEN name (EVOPROF) to conform to your site requirements. The profile is defined in DSIPRF.

Updating DSIPRF

Define a profile for the operator ID by adding a member named EVOPROF to your NetView/390 DSIPRF dataset.

The EVOPROF definition must contain the following three lines:

```
EVOPROF PROFILE
AUTH MSGRECVR=NO,CTL=GLOBAL
END
```

Although you may change the member name to conform to your site requirements, it must match the PROFILEN statement coded in DSIOPF.

Updating the Initial Command List

To ensure that the autotask defined in DSIOPF is started each time NetView/390 is brought up, add the following line to your initial command list:

```
AUTOTASK OPID=EVOAUTO1
```

➤ The initial command list is identified by the NCCFIC line in DSIDMN.

Phase 4: Copying Members to NetView/390 Libraries

Copy the following two members from EView/390z's *hlq*.CLIST dataset into a NetView/390 DSICLD dataset:

- `NTICMD`
- `NTIMVS`

Phase 5: Restarting NetView/390

As a final step in updating NetView/390, you must restart NetView/390 to activate all updates and exits.

6

Updating SOLVE:NETMASTER

This chapter describes the updates required for EView/390z to work in conjunction with Computer Associates SOLVE:NETMASTER. SOLVE:NETMASTER does not have to be restarted for the changes to take effect.

About Dataset Members

General information about the SOLVE:NETMASTER dataset members:

Types of Dataset Members

EView/390 includes the following dataset members for SOLVE:NETMASTER:

NMASTDOC	Documents procedures and how to implement EView/390z.
NETTKPPO	Documents changes to PPOPROC for EView/390z.
NETTKPPI	Sends PPI messages (originating from PPO) to EView/390z.
NETTKCMD	Receives PPI commands (EVNETV) and starts NETTKCMI.
NETTKCM1	Sends and receives commands

Location of Dataset Members

SOLVE:NETMASTER dataset members are located in the following EView/390z dataset:

hlq.CLIST

Phase 1: Verifying the Subsystem Interface Installation

The PPI is necessary for cross-memory communications between SOLVE:NETMASTER and EView/390z. Verify that the SOLVE:NETMASTER PPI address space is active, as defined in the Computer Associates *SOLVE:NETMASTER Implementation and Administration Guide*.

Phase 2: Updating PPOPROC

To enable EView/390z to receive system message information from SOLVE:NETMASTER, add the Network Control Language (NCL) code in *hlq*.CLIST(NETTKPPO) to the production PPOPROC at a point where all messages are seen. Add this code immediately following the mainline &PPOREAD.

To start the PPOPROC, specify SYSPARM PPOPROC=*procname*. To receive copies of VTAM commands in the PPOPROC, specify SYSPARMS PPOSOCMD=PPOPROC and PPOLOG=YES. To receive specific messages, issue the DEFMSF DELIVER=PPO command either in the PPOPROC or before starting PPO. For details, see the Computer Associates *SOLVE:NETMASTER Management Services Planning and Installation and Command Reference* manual.

Phase 3: Copying Members to SOLVE:NETASTER Libraries

Copy the following three members from *hlq*.CLIST into a SOLVE:NETMASTER COMMAND DD dataset:

- NETTKCMD
- NETTKCM1
- NETTKPPI

Phase 4: Updating PPI

NETTKPPI and NETTKCMD are the primary PPI procedures that send PPO data through the PPI and wait for commands coming from EView/390z through the PPI. For this reason, NETTKPPI and NETTKCMD must be active and running in the background at all times.

To keep both procedures active and running in the background at all times, add the following statements to your NMINIT or NMREADY initialization procedure:

```
Sub BSYS NETTKPPI
Sub BSYS NETTKCMD
```

These commands also may be issued from the OCS console.

Phase 5: Verifying Updates

After completing all updates to SOLVE:NETMASTER, verify correct installation by issuing the following command:

```
SH PPIUSERS
```

This command displays two receivers, EVNETV and EVOPEN, after the VP390 address space begins and the PPI subtask makes its connection. The command indicates the number of messages queued to allow monitoring of the number of messages that are sent to EView/390z.

7

Updating CICS

This chapter explains the updates required for EView/390z to use the XMEOUT exit in CICS to capture messages generated by a CICS region, and the XMNOUT exit to monitor CICS transaction response times.

XMEOUT Phase 1: Identify CICS Messages in Code

EView/390z provides the assembler source code for a CICS XMEOUT exit which will redirect CICS messages from a transient data queue to the console by changing the message route code. The EView/390z MCS console will then capture these messages and pass them on to the OMW server.

By default, this XMEOUT exit redirects all CICS messages to the console. Restricting the messages which are sent to the console requires identifying the message IDs in the XMEOUT source code before it is assembled.

To identify specific CICS messages for forwarding to the console:

1. Edit the `hlq.ASM(EVXMEOUT)` assembler code and change the RERTEALL flag from 'Y' to 'N'.
2. In the table labeled TDQTAB, add the four-character name of the queue where the desired message is usually directed and the four-digit message ID.
3. Save the modified EVXMEOUT code.

XMEOUT Phase 2: Assembling and Linking the CICS Exit

Modify and submit the JCL in `hlq.SAMP(ASMCICS)` according to the instructions in that member to create an EVXMEOUT load module. The output load module must be stored in a CICS STEPLIB or DFHRPL load library or a LNKLST load library.

XMEOUT Phase 3: Activating the Exit

Enter the following commands from a CICS session to activate the new XMEOUT exit:

```
CEDA DEFINE PROGRAM(EVXMEOUT) GROUP(EVOGRP)
CEDA INSTALL PROGRAM(EVXMEOUT) GROUP(EVOGRP)
CECI ENABLE PROGRAM(EVXMEOUT) EXIT(XMEOUT) START
```

The CECI ENABLE command must be executed again each time CICS is restarted unless it is incorporated into a CICS Program Load Table.

XMEOUT Phase 4: Set up Automatic Initialization

[Optional.] Use the following steps to add an entry to the CICS PLTPI table to activate the XMEOUT exit each time CICS is started (eliminating the need for entering the above CECI ENABLE command).

1. Create a PLTPI program.

   Run the CICS DFHEITAL procedure with assembler code to start the EVXMEOUT exit. The assembler input for the DFHEITAL job is in `hlq.ASM(EVPLTPI)`. A sample DFHEITAL job is available in `hlq.SAMP(DFHEITAL)`.

2. Update the PLTPI table.

   Add the name of the load module created in Step 1 to the PLTPI table after the DFHDELIM entry. By default, the name is EVPLTPI. A sample PLTPI table follows:

   ```
   * LIST OF PROGRAMS TO BE EXECUTED SEQUENTIALLY DURING SYSTEM
   * INITIALIZATION.
   *
            DFHPLT TYPE=INITIAL,SUFFIX=I1
            DFHPLT TYPE=ENTRY,PROGRAM=TRAQA
            DFHPLT TYPE=ENTRY,PROGRAM=TRAQB
   *
            DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
   *
            DFHPLT TYPE=ENTRY,PROGRAM=TRASA
            DFHPLT TYPE=ENTRY,PROGRAM=TRASB
            DFHPLT TYPE=ENTRY,PROGRAM=EVPLTPI
            DFHPLT TYPE=FINAL
   *
            END
   ```

   Use the DFHAUPLE job to assembler the PLTPI table. A sample DFHAUPLE job is available in *`hlq`*`.SAMP(DFHAUPLE)`.

3. Identify the PLTPI to CICS.

   Add a PLTPI entry to the CICS startup parameters if one is not already specified.

4. Add a definition for the new PLTPI module.

   Enter the following command from a CICS session to define the new PLTPI module:

   ```
   CEDA DEFINE PROGRAM(EVPLTPI) GROUP(EVOGRP) LANGUAGE(ASSEMBLER)
   ```

XMNOUT Phase 1: Assembling and Linking the Threshold Table

EView/390z provides a CICS XMNOUT exit which will monitor the time used by CICS transactions. If a CICS transaction runs for longer than the value in the time threshold table, an EVO351 message will be written to the console, which can then be captured by the EView/390z agent and forwarded to the OMW server.

The code and JCL for the threshold table is located in the EView/390z *`hlq`*`.ASM(EVTHRTBL)` member. Use the following steps to create the table for the XMNOUT exit to use:

1. Edit the EVTHRTBL CSECT in the member, listing each four-character transaction name and the four-digit transaction time threshold value. The format of the time threshold is:

   ```
   SSHH
   ```

   where:

   `SS` = seconds
   `HH` = hundredths of seconds

   For example, a threshold of half a second would be ‘0050’.

2. Verify that the last transaction name in the list is ‘####’.
3. Change the SYSLMOD DD card to point to a CICS LINKLIB dataset to receive the assembled table.
4. Submit the JCL job to create the EVTHRTBL for the exit.

XMNOUT Phase 2: Activating the Exit

Load the exit program into CICS with the following steps:

1. Copy the EVXMNOUT load module from the EView/390z *`hlq`*`.LOAD` library to a CICS STEPLIB or DFHRPL load library or a LNKLST load library.
2. Define the program with the CICS command (use any site-determined name for the GROUP):

   ```
   CEDA DEFINE PROGRAM(EVXMNOUT) GROUP(EVOGRP) LANGUAGE(ASSEMBLER)
   CONCURRENCY(THREADSAFE) API(OPENAPI)
   ```

3. Install the CICS program with the CICS program:

   ```
   CEDA INSTALL PROGRAM(EVXMNOUT) GROUP(EVOGRP)
   ```

4. Bring in the new copy of the program with the CICS command:

   ```
   CEMT SET PROGRAM(EVXMNOUT) NEW
   ```

5. Enable EVXMNOUT as the XMNOUT exit with the CECI transaction (or add to an existing PLTPI startup):

   ```
   CECI ENABLE PROGRAM(EVXMNOUT) EXIT(XMNOUT) START
   ```

To disable the exit, use the following CECI command:

```
CECI DISABLE PROGRAM(EVXMNOUT) EXITALL STOP
```

To refresh the exit after updating the threshold table, use the commands:

```
CECI DISABLE PROGRAM(EVXMNOUT) EXITALL STOP
CEMT SET PROGRAM(EVXMNOUT) NEW
CECI ENABLE PROGRAM(EVXMNOUT) EXIT(XMNOUT) START
```

8

Starting and Stopping the Mainframe Component

This chapter explains how to start and stop the EView/390z VP390 job, and the NetView/390 and SOLVE:NETMASTER PPI interfaces, if present.

Running NetView/390 Automatically

The EView/390z Mainframe Collector task `NTIPPI` starts automatically whenever NetView/390 is started.

To Stop the NTIPPI Task

To recycle NTIPPI, stop the task by issuing the following command from a NetView/390 command prompt:

```
STOP TASK=NTIPPI
```

To restart the NTIPPI Task

To restart `NTIPPI` task, issue the following command from a NetView/390 command prompt:

```
START TASK=NTIPPI
```

Running SOLVE:NETMASTER Continuously

`NETTKPPI` and `NETTKCMD` are the primary PPI procedures that send PPO data through the PPI, or await commands coming from the VP390 task through the PPI. For this reason, both procedures must be active and running in a background within SOLVE:NETMASTER at all times.

To make sure that `NETTKPPI` and `NETTKCMD` are active and running in the background at all times, add the following statements to your `NMINIT` or `NMREADY` initialization procedures:

```
Sub BSYS NETTKPPI
Sub BSYS NETTKCMD
```

These commands can also be issued from an OCS console.

Running EView/390z as a Started Task

The EView/390z job “VP390” may be run as a started task.

To Start the VP390 Job as a Started Task

To start VP390 as a task, follow these steps:

1. Copy the *`hlq`*`.SAMP(VP390)` procedure into the started tasks library.
2. Modify the dataset names according to the instructions at the top of the job.
3. Start the VP390 procedure from a z/OS console with the following command:

   ```
   S VP390
   ```

To Stop the VP390 Task

To stop the VP390 task, enter the following command from a z/OS console:

```
P VP390
```

Running EView/390z as a Batch Job

The EView/390z job “VP390” may be run as a batch job.

To Start the VP390 Job as a Batch Job

To start the VP390 as a batch job, modify and submit the JCL in `hlq.SAMP(VP390JCL)`.

To Stop the VP390 Batch Job

To stop the VP390 batch job, enter the following command from the operator console:

```
P VP390
```