

EView/390z Management for HP Operations Manager Unix/Linux

Installation Guide

Software Version: 6.1



February 2012

Copyright 2012 EView Technology, Inc.

Legal Notices

Warranty

EView Technology makes no warranty of any kind with regard to this document, 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.

A copy of the specific warranty terms applicable to your EView Technology product can be obtained from your local Sales and Service Office.

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 in DFARS 252.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

EVIEW TECHNOLOGY, INCORPORATED
4909 Green Road, #133
Raleigh, North Carolina 27616
United States of America

Copyright Notices

Copyright 2012 EView Technology, Inc.

Copyright 2005 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of EView Technology, Inc. or Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

MVS, OS/390, z/OS and VTAM are products of International Business Machines Corporation.

NetView/390 is a product of Tivoli Corporation.

OpenView® is a registered U.S. trademark of Hewlett-Packard Company.

SOLVE:NETMASTER is a product of Computer Associates International, Inc.

UNIX® is a registered trademark of the Open Group.

Java is a registered U.S. trademark of Sun Microsystems, Inc.

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

Support

Support questions may be sent to support@evview-tech.com

Revision History

This manual's title page contains the following identifying information:

- Version number, which indicates the software version.
- Print date, which changes each time the document is updated.

Table 1 indicates changes made to this document since the last released edition.

Table 1: Changes to This Document

Date	Description
18-Dec-2007	Version 6.0
12 Feb 2010	Version 6.0 Minor Corrections
01 May 2010	Version 6.1
22 July 2010	Version 6.1 Corrections to Solaris Installation Steps
15 Nov 2011	Version 6.1 Corrections to gathering License Key info
17 Feb 2012	Updated version 6.1 mainframe files to version 6.3

Table of Contents

Contents	7
Conventions	11
Documentation Map	13
EView/390z Printed Manuals	13
EView/390z Online Information.....	13
Installing and De-installing EView/390z	15
Installation Requirements.....	16
Hardware Requirements	16
Software Requirements	16
Obtaining License Keys	17
Verifying the Software Files.....	17
Installing EView/390z on the Management Server.....	19
Download the EView/390z Software Bundle.....	19
Install EView/390z on a Management Server with HP-UX.....	19
Install EView/390z on a Management Server with Solaris	21
Install EView/390z on a Management Server with Linux	21
Verify the Installation on the Management Server	22
Installed File Locations on the Management Server	22
Installing EView/390z on the Managed Nodes	24
What to Upload	24
Transferring Files to the Mainframe	25
Extracting Partitioned Datasets from Sequential Datasets.....	25
De-installing EView/390z.....	27
Remove EView/390z Components from the OM GUI	27
Remove EView/390z from the OM Management Server	27

Remove EView/390z from the OM Managed Nodes.....	28
Updating Mainframe Software	29
Phase 1: Updating TCP/IP Connectivity.....	30
Reserving Port Numbers in PROFILE.TCPIP	30
Identifying the active TCPIP.DATA file	30
Phase 2: Updating z/OS and VTAM.....	30
Authorizing the <i>hlq</i> .LOAD Dataset	31
Setting the Performance Group or Service Class	31
Adding an Entry to the Program Properties Table	31
Adding an Entry to the RACF Class.....	32
Allow Viewing of all JES2 Jobs.....	33
Adding PPOLOG to VTAM Startup Options.....	35
Adding the <i>hlq</i> .SAMP(EVAPPL) Definition	35
Phase 3: Updating VP390 Parameter Cards.....	35
CMD Parameter Card	36
DELAY Parameter Card.....	36
DROP_AUTOMATION_FLAGGED Parameter Card	37
DROP_HARDCOPY_ONLY Parameter Card	37
DROP_MPF_SUPPRESSED Parameter Card	38
DROP_SSI_SUPPRESSED Parameter Card	39
DROP_WTO_USER_EXIT_SUPPRESSED Parameter Card	39
FILTER Parameter Card	40
FILTERTABLE Parameter Card.....	41
MVS Parameter Card.....	42
NLS Parameter Card	45
NOMATCHLOG Parameter Card.....	45
OSINFO Parameter Card.....	46
OPC Parameter Card.....	47
PERF Parameter Card.....	48
PPI Parameter Card.....	48
PPO Parameter Card.....	49
PRINTCARDS Parameter Card	49
RESTART Parameter Card.....	50
RMA Parameter Card	50

RMFCYCLE Parameter Card.....	51
SEC Parameter Card	51
SMFBUFFER Parameter Card.....	52
SPO Parameter Card.....	52
TCP Parameter Card	53
Updating NetView/390	57
Phase 1: Verifying the Subsystem Interface Installation	58
Phase 2: Assembling and Linking NetView/390 Exits	58
Phase 3: Updating NetView/390 Datasets	59
Updating CNMSTYLE	59
Updating DSIDMN	60
Updating DSICRTTD	60
Updating DSICMD or CNMCMD	60
Updating DSIOPF	60
Updating DSIPRF	61
Updating the Initial Command List	61
Phase 4: Copying Members to NetView/390 Libraries	61
Phase 5: Restarting NetView/390	61
Updating SOLVE:NETMASTER.....	63
About Dataset Members.....	64
Types of Dataset Members	64
Location of Dataset Members	64
Phase 1: Verifying the Subsystem Interface Installation	64
Phase 2: Updating CNMPROC	64
Phase 3: Updating PPOPROC.....	65
Phase 4: Copying Members to SOLVE:NETMASTER Libraries	65
Phase 5: Updating PPI.....	65
Phase 6: Verifying Updates.....	65
Updating CICS	67
Phase 1: Identify CICS Messages in XMEOUT Code	68

Phase 2: Assembling and Linking the CICS XMEOUT Exit	68
Phase 3: Activating the XMEOUT Exit	68
Phase 4: Set up Automatic Initialization	68
Starting and Stopping the Mainframe Component.....	71
Running NetView/390 Automatically	72
To Stop the NTIPPI Task	72
To Restart the NTIPPI Task.....	72
Running SOLVE:NETMASTER Continuously	72
Running EView/390 as a Started Task.....	72
To Start the VP390 Job as a Started Task.....	72
To Stop the VP390 Task.....	73
Running EView/390 as a Batch Job	73
To Start the VP390 Job as a Batch Job	73
To Stop the VP390 Batch Job.....	73
Glossary	75



Conventions

The following typographical conventions are used in this manual.

Table 1: Typographical Conventions

Font	Meaning	Example
<i>Italic</i>	Book or manual titles, and man page names	See the <i>EView/390 Administrator's Reference</i> for more information.
	Provides emphasis	You <i>must</i> follow these steps.
	Specifies a variable that you must supply when entering a command	At the prompt, enter rlogin <i>your_name</i> where you supply your login name.
	Parameters to a function	The <i>oper_name</i> 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 <code>grep</code> command ...
	Function names	Use the <code>opc_connect()</code> function to connect...
	File and directory names	<code>/opt/OV/bin/OpC/</code>
	Process names	Check to see if <code>opcmona</code> is running.
	Window/dialog box names	In the Add Logfile window...
Computer Bold	Text that you must enter	At the prompt, enter ls -l
Keycap	Keyboard keys	Press Return .
[Button]	Buttons on the user interface.	Click [Operator]. Click the [Apply] button.

Font	Meaning	Example
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 ...

Documentation Map

EView/390z Management for HP Operations Manager Unix/Linux (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 to find it.

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

EView/390z Printed Manuals

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

EView/390z Management for OpenView Operations Unix Concepts Guide

Explains EView/390 features, functions, architecture, and data flow. Describes EView/390 agent and server components, process management, SNA discovery process, network topology, and message windows.

EView/390z Management for HP Operations Manager Unix/Linux Installation Guide

Explains how to install and de-install EView/390, how to upload OS/390 installation files from the OM management server, update OS/390, NetView/390, and SOLVE:NETMASTER software, and start and stop EView/390.

EView/390z Management for HP Operations Manager Unix/Linux Administrator's Reference

Explains how to configure and use EView/390, detailed troubleshooting procedures, explanations of z/OS messages, and z/OS console commands.

EView/390z Online Information

The following information is available online:

- EView/390z Software Release

Installing and De-installing EView/390z

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

Installation Requirements

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

Hardware Requirements

For detailed hardware requirements for the OM management server and managed nodes, see the following manuals:

- **OM Management Server**
HP Operations Manager for UNIX 9.0 Administration UI Installation Guide
- **EView/390 Managed Node**
EView/390z Management Installation Guide

In addition to the requirements listed in these manuals, make sure that the systems selected as the OM management server and managed node meet the disk space requirements described in Table 3-1:

Table 3-1: Additional Disk-Space Requirements

Machine	Operating System	Disk Space
OM Management Server	HP-UX 11.31 or later (Itanium only) Sun Solaris 10 RedHat Linux Enterprise 5.3	20 MB
EView/390 Managed Node	z/OS V1R3 or later	60 tracks of 3390 DASD

Software Requirements

Before installing EView/390, make sure the following software is installed:

- **OM Management Server**
One of the following operating systems must be installed:
 - HP-UX 11.31 or later (Itanium Only)
 - Sun Solaris 2.10
 - RedHat Linux Enterprise Edition 5.3 (64-bit)

In addition, Operations Manager 9.0 or later must be installed.

- **EView/390 Managed Node**

Operating system z/OS V1R3 or higher must be installed.

In addition, IBM TCP/IP for MVS V3R1 or higher must be installed.

Obtaining License Keys

EView/390 requires a license key to be applied to each S/390 node that is to be managed by the OM server. License keys are applied to physical machines. That is, one license key can be used for multiple LPARs on the same 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 the z/OS console command line and reading the last five digits in the CPC ND or CPC SI lines.

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.02.0000000070571
CPC SI = 7060.H30.IBM.02.00000000000070571
CPC ID = 00
```

Verifying the Software Files

Before you install EView/390 on HP-UX or Solaris, make sure that you are installing the correct software files.

There is one software depot bundle for EView/390:

- EV390

This bundle contains the following datasets required to run EView/390:

EV390-CORE

Common core files:

- Software executables

EV390-CONF

Server configuration files:

- Message policies
- S/390 node configuration files

- `rc.config.d` system configuration files

EV390-DOC

Documentation files:

- EView/390z Administrator's Reference
- EView /390z Concepts Guide
- EView /390z Installation Guide
- EView /390z Software Release Notes

EV390-MF

Mainframe files:

<code>EV390.V63.ASM.SEQ</code>	Source assembler modules
<code>EV390.V63.CLIST.SEQ</code>	Command lists
<code>EV390.V63.LOAD.SEQ</code>	Load modules
<code>EV390.V63A.LOAD.SEQ</code>	Load module patches
<code>EV390.V63.SAMP.SEQ</code>	Configuration and startup samples

Installing EView/390z on the Management Server

Install the EView/390 software on the OM management server with one of the following UNIX operating systems:

- HP-UX
- Solaris
- RedHat Linux

 If you receive any warning or error messages during installation, you must resolve these problems before continuing with the installation.

Download the EView/390z Software Bundle

Contact EView Technology to receive the Internet link to download the EView/390z package. Alternatively, EView Technology will provide, upon request, an installable CD with the EView/390z software.

The installation filename is dependent on the OM management server's operating system:

<u>Operating System</u>	<u>Installation Bundle Filename</u>
HP-UX 11.31 (Itanium)	EV390.A.06.10.depot
Sun Solaris 2.10	EV390.A.06.10.tar.gz
RedHat Linux Enterprise Edition 5.3 (64-bit)	EV390V61-6.10-1.x86_64.rpm

Install EView/390z on a Management Server with HP-UX

The quickest and easiest way to install the EView/390 software bundle is with the Software Distributor (SD). The SD installation installs the product bundle you select, verifies that OM is installed, verifies that the database is correctly configured, and uploads the configuration into OM.

To install the EView/390 software bundle on an OM management server with HP-UX, follow these steps:

1. Login to the OM management server as **root** user.
2. Copy the EV390.A.06.10.depot file to a temporary directory such as /tmp on the OM management server.

3. Install the EView/390z software by doing **one** of the following:

- Use the `swinstall` command.

At the Unix root prompt, enter the `swinstall` command as follows:

```
swinstall -s <directory>/EV390.A.06.10.depot EV390Eng
```

where **<directory>** is the temporary directory where the depot file was copied to. The `swinstall` command installs the EV390 software bundle from the software depot and performs basic configuration. The software bundle contains all the EView/390 software, configuration files, and documentation.

- Use the `swinstall` GUI shown in Figure 3-1 on page 21.

To install the EView/390 software with the `swinstall` GUI, follow these steps:

- a. Set the `DISPLAY` variable for the machine on which you want to run the `swinstall` GUI.

- b. At the root prompt, enter the following command:

```
swinstall
```

- c. In the Specify Source window, set the Source Depot Type to Local Directory, and in the Source Depot Path field, enter the following:

```
<directory>/EV390.A.06.10.depot
```

where **<directory>** is the temporary directory where the depot file was copied to. Then click [OK].

- d. Highlight the EV390 entry.

Right-click the entry and select Mark for Install.

- e. Select Actions:Install...

The Install Analysis window appears.

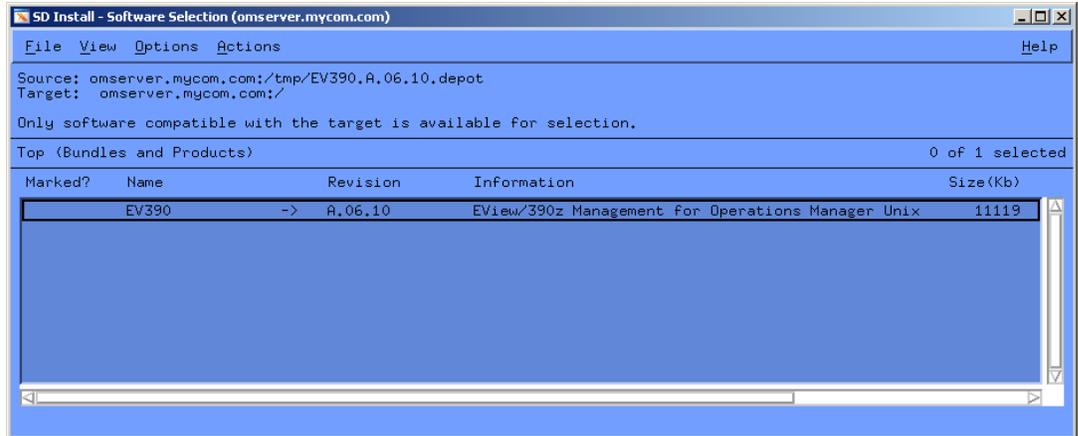
- f. When the analysis is completed, click [Logfile].

Scroll to the bottom of the file and ensure that you have no warnings or errors, then click [OK].

- g. In the Install Analysis window, click [OK].

The installation process starts.

- h. After the install process completes, select File:Exit in the SD Install Software Selection window.

Figure 3-1: Installing Software with the swinstall GUI

Install EView/390z on a Management Server with Solaris

To install the EView/390 software bundle on an OM management server with Solaris, follow these steps:

1. Login to the OM management server as **root** user.
2. Copy the downloaded EV390.A.06.10.tar.gz file to a temporary directory such as /tmp on the OM management server.
3. Unzip the .tar file using the command:
gunzip EV390.A.06.10.tar.gz
4. Extract the “EView390” directory structure from the .tar file using the command:
tar -xvf EV390.A.06.10.tar
5. Use the Solaris pkgadd command to install the product:
pkgadd -d /tmp EView390

where /tmp is the name of the temporary directory where the installation file was copied to.
6. Delete the EView390 directory structure from the temporary directory.

Install EView/390z on a Management Server with Linux

On Linux, the Redhat Package Manager (RPM) is used for installation.

To install the EView/390z software package on an Operations Manager server with Red Hat Enterprise Linux, follow these steps:

1. Login to the OM server as **root** user.
2. Download or transfer the installation package file to a directory such as /tmp on the OM management server.

3. Enter the rpm command as follows:

```
rpm --install /tmp/EV390V61-6.10-1.x86_64.rpm
```

where */tmp* is the name of the temporary directory where the installation file was copied to.

Verify the Installation on the Management Server

To verify that the installation of the EView/390 on the OM management server was successful, follow these steps:

1. Start the vp390elli process:

```
/opt/OV/bin/ovc -start vp390elli
```

2. Verify that the vp390elli process started under OpenView.

Enter the following command:

```
/opt/OV/bin/ovc -status
```

3. Start the Operations Manager Admin GUI from a browser:

```
http://OMservername:9662
```

4. Log in with the administrator ID.

5. Verify instrumentation:

- a. Verify that the 390 node group exists.
- b. Verify that the Mainframe Tools exists.
- c. Verify that the vp390_adm and vp390_op profiles are in the User Profile Bank.
- d. Verify that the EV390 and the EV390 Samples groups are in the listed under policy groups.

6. If any of the previous steps fails to produce the expected results, do the following:

- a. Check log files in the /var/opt/OV/log/vp390 directory for errors.
- b. Check log files in the /var/adm/sw directory for errors.

Installed File Locations on the Management Server

The installation process copies the necessary files to the OM management server. The directories created for the EView/390 on the OM management server (if not already existing) are shown in Table 3-2.

Table 3-2: EView/390 File Locations on the OM Management Server

File Type	Directory
-----------	-----------

File Type	Directory
Binary and Script	/opt/OV/vp390/bin
Configuration	/etc/opt/OV/share/conf/vp390
Mainframe	/opt/OV/vp390/mf
Process and Command Log	/var/opt/OV/log/vp390
Language Localization	/opt/OV/vp390/local
Temp	/var/opt/OV/share/conf/vp390
Performance Data Collection	/var/opt/OV/vp390/datafiles

Installing EView/390z on the Managed Nodes

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

What to Upload

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

```
/opt/OV/vp390/mf
```

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

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

File Name	Description	3390 DASD Tracks	Required?
EV390.V63.LOAD.SEQ	EView/390 agent executables	25	Yes
EV390.V63.SAMP.SEQ	Sample JCL, startup parameter cards, and VTAM definitions	5	Yes
EV390.V63.CLIST.SEQ	Command lists used by NetView/390 and SOLVE:NETMASTER for executing mainframe commands	2	Only if NetView/390 or SOLVE:NETMASTER is installed
EV390.V63.ASM.SEQ	Assembler source code for onsite assembly of NetView/390 and CICS exits	8	Only if NetView/390 or CICS is installed
EV390.63A.LOAD.SEQ	EView/390 agent executable patched files	20	Yes

Transferring Files to the Mainframe

Use FTP to send the several files from the OM server to the S/390. Use binary mode when transmitting the files, and use the SITE or LOCSITE command to create the target datasets with attributes DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120) with a primary 3390 dataset size of 35 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:

```
# cd /opt/OV/vp390/mf
# 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.V63.LOAD.SEQ hlq.LOAD.SEQ
ftp> put EV390.V63.SAMP.SEQ hlq.SAMP.SEQ
ftp> put EV390.V63.CLIST.SEQ hlq.CLIST.SEQ
ftp> put EV390.V63.ASM.SEQ hlq.ASM.SEQ
ftp> put EV390.V63A.LOAD.SEQ hlq.LOAD.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 3-3.

Extracting Partitioned Datasets from Sequential Datasets

After uploading the files to sequential datasets on the mainframe, use the 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.V63” enter the following:

```
DA('EV390.V63.SAMP')
```

After loading the EV390.V63A.LOAD dataset, copy all members to the EV390.V63.LOAD dataset.

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

De-installing EView/390z

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

- OM GUI
- OM management server
- OM managed nodes

Remove EView/390z Components from the OM GUI

You must manually remove EView/390z components from the Operations Manager (OM) Admin GUI.

To remove EView/390z components from the OM Admin GUI, follow these steps:

1. From the `Node Bank` window, select and delete each defined `S/390` mainframe node.
2. From the `All Node Groups`, select and delete the `390 Node Group`.
3. From the `Tool Bank`, select and remove the `EV390:Mainframe Tools` applications and application group.
4. From the `Policy Bank`, remove the `EV390` and `EV390 Samples` policies, then re-distribute the remaining templates to the OM management server.
5. Remove the `vp390_adm` and `vp390_op` user profiles.
6. Stop all EView/390 processes running on the OM server:

```
/opt/OV/bin/ovc -stop vp390elli
```

```
/opt/OV/vp390/bin/vp390sv -stop
```

7. Delete any configuration files for any mainframe nodes added:

```
cd /etc/opt/OV/share/conf/vp390
```

```
rm evopen_config_*
```

For more information about removing elements from the OM Admin GUI, see the *HP OpenView Operations Administrator's Reference*.

Remove EView/390z from the OM Management Server

To remove the EView/390 components from the OM management server and complete the general clean up process, enter one of the following:

- On HP-UX, enter the following command:
`swremove EV390`
- On Solaris, enter the following command:
`pkgrm EView390`
- On Linux, enter the following command:
`rpm -erase EV390V61`

Remove EView/390z from the OM Managed Nodes

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

1. Stop the EView/390 job on the S/390 managed node.

To find out how to stop the EView/390 job on the managed nodes, see "Running EView/390 as a Started Task" on page 72.

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

Updating Mainframe Software

This chapter contains instructions for updating OS/390 and VTAM resources on the mainframe, and updating the EView/390 Management 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 the active `TCPIP.DATA` file

Reserving Port Numbers in `PROFILE.TCPIP`

Choose two available port numbers for use by EView/390 and add them to 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 jobname for EView/390 is “VP390”.)

 This step is optional. If specific port numbers are not reserved for EView/390 use, the OM management server connection will still succeed, but this reservation ensures that other mainframe applications will not use the ports needed for EView/390.

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 8 when defining the `SYSTCPD DD` card in the startup JCL job.

Phase 2: Updating z/OS and VTAM

To run EView/390, modify the z/OS and VTAM 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 *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/390 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*, enter the following z/OS console 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 VP390 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/390 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 console command:

```
SET ICS=xx
```

where *xx* is the two-digit suffix of the IEAICSxx 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 console command:

```
SET SCH=xx
```

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

Adding an Entry to the RACF Class

The VP390 task requires an OS/390 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 VP390 to run as a started task. If 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 STARTED 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

```

ICHRIN03 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/390 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 (described in Chapter 4 of the *EView/390 Administrator's Reference*), follow one of these two options, depending on your mainframe configuration:

- If RACF is being used for SDSF security and the ISFCMD class is active, use the following RACF commands:

```

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)

```

where *userid* is the ID that the VP390 job is running under (which was named in the RDEFINE statement above).

- If the SDSF server address space is running, edit PARMLIB (ISFPRMxx) and add another GROUP **ahead** of the default ISFUSER profile. 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 example 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

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 console command:

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

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 into your VTAMLST library:
`hlq.SAMP(EVAPPL)`
2. If NetView/390 or SOLVE:NETMASTER is running in this S/390 LPAR, comment out the PPO and CNM definitions.
3. Add EVAPPL to the list of auto-activated major nodes in the following:
`SYS1.VTAMLST(ATCCONxx)`
4. Activate the major node with the z/OS console command:
VARY NET,ACT,ID=EVAPPL
5. Verify that the APPL definitions are active/connectable with the z/OS console command:
D NET,ID=EVAPPL,E

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

Phase 3: Updating VP390 Parameter Cards

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

The EView/390 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 use the value of &SYSNAME. in the name of the console defined for the CMD subtask, the parameter card would look like:

CMD &SYSNAME.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/390 with the ability to issue z/OS (MVS) console commands.

Valid Parameters

<Console Name> [LOG|NOLOG] [HC={YES|NO}]

Sample Syntax

```
CMD EVOCONS2 LOG HC=YES
```

Description

This card will initialize a CMD subtask to process MVS commands through an MCS console.

Parameters

Console Name	Required. Specify a 1-8 character name for the extended MCS console you wish to define for issuing z/OS (MVS) commands from EView/390. If this name is defined in RACF, the OPERPARM values in the RACF entry are used for the console definition. Otherwise, a console is defined with default parameters AUTH=MASTER and ROUTCDE=NONE.
LOG <u>NOLOG</u>	Optional. Specifying LOG will cause all MVS commands entered from an EView/390 server to be 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 <u>NO</u> }	Optional. Specifying HC=YES will cause all commands and responses from the MCS console to be written to the hardcopy log. HC=NO will suppress the writing of commands and command responses to the hardcopy log. HC=NO is the default.

DELAY Parameter Card

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

Valid Parameters

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. SPO subtasks might be restarted immediately. A TCP subtask may require time to reset the port through which the workstation is connected. The default DELAY value is 60 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 MVS messages with the “Automation Requested” bitflag set.

Parameters

YES <u>NO</u>	“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 OM 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 Parameters

{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” bitflag 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 OM 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 Parameters

{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” bitflag 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 OM 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 Parameters

{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 OM 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 OM server. “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 OM 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 Parameters

{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 OM server.

Parameters

YES <u>NO</u>	“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 OM server. “NO” indicates that the WTO user exit suppression flag will be ignored, but the message ID must still be present in the message filter table to be passed on to the OM 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 OM server.

Valid Parameters

```
MSG {[+|-]msgid [JOBNAME=(jobname[,jobname]...)]
[JOBID=(jobid[,jobid]...)] [REXX=script]} ...
```

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
```

Description

The FILTER card adds message IDs to the message filter table.

z/OS messages must have a match in the message filter table to be forwarded to the OM management server. The message table holds up to 2,000 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

<i>msgid</i>	The message ID of the message to pass to the OM 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 OM 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 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. A *msgid* can be up to 15 characters in length.

<i>jobname</i>	The job name of the job that issued the message. The same wildcarding rules used in <i>msgid</i> may be applied to this parameter. A <i>jobname</i> may be up to 8 characters in length.
<i>jobid</i>	The job ID of the job that issued the message. The same wildcarding rules used in <i>msgid</i> may be applied to this parameter. A <i>jobid</i> may be up to 8 characters in length.
<i>script</i>	A REXX program to be called if this message is encountered. The program 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 program must be available in the dataset specified by the DD card named on the RMA subtask card (see page 50).

FILTERTABLE Parameter Card

Specify the size of the message filter table.

Valid Values

MSGSIZE=*entries*

Sample Syntax

```
FILTERTABLE MSGSIZE=3000
```

Description

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

Parameters

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

MVS Parameter Card

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

Valid Parameters

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

Sample Syntax

```
MVS EVOCONS1 DOM DOMLIST=800 ROUT=22,116-128 MON=NAME,SESS AUTO=NO
UD=NO QL=15000 QLP=70 HILITE=ALL MSCOPE=LPARA,LPARB
```

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 OM server.

Parameters

Console Name	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.
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 flow. However, the IST parameter is not recommended unless both the PPO and the PPI interfaces are unavailable. (If the IST parameter is used in conjunction with either the PPO or PPI subtasks, the same VTAM message may be sent to the OM server twice.)
DOM	Optional. Instruct the console to capture MVS "Delete Operator Messages" and forward them to the OM 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, disclosing the MVS message key or token of the deleted message(s). This

information may be used for OM 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 OM server, and then only forwards DOMs which match an entry in this list. <i>size</i> is the number of message IDs and tokens to be saved in the list. If no <i>size</i> is specified, the DOMLIST defaults to 1000 messages. Use this parameter if the OM server is receiving a large number of DOM messages which do not match any relevant message on the OM browser. The DOMLIST parameter is valid only if the DOM parameter is also specified.
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.
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.
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, this console will capture messages even if they were previously processed by MPF automation.
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.
HC	Optional. Specifies whether this console should receive all messages destined for the hardcopy console. The default setting is NO.
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, <i>limit</i> 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 OM server. <i>percentage</i> may be between 1 and 100. Message EVO703 is sent to the OM server when the console's queue has met or exceeded this level. When the queue drops below the specified <i>percentage</i> , message EVO704 is sent

to the OMserver to inform that the backlog has been relieved. If not specified, VP390 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
```

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 OM 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 OM server.
- ONLY Only highlighted messages with their message ID in the message filter table will be forwarded to the OM server. (To send all highlighted messages, add an asterisk entry to the message filter table.)
- ALL All highlighted messages will be forwarded to the OM 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.

MSCOPE

Optional. Allows you to specify which systems in a sysplex will send messages to the EView console. Set the MSCOPE parameter to one of these options:

- * Messages will only be received from the local system on which the EView/390 job is running. This is the default.
- *ALL Messages will be received from all systems in the sysplex. (The optional variable \$OPTION(systemname) will hold the name of the originating system when sending messages to OM).
- name, . . .* Only messages from the named systems will be received by EView. The *name* values must be no more than eight characters long, with a comma between each. A maximum of eight names is allowed.

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

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

NOMATCHLOG Parameter Card

Identify the logging dataset(s) that record the messages that are not forwarded to the OM Server.

Valid Parameters

```
log1 [log2 ...]
```

Sample Syntax

```
NOMATCHLOG NMLOG1 NMLOG2 NMLOG3
```

Description

This card will initialize a NOMATCH subtask, which is responsible for writing any mainframe messages that were not passed on to the OM server due to a defined filtering restriction. The logging datasets will be written in the following format:

Table 3-1: Data in the NOMATCH Message Log

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/390 message filter table.
3-9	Date of message in Julian format: YYYYDDD
11-18	Time of message in 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 names 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 OS/390 Operating System information and statistics.

Valid ParametersSDSFMAX=*n***Sample Syntax**

OSINFO SDSFMAX=400

Description

Use the OSINFO card to initialize a subtask that will accept command type 46 requests from the OM server and return information about z/OS jobs and performance statistics. See the *EView/390 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

<i>n</i>	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 OVOPCMMSG calls and passes them on to the OM 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 OVOPCMMSG calls from other mainframe scripts to be forwarded to the OM server. This port number can be defined here or in the /etc/vp390info file with the identifier EV390_OPCMSG_PORT. See the *Administrator's Reference* for information on sending OPCMSG commands from the mainframe agent to the OM server.

Parameters

<i>portnum</i>	Optional. An integer value between 1-65535 indicating the TCP/IP port number to listen to for OVOPCMMSG data, but the preferred place to specify the port number is in the /etc/vp390info file. A port number specified on this card will override a number specified in the /etc/vp390info file. If no port number is specified on this card or in the /etc/vp390info file, the port number defaults to 6108.
----------------	--

PERF Parameter Card

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

Valid Parameters

INTERVAL=*n*

Sample Syntax

PERF INTERVAL=15

Description

Use the PERF card to initialize a subtask that will send RMF data to the OM server at a defined interval. On the OM server, the RMF data will be directed to the CODA subagent or the OV Performance Agent, if available.

Parameters

<i>n</i>	An integer value indicating how often, in minutes, the RMF data is sent to the OM server. The default is 15 minutes.
----------	--

PPI Parameter Card

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

Valid Parameters

[BUFLLEN=*n*]

Sample Syntax

PPI

Description

This card will initialize a PPI subtask. Add this card to connect EView/390 to the NetView/390 or SOLVE:NETMASTER PPI for the receipt of VTAM messages and alerts. The PPI must be active in accordance with the NewView/390 or SOLVE:NETMASTER documentation. This executable is responsible for communicating with the NTIPPI executable that 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 and CNM connections to VTAM information. The PPI subtask should only be used when a 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 and CNM parameter cards 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/390 to act as the VTAM Primary Program Operator.

Valid Parameters

<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/390 is running in combination with other network 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/390 application major node is "EVOPPO1". (See "Adding the <i>hlq</i> .SAMP(EVAPPL) Definition" on page 35.)
ECHO	Optional. If used, it directs EView/390 to forward a copy of the VTAM messages received to the operator console.

PRINTCARDS Parameter Card

Write the SYSIN cards to the SYSPRINT.

Valid Parameters

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 Parameters

1 to 65535, or UNLIMITED

Sample Syntax

```
RESTART 100
```

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 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 that 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=OVEXEC
```

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 40). 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 OM 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 Parameters

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 Parameters

<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/390 has been tested 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 OM application based on the response from RACF or ACF2.

SMFBUFFER Parameter Card

Define the size (in bytes) of the memory buffer that holds information received from RMF.

Valid Parameter

260 to 2000000000

Sample Syntax

SMFBUFFER 5000000

Description

If the PERF or OSINFO subtasks are defined, the data requested from RMF will be stored in a memory buffer in the VP390 address space. Depending on the size of the mainframe, the size of this memory buffer may need to be increased to accommodate larger data reports from RMF. Use this parameter card to increase the size of the SMF buffer if, during the course of the VP390 job run, you receive an EVO131 message with a code of -104, indicating that the SMF buffer was not big enough to hold all of the statistics coming from SMF. (See also the SMFBUFFER console command in the *EView/390 Administrator's Reference* to change the size of the buffer while the VP390 job is running.) Note that the VP390 job's region size may need to be increased if the SMFBUFFER value is set too high. The default SMFBUFFER size is 1000000.

SPO Parameter Card

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

Valid Parameters

<VTAM APPL Definition>

Sample Syntax

SPO EVOSPO1

Description

This card will initialize a SPO subtask with the ID of a VTAM 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 EView/390 server. This subtask executable is responsible for initializing a SPO ACB to VTAM, then receiving VTAM commands (for example, `Vary` or `Display`) from OM management servers, sending the commands to VTAM over the SPO, and sending the VTAM responses to the EView/390 server that initiated the command. Multiple SPO subtask cards may be defined to distribute the work if several commands come in at nearly the same time from different OM operators.

Parameters

VTAM APPL Definition	Required. Specify the name of a VTAM APPL definition coded with <code>AUTH=SPO</code> . The default APPL provided in the EView/390 application major node is "EVOSPO1". (See "Adding the <i>hlq</i> .SAMP(EVAPPL) Definition" on page 35.)
-------------------------	--

TCP Parameter Card

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

Valid Parameters

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

Sample Syntax

```
TCP 6106 6107 BUFDD=BUFFR1,BUFR2 ACK=5 LIMIT=20 HB=30 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/390 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. It is generally necessary to have one TCP subtask defined for each OM server that will be connecting to the mainframe. Multiple TCP cards are allowed.

Parameters

<i>mmsport</i>	Required. Port number opened on the mainframe for establishing a socket connection with the Master Message Server task on the OM management server. This number must match the <code>EVOMF_HCI_AGENT_PORT</code> value entered when adding the S/390 node to OM.
----------------	--

<i>cmdport</i>	Required. Port number opened on the mainframe for establishing a socket connection with the Command Server task on the OM management server. This number must match the <code>EVOMF_CMDS_AGENT_PORT</code> value entered when adding the
----------------	--

S/390 node to OM.

<i>hlq</i>	Optional. 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 TCP subtask will not initialize if the <i>hlq</i> is misstated.
<i>dd1, dd2</i>	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).
<i>ack</i>	Optional. The number of unsolicited mainframe messages that will be passed on to the OM server before an acknowledgment is expected from the OM server. By default, the VP390 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 OM 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 OM server, and these messages may appear duplicated in the OM browser. If this causes a problem, set the ACK parameter to 1 to ensure an acknowledgment from the OM server after every message is sent. The ACK parameter is only valid if BUFDD is specified.
<i>limit</i>	Optional. The age limit (in minutes) of buffered messages that the mainframe agent will send to the OM server. By default, messages read from the buffering files that are over 20 minutes old will not be forwarded to the OM server. Set this value to 0 to receive all buffered messages regardless of their age. The LIMIT parameter is only valid if BUFDD is specified.
<i>hb</i>	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 OM server and the mainframe agent every 30 seconds.
<i>bindaddr</i>	Optional. Use the BINDIP option to specify which address the TCP ports should bind on (in cases where a system may be known by multiple IP addresses). <i>ipaddr</i> must be specified in the nnn.nnn.nnn.nnn IP address format.
<i>servaddr</i>	Optional. Use the SERVERIP option to specify a specific OM server IP address that the listening ports will accept a connection from. <i>servaddr</i> must be specified in the nnn.nnn.nnn.nnn IP address format. Additionally, the <i>servaddr</i> can be masked to allow a range of IP addresses to connect to the ports. To specify a mask, add a slash (/) after the <i>servaddr</i> 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, "16" would be equivalent to 255.255.0.0).

Updating NetView/390

This chapter explains the updates required for EView/390 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/390. 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/390 uses three 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, alerts, and command responses from the other four 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 four 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 OM 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.

Load Module Name	Description
NTIITCI	This program is used as the NetView XITCI exit, capturing CNM (SNA alert) data such as NMVT, RECMS, and RECFMS hardware alerts. When the alert data is forwarded to the OM server, the alert summary will be shown in the OM browser, similar to NetView's NPDA Alerts Dynamic screen. A translation program provided on the OM server, <code>vp390alertxlate</code> , parses out and describes the alert data.
NTIMQS	This program is used by CLISTs running under a NetView autotask. The CLISTs execute NetView commands sent to the mainframe from the OM 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 OM 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.

-  **NOTE** 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 DSIDMN as follows:

1. Define the EView/390 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 VP390.

Updating DSICRTTD

Define the mainframe alert collection exit by adding the following definition to the DSICRTTD member of your NetView/390 DSIPARM dataset:

```
DSTINIT XITCI=NTIITCI
```

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 the 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 the 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 OM management server. For details about EView/390 configuration parameters on the OM 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.

Make sure `EVOPROF` contains 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 `hlq.CLIST` 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.

Updating SOLVE:NETMASTER

This chapter describes the updates required for EView/390 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

SOLVE:NETMASTER includes the following dataset members:

NMASTDOC	Documents procedures and how to implement EView/390.
NETTKCNM	Documents changes to CNMPROC for EView/390.
NETTKPPO	Documents changes to PPOPROC for EView/390.
NETTKPPI	Sends PPI messages (CNM and PPO) to EView/390.
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/390 dataset:

`hlq.CLIST`

Phase 1: Verifying the Subsystem Interface Installation

The PPI is necessary for cross-memory communications between SOLVE:NETMASTER and EView/390. 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 CNMPROC

To enable EView/390 to receive alert information from SOLVE:NETMASTER, add the Network Control Language (NCL) code in `hlq.CLIST(NETTKCNM)` to the production CNMPROC at a point where all CNM flow can be seen. Add this code immediately after the mainline `&CNMREAD`. In the distributed CNMPROC (`$NWCNMPR`), the label `".READOK"` is the point determining the `&CNMREAD` to be successful. Make sure to insert the NCL code immediately after `".READOK` label.

Phase 3: Updating PPOPROC

To enable EView/390 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 4: Copying Members to SOLVE:NETMASTER Libraries

Copy the following three members from EView/390's `hlq.CLIST` into a SOLVE:NETMASTER `COMMAND DD` dataset:

- `NETTKCMD`
- `NETTKCM1`
- `NETTKPPI`

Phase 5: Updating PPI

`NETTKPPI` and `NETTKCMD` are the primary PPI procedures that send CNM and PPO data through the PPI and wait for commands coming from EView/390 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 6: 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 EVOOPEN, 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/390.



Updating CICS

This chapter explains the updates required for EView/390 to capture messages generated by a CICS region.

Phase 1: Identify CICS Messages in XMEOUT Code

EView/390 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/390 MCS console will then capture these messages and pass them on to the OM server.

By default this XMEOUT exit redirects all CICS messages to the console. Restricting the messages that 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.

Phase 2: Assembling and Linking the CICS XMEOUT 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.

Phase 3: Activating the XMEOUT 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 in to a CICS Program Load Table.

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)
```


Starting and Stopping the Mainframe Component

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

Running NetView/390 Automatically

The EView/390 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 the `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 CNM and PPO data through the PPI, or await commands coming from VP390 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 issue from an OCS console.

Running EView/390 as a Started Task

The EView/390 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 an 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/390 as a Batch Job

The VP390 job can be run as a batch job.

To Start the VP390 Job as a Batch Job

To start 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
```


ACB

Application Control Block. Data area opened for communication with VTAM.

See also VTAM.

ACF2

Active Communications Functions 2. Mainframe security package comparable to RACF.

See also RACF.

Active Communications Functions 2

See ACF2.

APF

Authorized Program Facility. Facility permitting identification of programs authorized to use restricted functions.

Application Control Block

See ACB.

Authorized Program Facility

See APF.

CA

See SOLVE:NETMASTER.

central processing unit

See CPU.

CDRSC

Cross-domain resource. In VTAM programs, synonym for other-domain resource.

See also VTAM.

CNM

Communication Network Management. Generation and processing of hardware alerts.

See also CNMPROC.

CNMPROC

SOLVE:NETMASTER NCL procedure used to intercept CNM records across the VTAM CNM interface.

See also CNM; NCL; VTAM.

Communication Network Management

See CNM.

Computer Associates

See SOLVE:NETMASTER.

CPU

central processing unit. Part of computer with circuits that controls the interpretation and execution of instructions.

cross-domain resource

See CDRSC.

DASD

direct access storage device. Also known as “disk pack” or “disk drive.” Device in which access time is effectively independent of the data location.

Data Base 2

See DB2.

data definition card

See DD Card.

Data Service Task

See DST.

DB2

Data Base 2. Relational database management system from IBM.

DD Card

Data definition card. Data definition statement used in JCL to associate physical data or datasets with logical dataset names defined by the running program.

See also JCL.

direct access storage device

See DASD.

disk drive

See DASD.

disk pack

See DASD.

domain

In SNA, a discrete mainframe processor, along with all of its PUs, LUs, and other associated resources controlled by a single VTAM.

See also LU; PU; SNA; VTAM.

DST

Data Service Task. NetView/390 program subtask that gathers, records, and manages data in a VTAM file or a network device containing network management information.

See also VTAM.

high-level qualifier

See HLQ.

HLQ

High-level qualifier. Portion of a dataset name up to the first period.

HP OpenView Windows

See OVW.

IEBCOPY

MVS utility batch job used to copy datasets or dataset members from one medium to another.

See also MVS.

Initial Program Loader

See IPL.

IPL

Initial Program Loader. Also known as “system restart” or “system startup.” 1. Initialization procedure that causes an operating system to begin operation. 2. Process by which a configuration image is loaded into storage at the beginning of a workday or after a system malfunction. 3. Process of loading system programs and preparing a system to run jobs.

JCL

Job Control Language. Language used to identify a job to an operating system and to describe the job’s requirements.

See also DD Card.

JES

Job Entry Subsystem. Also known as “JES2” or “JES3.” Set of programs that control customer application submissions.

JES2

See JES.

JES3

See JES.

Job Control Language

See JCL.

Job Entry Subsystem

See JES.

Legacy Link Interface

See LLI.

LLI

Legacy Link Interface. OM option that allows external processes to connect to OM action and message managers.

logical unit

See LU.

LU

Logical unit. 1. In SNA, a port through which end users access the SNA network to communicate with other end users, and through which end users access the functions provided by SSCPs. This port can support at least two sessions, one with an SSCP and one with another port, and may be capable of supporting many sessions with other ports. 2. In general, a type of network addressable unit that enables end users to communicate with each other and gain access to network resources.

See also domain; PU; SNA; SSCP.

MCS

Multiple Console Support. Method of programmatically defining a z/OS console for command and message support.

MQSeries

Message Queuing Series.

Multiple Console Support

See MCS.

Multiple Virtual Storage

See MVS.

MVS

Multiple Virtual Storage. 1. MVS/390 operating system. 2. MVS/XA product. 3. MVS/ESA product. 4. OS/390 product. 5. z/OS product.

See also IEBCOPY.

NCL

Network Command List. Command list used in SOLVE:NETMASTER.

See also CNMPROC; PPOPROC; SOLVE:NETMASTER.

NCP

Network Control Program. Licensed program from IBM that provides communication controller support for single-domain, multiple-domain, and interconnected networks.

NetView/390

Licensed program from IBM/Tivoli used to monitor, manage, and diagnose problems with a VTAM network.

See also NTIPPI; VTAM.

Network Command List

See NCL.

Network Control Program

See NCP.

Network Node Manager

See NNM.

NNM

Network Node Manager. Comprehensive Hewlett Packard network management solution that discovers network devices, and provides a map to illustrate the structure of the network and the status of devices and segments. When a major device fails, the event correlation engine evaluates the event stream to pinpoint the root cause of the failure. The manager also helps identify potential trouble spots before a failure occurs.

NTIPPI

Networking Program-to-Program Interface. EView/390 message and command interface to NetView/390.

See also NetView/390; PPI.

OCS

Operator Control Services. SOLVE:NETMASTER component that provides general operational control and an advanced operator interface to VTAM for network management.

See also SOLVE:NETMASTER; VTAM.

OpenView NNM

See NNM.

OpenView Windows

See OVW.

Operator Control Services

See OCS.

OVW

OpenView Windows. Customizable OpenView network management GUI.

physical unit

See PU.

PPI

Program-to-Program Interface. Interface that allows data buffers to be sent between programs running in different address spaces.

See also NTIPPI.

PPO

Primary Program Operator. Operator application program that is authorized to receive unsolicited VTAM messages. When the authorized application program is active, all unsolicited messages go to this authorized application program. Conversely, when it is inactive, unsolicited messages go to the system console. There can be only one such authorized application program in any domain.

See also PPOPROC; SPO; VTAM.

PPOPROC

SOLVE:NETMASTER NCL procedure used to intercept unsolicited VTAM (PPO) messages.

See also NCL; PPO; SOLVE:NETMASTER; VTAM.

Primary Program Operator

See PPO.

Program-to-Program Interface

See PPI.

PU

Physical unit. In SNA, the component that manages and monitors the resources (for example, attached links and adjacent link stations) associated with a node, as requested by an SSCP. An SSCP activates a session with the component to indirectly manage,

through the component, resources of the node (for example, attached links). The term applies to type 2.0, type 4, and type 5 nodes only.

See also domain; LU; SSCP.

RACF

Resource Access Control Facility. Licensed IBM program providing user and resource authorization security.

See also ACF2.

Resource Access Control Facility

See RACF.

Secondary Program Operator

See SPO.

server

1. In general, a functional unit that provides shared services or facilities to workstations over a network (for example, a file server, a print sever, or a mail server). 2. In the UNIX operating system, an application program that usually runs in the background and is controlled by the system program controller.

SNA

System Network Architecture. Network architecture that enables the reliable transfer of data among end users, and provides protocols for controlling the resources of various network configurations.

See also domain; LU.

SOLVE:NETMASTER

Licensed program from Computer Associates/Sterling Software that is used to monitor and manage a VTAM network.

See also NCL; OCS; PPOPROC; VTAM.

SOLVE:NETMASTER Operator Control Services

See OCS.

SPO

Secondary Program Operator. VTAM Operator application program that is not authorized to received unsolicited messages. This unauthorized application program can receive only messages generated by the commands it issues. There can be more than one such unauthorized application program in a domain, in addition to the PPO.

See also PPO.

SSCP

System Services Control Point. Focal point of a SNA network for managing network resources.

See also LU; PU.

STC

System-defined subsystem in SYS1.PARMLIB(IEAICSxx) that holds names of address spaces initiated by START or MOUNT commands.

Sterling Software

See SOLVE:NETMASTER.

SYSIN

System Input. Sequential file or partitioned dataset member that stores input data for a mainframe job.

System Input

See SYSIN.

System Network Architecture

See SNA.

System Restart

See IPL.

System Services Control Point

See SSCP.

System Startup

See IPL.

TCP

Transmission Control Protocol. Communications protocol used in the Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. This protocol provides reliable host-to-host communication between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

See also TCP/IP.

TCP/IP

Transmission Control Protocol/Internet Protocol. Set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

See also TCP.

Transmission Control Protocol

See TCP.

Transmission Control Protocol/Internet Protocol

See TCP/IP.

Virtual Telecommunications Access Method

See VTAM.

VTAM

Virtual Telecommunications Access Method. Set of programs that maintain control of the communication between terminals and application programs running on SNA networks.

See also ACB; CDRSC; CNMPROC; domain; DST; NetView/390; OCS; PPO; PPOPROC; SOLVE:NETMASTER.