HP OpenView
OS/390 Management

Administrator's Reference

HP OpenView

Manufacturing Part Number: B9122-90010

Version A.05.00

December 2003

Legal Notices

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard 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.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend. All rights are reserved. No part of this document may be copied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

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

HEWLETT-PACKARD COMPANY
United States of America

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, 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 trademark of Sun Microsystems, Inc.

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

Contents

Contents

Contents

Conventions

The following typographical conventions are used in this manual.

Table 1 Typographical Conventions

Font	Meaning	Example
Italic	Book or manual titles, and man page names	See the *HP OV OS/390 Management Administrator's Reference* for more information.
	Provides emphasis	You *must* follow these steps.
	Specifies a variable that you must supply when entering a command	At the prompt, enter `rlogin` *`your_name`* where you supply your login name.
	Parameters to a function	The *`oper_name`* parameter returns an integer response.
Bold	New terms	The **monitor agent** observes...
`Computer`	Text and items on the computer screen	The system replies: `Press Enter`
	Command names	Use the `grep` command ...
	Function names	Use the `opc_connect()` function to connect...
	File and directory names	`/opt/OV/bin/OpC/`
	Process names	Check to see if `opcmona` is running.
	Window/dialog box names	In the `Add Logfile` window...
`Computer Bold`	Text that you must enter	At the prompt, enter **`ls -l`**
Keycap	Keyboard keys	Press **Return**.
`[Button]`	Buttons on the user interface.	Click `[Operator]`. Click the `[Apply]` button.

Table 1 Typographical Conventions

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

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

NOTE

In addition to OV OS/390 documentation, related OpenView products provide a comprehensive set of manuals that help you use the products and improve your understanding of the underlying concepts.

OV OS/390 Printed Manuals

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

HP OpenView OS/390 Management Concepts Guide

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

HP OpenView OS/390 Management Installation Guide

Explains how to upload OV OS/390 installation files from the VPO management server, update OV OS/390, NetView/390, and SOLVE:NETMASTER software, and start and stop OV OS/390.

HP OpenView OS/390 Management Administrator's Reference

Explains how to install, de-install, configure, and use OV OS/390. Includes detailed troubleshooting procedures and explanations of OV OS/390 system messages. Also describes OS/390 console commands

OV OS/390 Online Information

The following information is available online:

- *OV OS/390 Software Release Notes*

1 Configuring OV OS/390

This chapter describes how to configure HP OpenView OS/390 Management (OV OS/390).

Phase 1: Adding Mainframe Nodes to the Node Bank Window

The `vp390addagt` program adds a new mainframe computer to be monitored by OVO. After you enter some identifying information about the mainframe computer, `vp390addagt` adds the mainframe to the `Node Bank` and places it under the 390 node group.

To add mainframe nodes to the `Node Bank` window, follow these steps:

1. Start the "Add Agent" application.

   Do one of the following:

   - From the `Applications Bank`, select `VP390 Tools`, then select `Add 390 Node`.
   - From a UNIX command line, enter the following command:

     `/opt/OV/vp390/bin/vp390addagt`

2. Enter the local IP name of the mainframe to be added.
3. Configure the OV OS/390 server processes.

   Do one of the following:

   - *Default Configuration*

     Press **Enter** to accept the default values for the OV OS/390 processes.

   - *Custom Configuration*

     Enter the configuration values for the following OV OS/390 processes:

     — *Path Parameters*

     For valid values, see "About Path Parameters" in this chapter.

     — *Agent Address Parameters*

     For valid values, see "About Agent Address Parameters" in this chapter.

     — *Agent Port Parameters*

     For valid values, see "About Agent Port Parameters" in this chapter.

     — *Configuration and Status Parameters*

     For valid values, see "About Configuration and Status Parameters" in

this chapter.

— *Mainframe Command Parameters*

For valid values, see “About Mainframe Command Parameters” in this chapter.

— *Miscellaneous Parameters*

For valid values, see “About Miscellaneous Parameters” in this chapter.

Phase 2: Assigning and Distributing the VP390 Templates

In this phase of the OV OS/390 configuration process, you assign and distribute the VP390 templates to the agent component of the OVO management server, which then acts as the agent for the S/390.

NOTE You must assign VP390 templates to the agent component of the OVO management server before you can distribute them.

To Assign VP390 Templates

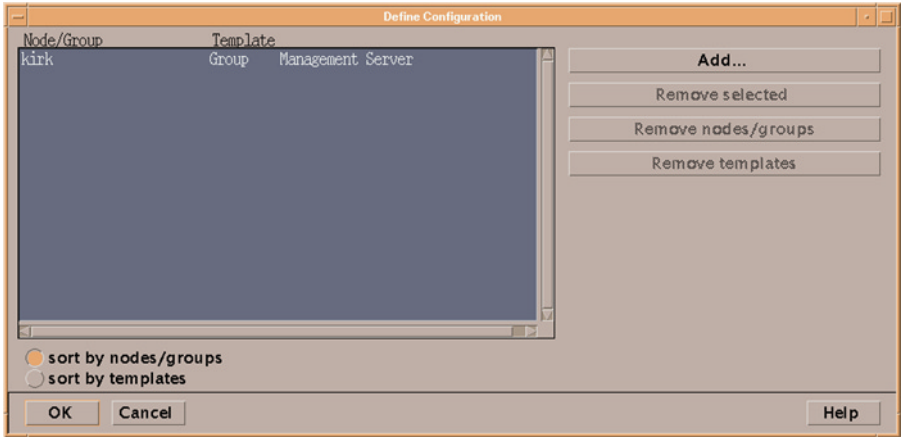
To assign VP390 templates to the agent component of the OVO management server, follow these steps:

1. Start the OVO GUI.
2. In the `Node Bank` window, click the OVO management server and select the following menu option:

   `Actions:Agents-> Assign Templates`

   The `Define Configuration` window opens (see Figure 1-1).

Figure 1-1 Assigning the VP390 Templates



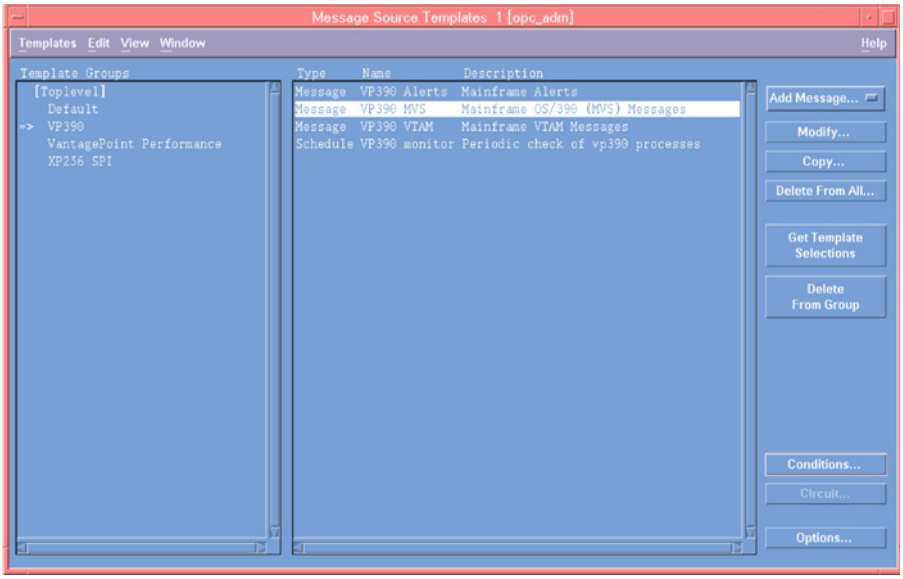
3. In the `Define Configuration` window, click `[Add...]`.

   The `Add Configuration` window opens.

4. In the `Add Configuration` window, click `[Open Template Window...]`.

   The `Message Source Template` window opens (Figure 1-2).

Figure 1-2 Message Source Template Window



5. In the left pane of the `Message Source Templates` window, click the `VP390` template group.

6. In the `Add Configuration` window, click `[Get Template Selections]`.

7. Make sure that the `VP390` template group appears in the `Templates` list, then click `[OK]`.

8. In the `Define Configuration` window, make sure that the VP390 templates appear, then click `[OK]`.

9. Close the `Message Source Templates` window.

To Distribute VP390 Templates

NOTE Before you distribute VP390 templates, the OVO agent software must be installed and active on the management server. To find out how to install the OVO agent, see the *HP OpenView Operations for UNIX Administrator's Reference, Volume I.*

To distribute VP390 templates to the agent component of the OVO management server, follow these steps:

1. In the `Node Bank` window, click the OVO management server and select the following menu option:

   `Actions:Agents-> Install/Update S/W & Config...`

   The `Install/Update S/W & Config...` window opens.

2. Verify that the management server node appears in the `Target Nodes` list.

3. In the left pane of the `Install/Update S/W & Config...` window, check the items you want to distribute along with the templates:

   - `Templates`
   - `Actions`
   - `Monitors`
   - `Commands`

   Then click `[OK]`.

4. Verify that the distribution was successful.

   When the assigned templates are distributed to the selected node, a message appears in the `Message Browser` window indicating whether the operation completes successfully.

   Click on `Window`, then `Message Browser`, to open the `Message Browser`.

NOTE You may be required to supply a password to access the node you selected to receive the distribution.

Phase 3: Starting the OV OS/390 Servers

Once you have added the mainframe nodes to the `Node Bank` window, and assigned and distributed the VP390 templates, you can start the OV OS/390 servers, verify the configuration, and verify that processes are running.

To Start the OV OS/390 Servers

To start the associated servers for the mainframe node you added in “Phase 1: Adding Mainframe Nodes to the Node Bank Window” on page 14, enter the following command:

`/opt/OV/vp390/bin/vp390sv -start <domain>`

To confirm the status of processes you started, enter the following command:

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

To Verify OV OS/390 Agent Processes are Running

To verify that mainframe processes are running, follow these steps:

1. Open the `VP390 Tools` group of the `Application Bank`, then drag the mainframe node from the Node Bank to the `Command Interface` application.
2. Log into the mainframe.
3. Issue a command that will generate a message defined in the mainframe filter file (for example, **`V NET,INACT,ID=resource`**).

   If OV OS/390 is configured correctly, the message will be displayed in the `Message Browser`.

Phase 4: Assigning OV OS/390 Applications to OVO Users

OVO users can see only those applications that are assigned to their user group. The installation of the VP390 software bundle automatically assigns the entire `VP390 Tools` Application Group to the `vp390_adm` user profile. The VTAM Display applications are assigned only to the `vp390_op` user group. You should assign your OVO users to the group that matches their authority on the mainframe computer.

To assign OV OS/390 applications to OVO users, follow these steps:

1. Open the `User Bank` and `User Profile Bank` windows.
2. Right-click `[OVO` *`operator id`*`]` and select `Modify`.
3. In the `Modify User` window, click the `[Profiles]` button.

   The `Profiles of Users` window opens.
4. Drag a VP390 profile from the `User Profile Bank` window and drop it into the `Profiles of User` window.
5. Close the `Profiles of User` window.
6. Click `[OK]` in the `Modify User` window.

About Path Parameters

NOTE

You cannot change the *EVOPATH* and *EVOWORK_AREA* parameters when running `vp390addagt`. You can change their default values later by editing the configuration file created in `/etc/opt/OV/share/conf/vp390`.

EVOPATH

Must be set to the value of the OV OS/390 core directory.

Default Value

`/opt/OV/vp390`

Valid Values

OV OS/390 home directory

EVOWORK_AREA

Specifies where OV OS/390 places temporary work files.

Default Value

`/var/opt/OV/share/tmp/vp390`

Valid Values

Any existing directory

About Agent Address Parameters

Agent address parameters are used by OV OS/390 client processes to locate and establish communications with OV OS/390 servers. You must set each agent address configuration parameter to the IP hostname of the machine where the specified server is running.

NOTE In most situations, all agent address parameters, except EVOMF_AGENT_ADDR, are the machine name of the UNIX server where OV OS/390 is installed.

EVOALERT_AGENT_ADDR

Machine name of the UNIX server where the `Alert Server` is installed.

Default Value

`hostname`

Valid Values

Hostname of `Alert Server`

EVOHCI_AGENT_ADDR

Machine name of the UNIX server where the OV OS/390 `Master Message Server` is installed.

Default Value

`hostname`

Valid Values

Hostname of `Master Message Server`

EVOCMDS_AGENT_ADDR

Machine name of the UNIX server where the `Command Server` is installed.

Default Value

`hostname`

Valid Values

Hostname of `Command Server`

EVOMF_AGENT_ADDR

Machine name of the mainframe computer where the OV OS/390 managed node component is installed.

Default Value

None

Valid Values

Hostname of OVO managed node

About Agent Port Parameters

You can change the agent port default values if any of these ports are already in use on your machine. Port numbers can be any number from `6000` to `9999`.

When managing more than one domain with OV OS/390, the installation program automatically increments port values for each specified domain by 10. For example, if a port number on `DOMAIN1` is `6100`, OV OS/390 automatically assigns the value for the same configuration parameter in `DOMAIN2` to `6110`.

EVOCMDS_AGENT_PORT

TCP port number assigned to the OV OS/390 `Command Server`.

Default Value

`6100`

Valid Values

Any unused port number

EVOHCI_ALERT_PORT

TCP port number assigned to the OV OS/390 `Message Server`.

Default Value

`6101`

Valid Values

Any unused port number

EVOALERT_AGENT_PORT

TCP port number assigned to the OV OS/390 `Alert Server`.

Default Value

`6102`

Valid Values

Any unused port number

EVOHCI_STATUS_PORT

TCP port assigned to the OV OS/390 `Status Manager`.

Default Value

`6103`

Valid Values

Any unused port number

EVOMF_HCI_AGENT_PORT

TCP port assigned to the mainframe `Message` subtask.This number must match the first number on the TCP SYSIN parameter card on the mainframe system.

Default Value

`6106`

Valid Values

Any available mainframe TCP port

EVOMF_CMDS_AGENT_PORT

TCP port assigned to the mainframe `Command` subtask. This number must match the second number on the TCP SYSIN parameter card on the mainframe system.

Default Value

`6107`

Valid Values

Any available mainframe TCP port

About Configuration and Status Parameters

Configuration and status parameters determine the type of SNA discovery used by OV OS/390, whether OV OS/390 adds new resources to the management database as they are encountered, whether OV OS/390 discovers logical units (LUs) and adds them to the object database, whether OV OS/390 discovers and monitors switched SNA physical units (PUs), and whether OV OS/390 keeps files created during the discovery process.

DISCOVER_MODE

Determines the type of SNA discovery used by OV OS/390.

Default Value

`ACTIVE`

Valid Values

`ACTIVE`	OV OS/390 queries the mainframe with VTAM commands to discover the SNA environment. The `Status Manager` is started after completion of this process to continually update the status of maps.
`PASSIVE`	OV OS/390 allows the `Status Manager` to passively build the SNA environment. The `Status Manager` receives VTAM status changes and adds the SNA resources as they are encountered. Set the *`UPDATE_DB_W_NEWNODES`* parameter to **yes** for the passive discovery to function properly.

UPDATE_DB_W_NEWNODES

Determines if OV OS/390 adds new resources to the management database as they are encountered. To effectively monitor switched physical units (PUs), set this parameter to **yes**. When you set the parameter to **yes**, the `Status Manager` will keep track of new SNA devices.

Default Value

yes

Valid Values

yes	As new SNA resources are encountered by OV OS/390, they are dynamically added to the management database.
no	Any new SNA resources encountered following the initial SNA map build are *not* added to the management database. Only current nodes are monitored.

INCLUDE_LUS

Determines if OV OS/390 should discover logical units (LUs) and add them to the object database.

Default Value

no

Valid Values

yes or no

MONITOR_SWITCHED_PUS

Determines if OV OS/390 should discover and monitor switched SNA physical units (PUs).

Default Value

no

Valid Values

yes or no

KEEP_DISCOVER_INPUT_FILES

Indicates whether the OV OS/390 discovery process will keep files created during the discovery process.

Default Value

yes

Valid Values

yes or no

About Mainframe Command Parameters

Mainframe command parameters indicate whether the output of `VTAM Display` commands should be restricted, define the autotask ID under which NetView/390 commands may be issued, and define the amount of time to wait for a mainframe command response.

EVOCMD_MAX_OVERRIDE

Indicates whether the output of `VTAM Display` commands should be restricted by using the `MAX=*` operand. On a large SNA network, setting this value to `yes` will result in an abbreviated discovery. Each VTAM Display used in the discovery will have its output lines limited to the `DSPLYDEF` value defined in VTAM.

Default Value

`no`

Valid Values

`yes` or `no`

EVOCMD_OPERATOR

Defines the name of the NetView/390 autotask ID under NetView/390 which commands may be issued. If NetView/390 is in use on the mainframe, this name must match the name of the autotask defined in the NetView/390 `DSIPARM(DSIOPF)` member.

Default Value

`EVOAUTO1`

Valid Values

Any OV OS/390 defined NetView/390 operator task

EVOCMD_TIMEOUT

Defines the amount of time to wait for a mainframe command response (in seconds).

Default Value

`30`

Valid Values

Any integer greater than or equal to `1` (second)

About Miscellaneous Parameters

Miscellaneous parameters determine the size of OV OS/390 generated log files and provide optional mainframe domain names.

EVOLOGSIZE

Maximum size of any OV OS/390 generated log file (in kilobytes).

Default Value

`3000`

Valid Values

Any integer greater than or equal to `1` (kilobyte)

EVOHOST_NAME

Optional name for the mainframe domain, comparable to the *`DOMAINID`* parameter in the `NetView/390 DSIPARM(DSIDMN)` member.

Default Value

`NET1`

Valid Values

SNA hostname

EVO_ADDFIELDS_TO_MSG

Determines if OV OS/390 messages should be prefixed with additional S/390 information. If this parameter is set to `yes`, each MVS message sent to OVO will have the originating S/390 system name, job name, and job ID (if any) attached to the front of the message. Each of these fields may be up to 8 characters long, and the fields will be separated by a space. If a field is blank, it will be represented by the the string “N/A”. SNA network messages which are received by OV OS/390 through one of the VTAM interfaces (the PPO or PPI) will have only the jobname “VTAM”.

NOTE Using this option will require the OS/390 message templates to be recoded to expect these additional fields.

Sample of incoming messages with parameter set to no:

```
*$HASP050 JES2 RESOURCE SHORTAGE OF TGS - 80%
UTILIZATION REACHED

IST105I NCP01 NODE NOW INACTIVE
```

Sample of incoming messages with parameter set to yes:

```
Z390 JES2 N/A *$HASP050 JES2 RESOURCE SHORTAGE OF TGS -
80% UTILIZATION REACHED

N/A VTAM N/A IST105I NCP01 NODE NOW INACTIVE
```

Default Value

no

Valid Values

yes or no

EVO_ETOAFILE

Identifies the file to be used to translate S/390 EBCDIC characters to ASCII characters. If a new character mapping file is to be used, make a copy of the default file `vp390.etoa.engus` and change the hexadecimal values for the desired characters, then enter the name of the new file in this parameter.

Default Value

`/opt/OV/vp390/local/C/vp390.etoa.engus`

Valid Values

Filename on the OVO server

EVO_DSILOG_DIR

Identifies the directory path where measurement logf files for performance data collection are stored. If this value is changed, the OpenView Performance agent

configuration for OS/390 performance data collection must use the same directory path.

Default Value

`/var/opt/OV/vp390/datafiles`

Valid Values

Any valid Unix directory path

Phase 5: Configuring Mainframe Nodes for RMF Performance Data Collection

NOTE Before performing this configuration you must have the "HP OpenView Smart Plug-ins DSI-to-DDF wrapper utilities" (DSI2DDF) component installed. This component is available on the HP OpenView Smart Plug-in CD.

Collecting performance metrics for mainframe nodes is an optional task and requires configuration steps to be performed on the mainframe agent and the OVO Management server. Collection of performance data requires either the CODA performance sub-agent (OVO 7.0 or higher) or the OpenView Performance agent to be running. The mainframe node must be running the IBM RMF product and have an RMF Monitor I task started.

The `vp390addperf.pl` script creates a performance class specification file for a mainframe node and updates either the CODA performance sub-agent or the OpenView Performance agent (OVPA) for the collection of mainframe performance metrics. If OVPA is available it will be used by default. If you wish to use CODA performance sub-agent instead of the OpenView Performance Agent, you must create an empty file named hocoda.opt. It must be located in `/var/opt/OV/conf/dsi2ddf/nocoda.opt`.

To configure a mainframe node for performance data collection, follow these steps:

1. - From the Node Bank, select the mainframe node to be configured. From the Applications Bank, select `VP390 Tools`, and then select and execute `Configure Perf Collection`

   Or from the UNIX command line, enter the following command:

   `/opt/OV/vp390/bin/vp390addperf.pl <OS390Host-FQDN>`

   The `vp390addperf.pl` will create a class specification file for the mainframe node, and update either the CODA performance sub-agent or the OpenView Performance agent.

2. If the OpenView Performance agent is installed, the OVPA processes must be restarted to activate the new configuration using the command:

   `/opt/perf/bin/mwa restart`

3. The OV OS/390 server processes must be restarted to activate the interface to

either the CODA sub-agent or the OpenView Performance agent. To restart the OV OS/390 server processes, enter the commands:

```
/opt/OV/vp390/bin/vp390sv -stop <OS390Host-FQDN>
/opt/OV/vp390/bin/vp390sv -start <OS390Host-FQDN>
```

4. On the mainframe agent, the PERF card must be added to the VP390 `SYSIN` parameter cards. After the PERF card has been added, you must restart the OV OS/390 agent started task to activate the new configuration. The interval at which data is collected is specified on the PERF card. The default value is 15 minutes. (See the HP OpenView OS/390 Management Installation Guide for details on the OV OS/390 agent parameter cards).

See Appendix C for the class specification file for RMF performance data.

Phase 6: Configuring Mainframe Nodes for DASD Statistics Collection

NOTE Before performing this configuration you must have the "HP OpenView Smart Plug-ins DSI-DDF wrapper utilities" (DSI2DDF) component installed. This component is available on the HP OpenView Smart Plug-in CD.

Collecting DASD statistics for mainframe nodes is an optional task and requires configuration steps to be performed on the mainframe agent and the OVO Management server. Collection of performance data requires either the CODA performance sub-agent (OVO 7.0 or higher) or the OpenView Performance agent to be running. There are two scripts available for configuring DASD statistics collections. The `vp390add_dasd_stat.pl` script creates a class specification file for the collection of statistics from all or a subset of DASD volumes. The `vp390_dasd_summ.pl` script creates a class specification file for the collection of summarized statistics from all or a subset of DASD volumes.

To configure DASD Statistics collection, perform the following steps:

1. From the `Node Bank`, select the mainframe to be configured. From the `Application Bank`, select `VP390 Tools`, and then select and execute either `Config DASD Stat Collection` or *`Config DASD Summ Collection`*.

   Or from a UNIX command line, enter one of the following commands:

   `/opt/OV/vp390/bin/vp390add_dasd_stat.pl <OS390Host-FQDN>`

   or

   `/opt/OV/vp390/bin/vp390add_dasd_summ.pl <OS390Host-FQDN>`

   The script will create a class specification file for the mainframe node, and update either the CODA performance sub-agent or the OpenView Performance agent.

2. If the OpenView Performance agent is installed, the OVPA processes must be restarted to activate the new configuration using the command:

   `/opt/perf/bin/mwa restart`

3. From the Message Source Templates window, select the VP390 (A.05.00) group, and select the `VP390DASD_COLLECT` schedule template. In this template specify the schedule for DASD Statistics collection to occur. The Scheduled Action template executes the script `vp390dasd_collect.pl` to perform the DASD data collection. The default command line action in this template is for the collection of data on all online DASD volumes. If you want to limit the collection to a subset and volumes, add the following parameter to the command line:

   `-e <expression>`

   where *expression* is a Unix style regular expression to be used for selecting which DASD volumes should be collected. For example:

   `-e OS39*`

   collects statistics from all volumes starting with “OS39”

   To collect only summarized statistics, remove the “`-detail`” parameter from the command line.

4. Save the Scheduled Action template and assign and distribute this template to the OVO management server.

5. On the mainframe agent, the `OSINFO` card must be added to the VP390 `SYSIN` parameter cards. If necessary, add the `OSINFO` parameter card to the VP390 agent and restart the VP390 started task. (See the *HP OpenView OS/390 Management Installation Guide* for details on the agent’s `SYSIN` parameter cards).

See Appendix C for the class specification file for DASD statistics.

Phase 7: Configuring Services for the Mainframe Node

Configuring services for a Mainframe node is an optional task that requires the modification of the sample OV OS/390 XML file and execution of service navigator commands to add the configured services.

Planning and configuring a services map for use with the Service Navigator requires thought and planning. The sample XML service configuration mapping provided with OV OS/390 can be used as a starting point to configure low-level component services for OS/390 or z/OS components and mapping mainframe messages to those component services.

The example services configuration file is located in the OV OS/390 configuration directory, `/etc/opt/OV/share/conf/vp390` and is named `vp390.services`. To use this file, first make a copy of the file. By default, the parent (top level service) in this file is the mainframe node. The mainframe components (for example, JES2, VTAM, etc.) are defined as subordinate services of the mainframe node. This may need to be modified, depending on your service mapping strategy. The low-level component services have service names that begin with the mainframe node name, followed by a colon (:) and the component name, for example JES2. In the example services file the mainframe node name will need to be modified for the actual node name being configured.

After modifications have been made to the example services file, use the `opcservice` command to add the services and assign operators to the new services.

2 Using OV OS/390

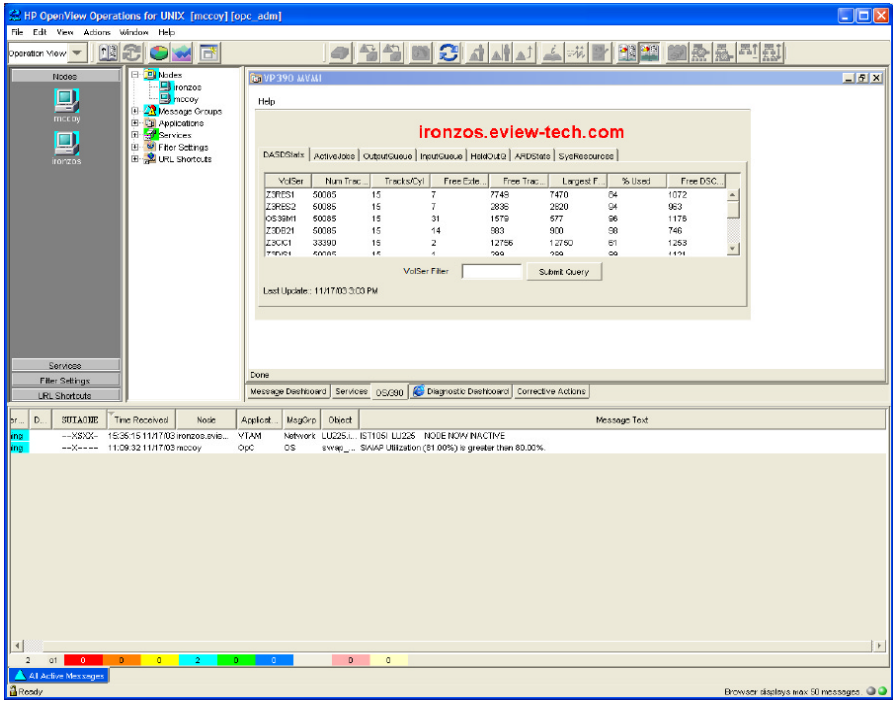
This chapter describes how to use HP OpenView OS/390 Management (OV OS/390) to perform daily tasks.

About the OV OS/390 Mainframe Visual Management Interface

The OV OS/390 Mainframe Visual Management Interface (MVMI) is a tool that provides the operator with features to monitor and manage system resources such as online DASD volumes, active, and queued jobs, job output queues and system performance data.

The figure below shows the OV OS/390 MVMI launched within the OVO Java Console workspace.

Figure 2-1 **OV OS/390 MVMI**



Using the OV OS/390 Management Visual Interface

1. Start the OVO Java Console.
2. Create a new workspace or select an existing workspace, making sure the new or selected work space is NOT an ActiveX container.
3. In the Objects pane, expand the `Nodes` folder in the Object tree.
4. Right click on the mainframe node to be monitored, then:
   - Select `Start`
   - Select `VP390 Tools (A.05)`
   - Select `VP390 MVMI`
5. If you are using the embedded browser, the applet will start in the current workspace. If using an external browser a new browser window will open.

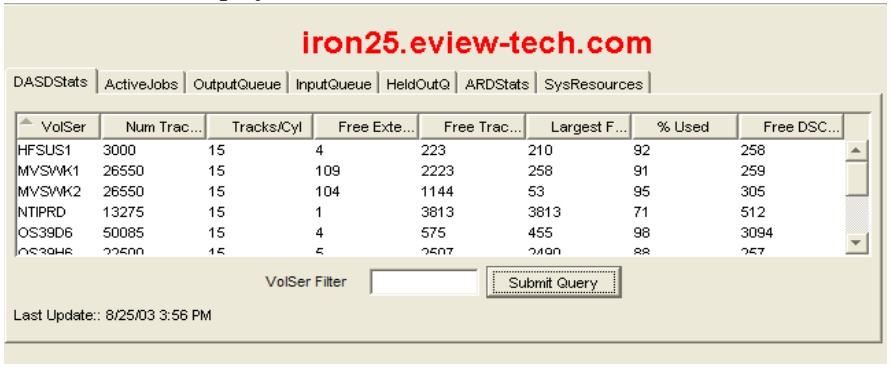
NOTE If you are using an external browser, the Sun Java plugin Version 1.3 or later must be installed and integrated with the browser.

To Monitor DASD Statistics

1. Select the `DASDStats` tab.
2. To limit the DASD volumes displayed, you may enter a Unix-style regular expression in the `VolSer Filter` input box. To retrieve information for all online DASD volumes leave the`VolSer Filter` field blank.
3. Use the`[Submit Query]` button to retrieve the DASD information.
4. The output can be sorted by clicking on the column headers. The initial sort direction is ascending. Subsequent clicks will reverse the sort direction.

NOTE Changes to the data will not be updated until the `[Submit Query]` button is clicked.

Figure 2-2 DASD Statistics Display

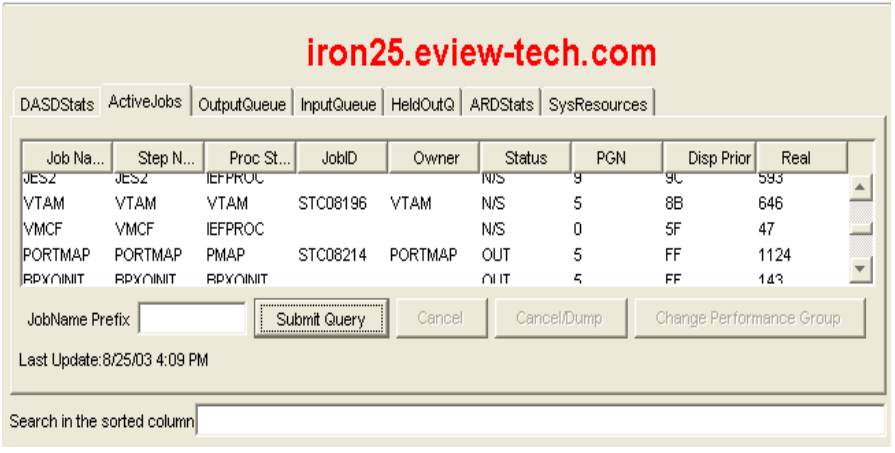


To Monitor Active Jobs

1. Select the `ActiveJobs` tab.
2. To limit the active jobs displayed, you may enter a job name prefix in the `JobName Prefix` input box. To retrieve information for all active jobs, leave this field blank.
3. Use the `[Submit Query]` button to retrieve the active job information.
4. The output can be sorted by clicking on the column headers. The initial sort direction is ascending. Subsequent clicks will reverse the sort direction.
5. By selecting a job, and clicking the appropriate button you may perform the following actions on active jobs:
   - `[Cancel]` - cancel the selected job
   - `[Cancel/Dump]` - cancel the selected job with a dump
   - `[Change Performance Group]` - reset the job to run in a different performance group. (This button is not relevant if the mainframe is running WorkLoad Manager in goal mode.)

NOTE Changes to the data will not be updated until the `[Submit Query]` button is clicked.

Figure 2-3 **Active Jobs Display**

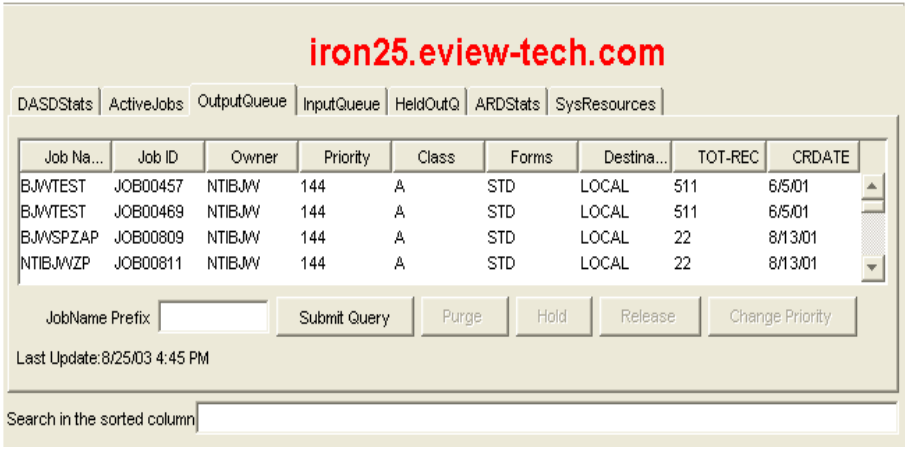


To Monitor the JES2 Output Queue

1. Select the `OutputQueue` tab.
2. To limit the output jobs displayed, you may enter a job name prefix in the `JobName Prefix` input box. To retrieve information for all output jobs, leave this field blank.
3. Use the `[Submit Query]` button to retrieve the output jobs information.
4. The output can be sorted by clicking on the column headers. The initial sort direction is ascending. Subsequent clicks will reverse the sort direction.
5. By selecting a job and clicking the appropriate button, you may perform the following actions on output jobs:
   - `[Purge]` - purge the selected job
   - `[Hold]` - hold the selected job
   - `[Release]` - release the job
   - `[Change Priority]` - change the jobs priority in the output queue

NOTE Changes to the data will not be updated until the `[Submit Query]` button is pressed.

Figure 2-4 **JES2 Output Queue Ddisplay**



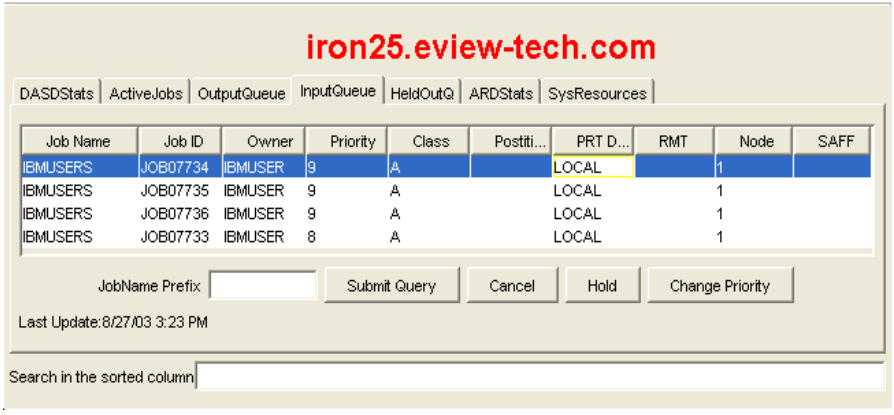
To Monitor the JES2 Input Queue

1. Select the `InputQueue` tab.
2. To limit the jobs displayed, you may enter a job name prefix in the `JobName Prefix` input box. To retrieve information for all jobs on the input queue, leave this field blank.
3. Use the `[Submit Query]` button to retrieve the active job information.
4. The output can be sorted by clicking on the column headers. The initial sort direction is ascending. Subsequent clicks will reverse the sort direction.
5. By selecting a job and clicking the appropriate button, you may perform the following actions on jobs:
   - `[Cancel]` - cancel the selected job
   - `[Hold]` - hold the selected job
   - `[Release]` - release the job
   - `[Change Priority]` - change the jobs priority in the output queue

NOTE

Changes to the data will not be updated until the `[Submit Query]` button is clicked.

Figure 2-5 **JES2 Input Queue Display**

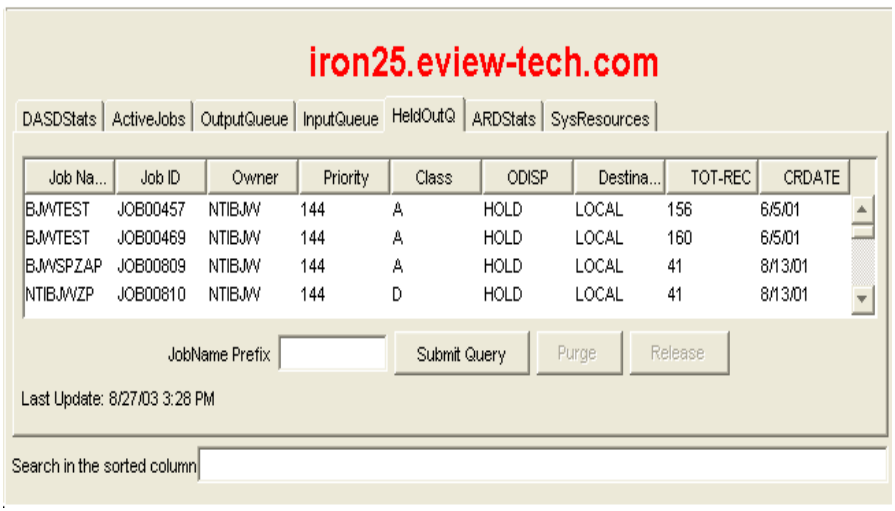


To Monitor the JES2 Held Queue

1. Select the `HeldOutQ` tab.
2. To limit the active jobs displayed you may enter a job name prefix in the `[HeldOutQ]` input box. To retrieve information for all held jobs, leave this field blank.
3. Use the `[Submit Query]` button to retrieve the held jobs information.
4. The output can be sorted by clicking on the column header. The initial sort direction is ascending. Subsequent clicks will reverse the sort direction.
5. By selecting a job, and clicking the appropriate button you may perform the following actions on held jobs:
   - `[Purge]` - purge the selected job
   - `[Release]` - release the job

NOTE Changes to the data will not be updated until the `[Submit Query]` button is clicked.

Figure 2-6 **JES2 Held Queue Display**

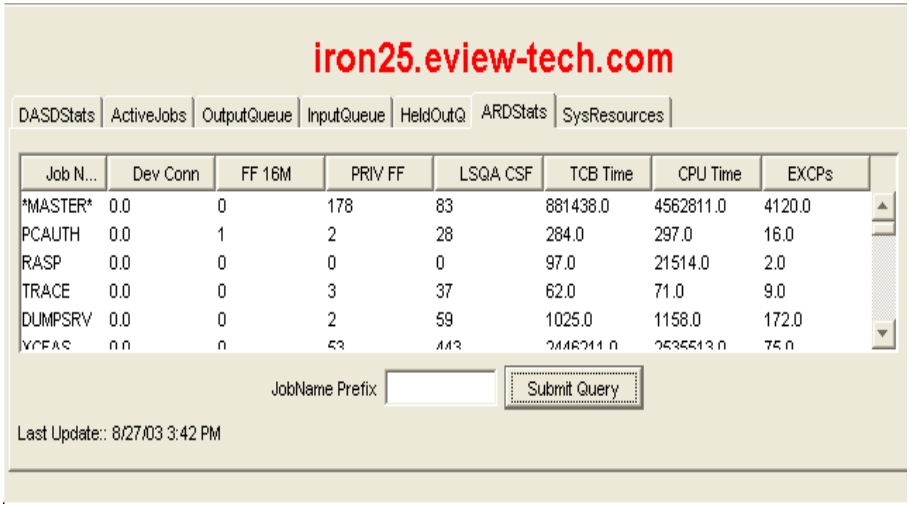


To Monitor Address Space Resource Statistics

1. Select the `ARDStats` tab.
2. To limit the active jobs displayed, you may enter a job name prefix in the `JobName Prefix` input box. To retrieve information for all active jobs, leave this field blank.
3. Use the `[Submit Query]` button to retrieve the active job information.
4. The output can be sorted by clicking on the column header. The initial sort direction is ascending. Subsequent clicks will reverse the sort direction.

NOTE Changes to the data will not be updated until the `[Submit Query]` button is clicked.

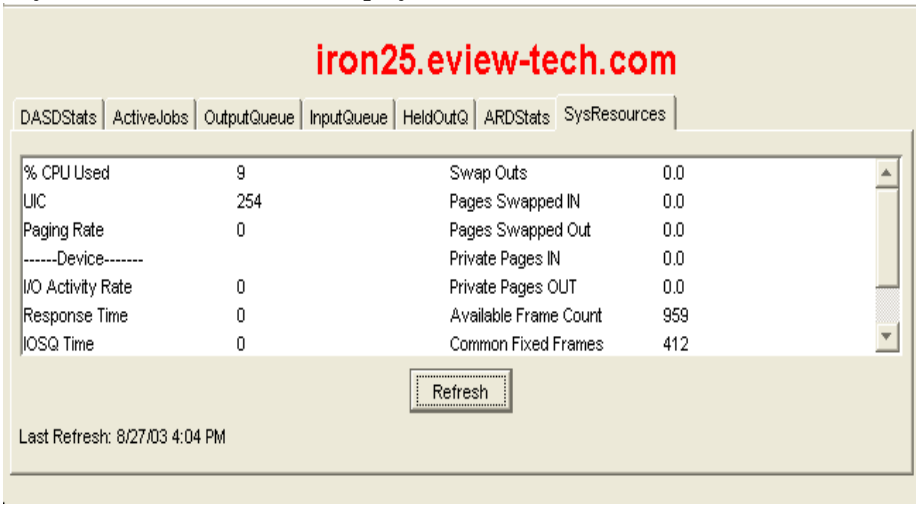
Figure 2-7 **Address Space Statistics Display**



To Monitor System Resource Statistics

1. Select the `SysResources` tab.
2. Statistics for Swap Outs, Pages Swapped In, Pages Swapped Out, Private Pages In and Private Pages Out will not be displayed on the initial view. These numbers are calculated each time the `[Refresh]` button is clicked using the time between refreshes to calculate the current rate.

Figure 2-8 **System Resource Statistics Display**

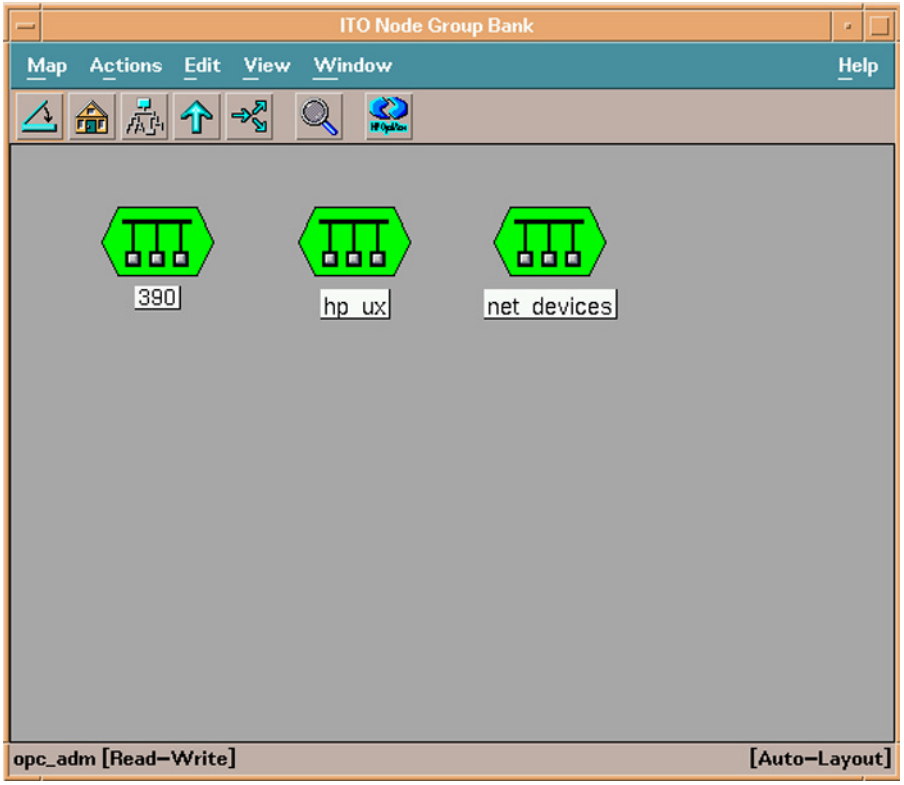


About the OV OS/390 Node Group

The installation of OV OS/390 creates a node group, `390`, which includes all S/390 systems in the OV OS/390 environment.

Figure 4-1 shows the `390` node group in the `Node Group Bank`.

Figure 2-9 **390 Node Group**



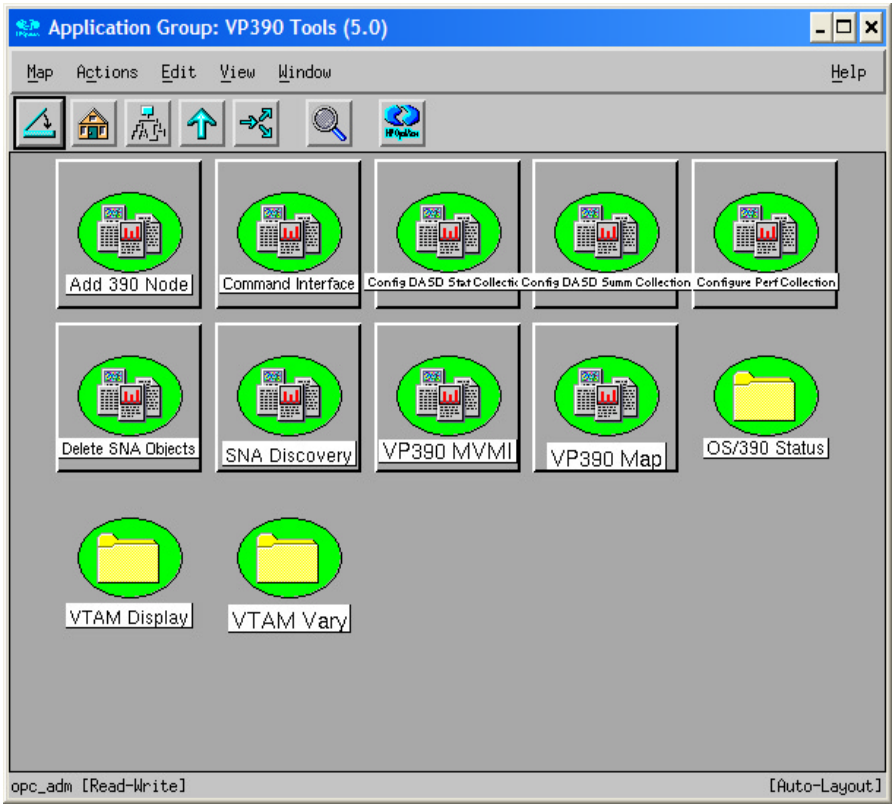
By assigning the `390` node group and the `OS` and `Network` message groups to the OVO operator responsible for the OV OS/390 environment, you ensure that messages relating to errors and potential problems with S/390 system appear in the appropriate operator's `Message Browser`.

About the OV OS/390 Application Bank

The installation of OV OS/390 creates an Application Bank, called `VP390 Tools`, which contains a number of applications designed to help OVO operators manage and monitor the S/390 mainframe environment. From the main `Application Bank`, double-click `[VP390 Tools]` to display the `VP390 Application Group`.

Figure 2-10 shows the `VP390 Application Group`.

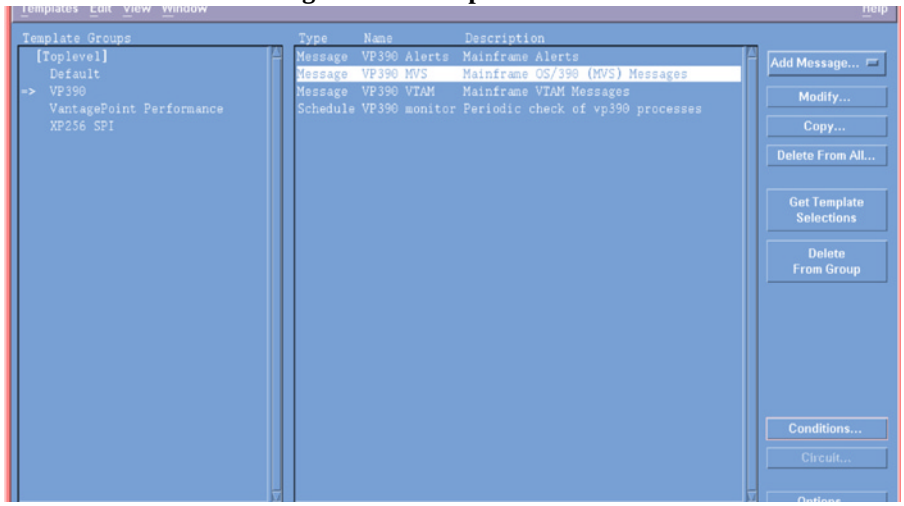
Figure 2-10 **OV OS/390 Application Group**



About OV OS/390 Message Source Templates

One template group, VP390, contains all the default message source templates provided with OV OS/390, as shown in Figure 2-11.

Figure 2-11 **Default OV OS/390 Message Source Templates**



To Add or Modify Actions

To add or modify automatic or operator-initiated actions, follow these steps:

1. From any OVO GUI, select the Window menu item.
2. Click [Message Source Templates].
3. In the left pane of the Message Source Template window, click VP390.
4. Click any [Message Group].
5. Click [conditions].
6. Click any [Message ID].
7. Click [Modify].

   The Condition Template window opens. From here, you can add or modify automatic and operator-initiated actions.

Intercepting Messages and Alerts

The OV OS/390 message source templates intercept messages and alerts that are generated on the mainframe managed node. OV OS/390 is shipped with three default message templates:

- OS/390 (MVS) messages
- VTAM messages
- VTAM alerts

Use the default messages templates listed as a guide for creating automatic and operator-initiated actions for other messages.

NOTE

Message IDs must match the S/390 agent's message filter table to be sent to the OVO Management Server.

Filtering Messages and Alerts

Message and alert filtering is performed on the S/390 agent to reduce the amount of network traffic into the OVO server. For details about message and alert filtering, see the description of the FILTER mainframe console command in Appendix A.

Filtering Messages and Alerts

Message received from the S/390 are passed through the OVO Message stream interface. Although most mainframe system and application messages consist of a message ID and message text, the free form capacity of mainframe messages does not require this format.

In addition to the original message text, the following OVO optional variables are passed along with the message:

Table 2-1

$OPTION(jobname)	Job name (if any) which issued the message (up to 8 characters)
$OPTION(jobid)	Job ID (if any) which issued the message (up to 8 characters)

Table 2-1

$OPTION(datestamp)	Date when the message was issued in MM.DD.YY format
$OPTION(timestamp)	Time when the message was issued in HH.MM.SS format
$OPTION(systemname)	System name where the message was generated (up to 8 characters)
$OPTION(msgkey)	Unique integer assigned to this message by OS/390
$OPTION(msgtoken)	Integer assigned to this message by OS/390 to mark it as part of a group of related messages

These variables can be used in displayed message text as well as passed as arguments to automatic or operator initiated actions. Automatic and operator initiated actions may consist of native OS/390 commands as well as scripts or programs. OS/390 commands may be incorporated into scripts by using the `vp390hostcmd` utility. See the man page for details and syntax of the `vp390hostcmd` command.

OV OS/390 optionally allows some message attributes to be inserted directly into the original message text as it passes into OVO. Use the EVO_ADDFIELDS_TO_MSG option when configuring the agent connection to deliver the S/390 system name, job name, and job ID in the original message text. See the description of EVO_ADDFIELDS_TO_MSG on page 31.

Multi-line MVS messages will be reformatted before being presented in the OVO browser. If a MVS message ID is added to the mainframe message filter table with a preceding plus sign “+”, then each line of the multi-line output will be sent with the message ID prepended to the front of the secondary lines. If an MVS message ID is added to the mainframe message filter table with a preceding minus sign “-” then each line of the multi-line output will be sent without the original message ID prepended to the front of the secondary lines. If the message ID is listed in the filter table with neither the “+” nor the “-” sign, then all of the lines of the message will be concatenated together and displayed in the OVO browser as one continuous line. Each line of the multi-line message will be separated by a new-line character (\n). Select `[Show Original Message...]` from the `Message Details` window of the browser to see the message displayed as multiple lines.

Using Monitors

OV OS/390 provides three system monitors in the shipped product which can be modified for your needs. These monitor templates can also be used as samples for creating additional monitors needed in your environment. To implement one or more of these monitors, update the CPU monitor or DASD monitor template conditions if desired, or add job information to the job monitor configuration file. Then assign the monitor template(s) to the OVO Management server agent. Finally, distribute the monitor templates to the OVO Management server agent. (Do not distribute templates to the S/390 managed node.) The default monitors provided are:

OS390_CPUMON : CPU Monitor

Monitors the system CPU of all OS/390 nodes in the OVO Node Bank. The default template generates warning messages of increasing severity when CPU utilization exceeds 75, 85, 90, and 95 percent.

OS390_JOBMON : Job Monitor

The job monitor template may be used to monitor for critical jobs that should be active on a mainframe node. The job monitor script (`vp390jobmon.pl`) uses a configuration file to determine which jobs to monitor and when to monitor for the jobs. If a specified job is not running, a message will be sent to the OVO browser.

The default configuration file for the job monitor script is `/etc/opt/OV/share/conf/vp390/vp390jobmon.conf`. This file will need to be customized for each installation. The file contains default names for standard WebSphere process names as comments.

Each record (line) in the file must begin in column 1 and have the following syntax:

`<jobname> <host> [<starttime> <duration> [<daysofweek>]]`

where:

`<jobname>`	the name of the job/address space that the monitor is to check
`<host>`	the fully qualified name of the mainframe where the job should be active

The remaining parameters are optional and may be used to limit the time frame that the job monitor script will check for the specified job to be active.

`<starttime>`	the starting time for the interval that the monitor script will check for the job to be active. The time must be

	specified in the format HH:MM:SS
<duration>	the duration of time that monitor script will check for the job to be active. The duration must be a number followed by an “h”, “m”, or “s” to indicate hours, minutes, or seconds. To specify a duration of eight hours, enter “8h”.
<daysofweek>	a limit of the number of days in the week to check for the active job. The days are designated by number, with 0=Sunday and 6=Saturday. Multiple days must be separated by a comma. To specify Monday through Friday inclusive, enter “1,2,3,4,5”.

Examples:

To monitor job JOBA on mainframe bluebox1.mycom.com during the hours 0800-1600 on Mondays, Wednesdays, and Fridays, enter the line in the `vp390jobmon.conf` file:

```
JOBA bluebox1.mycom.com 08:00:00 8h 1,3,5
```

To monitor job JOBB on mainframe bluebox2.mycom.com every day during the hours of 2200-1000 (crossing midnight to the next day), enter the line:

```
JOBB bluebox2.mycom.com 22:00:00 12h
```

OS390_DASDMON : DASD Monitor

Monitors the DASD utilization of all online volumes. By default, warning messages of increasing severity are generated when DASD utilization exceeds 90 and 95 percent (%).

Using DOM Information

Mainframe OS/390 messages which require an action or are otherwise highlighted for severity will have their intensity removed when the appropriate action is taken (such as mounting a tape on a drive) or the severe condition is relieved (such as a disk space shortage). OS/390 signals the removal of the message highlighting by issuing a Delete Operator Message (DOM). OV OS/390 can optionally pick up these DOMs and forward them to the OVO server, where they can be used to automatically acknowledge messages that may be on the browser.

To get DOMs delivered to the OVO server, either add the `DOM` parameter to the MVS startup initialization card in the SYSIN for the mainframe agent’s job (see the description of the MVS card in the *HP OpenView OS/390 Management Installation*

Guide, or activate it using the DOM console command (see “About DOM Command”).

DOM information will arrive in the OVO browser in the form of an EVO211 message with the following format:

`EVO211 source key`

where:

source	Either “MSGKEY” or “MSGTOKEN” depending on whether the DOM is for a specific message or a group of messages grouped together with a token.
key	An integer value which will match either the $OPTION(msgkey) or $OPTION(msgtoken) optional variable that was delivered with the original message.

The EVO211 message can therefore be set up in the OVO message templates with correlation to auto-acknowledge a previous message with the optional variable that matches the `key`. The `VP390 MVS` messages template has conditions for action messages (see condition for $HASP050) that create a message key for acknowledgement by the EVO211 message. See the conditions for the EVO211 message for the creation of the acknowledgement key.

Discovering SNA Resources

OV OS/390 populates the OpenView database with SNA resource information through the discovery process. In OpenView Operations (OVO) and Network Node Manager (NNM), each System Network Architecture (SNA) resource is represented by an icon. The background color of the icon indicates the status of the SNA resource being managed by OV OS/390. Each possible status value has an associated color.

About Status Names and Colors

Table 2-2shows the default OVO colors assigned to several common SNA resource statuses (as defined by VTAM).

Table 2-2 **VP390 Color Status Representation**

VTAM Status	OpenView Status	Color
ACTIV	Normal	Green
CONCT	Normal	Green
INACT	Disabled	Dark Brown
INOP	Major	Orange
IINOP	Critical	Red
PACTL	Marginal	Yellow
PAPU2	Marginal	Yellow
NEVAC	Unknown	Blue
All other statuses	Unknown	Blue

To Change Status Names and Colors

You can change VTAM and OpenView status names and colors before starting the `Discovery` process.

To change status names and colors, edit the configuration file:

`/etc/opt/OV/share/conf/vp390/sna_status.conf`

Customizing the Discovery Process

To customize the functionality of the discovery process, use the following configuration and status parameters in the configuration file for the mainframe domain. The configuration files are stored in `/etc/opt/OV/share/conf/vp390`.

UPDATE_DB_W_NEWNODES

`yes`	Default value. Updates the database with new nodes. For a new discovery, use this parameter.
`no`	No attempt is made to add new nodes. Simply updates the statuses of existing nodes. When maps are stable and no new nodes are desired, use this value.

DISCOVER_MODE

OV OS/390 discovery process queries the mainframe domain for VTAM resources and creates a mapped resource hierarchy.

`ACTIVE`	Default value. When discovery is run on a previously mapped domain, OV OS/390 changes the status of all nodes to `UNKNOWN` for that domain and then proceeds to re-discover the network, updating and adding new nodes (if permitted). If a node is removed, the status remains `UNKNOWN`.
`PASSIVE`	Causes OV OS/390 to change the status of all nodes to `UNKNOWN` for that domain and allows the `Status Manager` to reset the status of the nodes as they are reported.

INCLUDE_LUS

`no`	Default value. Does *not* include logical units (LUs) in the discovery of your network.
`yes`	Include LUs in the discovery of your network.

KEEP_DISCOVER_INPUT_FILES

yes	Default value. Keeps the discover files in the EVOWORK_AREA after the discover process is complete.
no	Does *not* keep the discover files in the EVOWORK_AREA after the discover process is complete.

To Start the Discovery Process

To start the discovery process, click a mainframe managed node in the Node Bank and drag it to the SNA Discovery icon in the VP390 Tools group of the Application Bank.

About the VTAM SNA Map Application

To provide a graphical (submap) display of discovered VTAM resources, assign the VP390SNA application from the VP390 Tools application group to OVO users.

Designating VP390SNA Map Administrators

The VP390SNA application uses the concept of a map administrator to control which users are permitted to make permanent deletions from the VP390SNA Map. The opc_adm user is designated as the SNA Map administrator during the installation of OV OS/390. You can designate additional OVO operators as VP390SNA Map administrators by adding the VP390SNA Map and Delete SNA Objects Applications to the operator's Application Bank.

You must also add the new operator identifications to the mapadmin.conf file in the following directory:

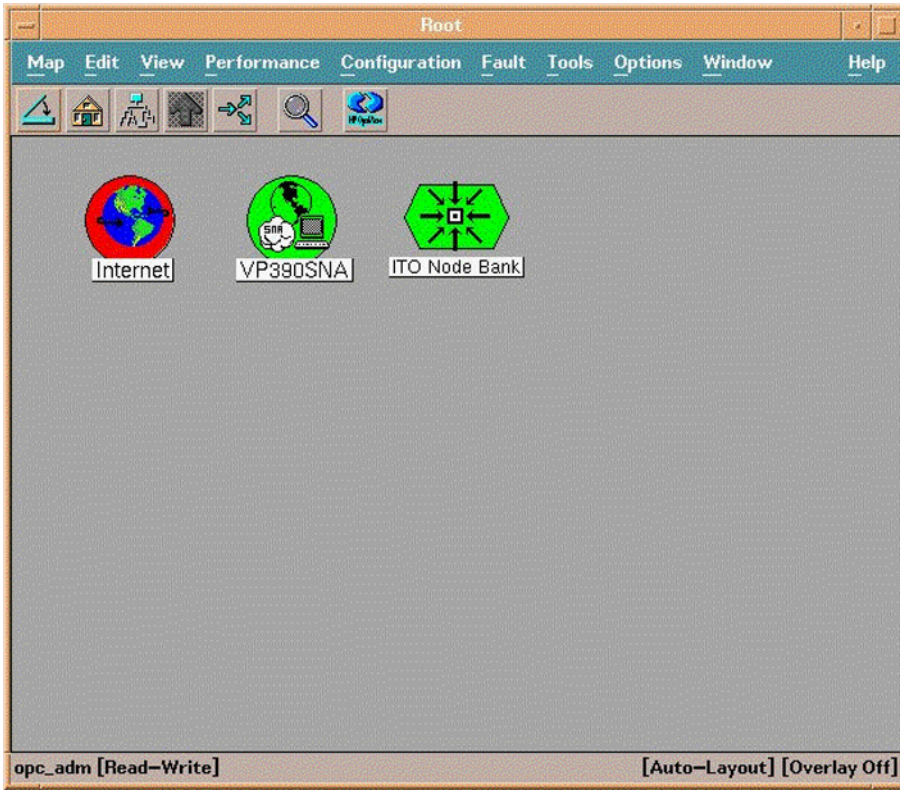
```
/etc/opt/OV/share/conf/vp390
```

You must place the identification of each additional operator on a separate line in the file, and must start in column 1.

Accessing SNA Submaps

The VP390SNA Map application builds hierarchical submaps for all discovered domains. You access the SNA submaps by starting at the VP390SNA icon on the root submap as shown in Figure 2-12.

Figure 2-12 **VP390SNA Icon on the Root Submap**



Deleting Objects from the VP390SNA Map

Only operators designated as VP390SNA Map administrators may make permanent deletions from the `VP390SNA Map`.

About Operator Rights

Each operator has a read-write map, and has the capability to delete icons from the read-write map. Operators who are not VP390SNA Map administrators cannot flag objects for deletion. When an operator who is not an VP390SNA Map administrator deletes icons, the VP390SNA Map application re-adds the icons for the deleted objects the next time that operator logs in.

Methods for Deleting SNA Objects

VP390SNA Map administrators can delete SNA objects from one of two locations:

- **Edit Toolbar**

  VP390SNA Map administrators can delete SNA objects from the `Edit` toolbar menu. The object and any child objects are removed from the operators's map and also marked for deletion. Selected objects are not deleted from any other operator's map until the operator closes the GUI session and restarts the GUI

- **Application Bank**

  VP390SNA Map administrators can delete SNA objects from the `Application Bank`. This method deletes the object and any child objects. It also sends an event to any open SNA Maps, which deletes the object immediately.

NOTE

With either method, objects are not be permanently deleted from the database until the icons have been deleted from all operator maps that have the `VP390SNA Map` application. Objects that are not deleted are renamed (the tag `REMOVED:` is added to the beginning of the selection name), and a delete flag is set for the object. As operator maps are opened, objects flagged for deletion are then removed from the operator's map. When the symbol is removed from the last map, the object is permanently deleted.

To Delete SNA Objects from the Edit Toolbar Menu

To delete SNA objects from the `Edit` toolbar menu, follow these steps:

1. Select the object icons to be deleted.

CAUTION

Selecting an object icon for deletion deletes the object, including any child objects on the child submaps.

2. From the `Edit` toolbar menu, select `Delete` to remove the object.

   The object and any child objects are removed from the operators's map and also marked for deletion. Selected objects are not deleted from any other operator's map until the operator closes the GUI session and restarts the GUI.

To Delete SNA Objects using "Delete SNA Objects" in the Application Bank

To delete SNA objects from the `Application Bank`, follow these steps:

1. Select the object icons to be deleted.
2. Open the VP390 Tools group of the Application Bank.
3. Execute the `Delete SNA Objects` application.

CAUTION

This method deletes the object and any child objects. It also sends an event to any open SNA Maps, which deletes the object immediately.

Using VTAM Command Applications

OV OS/390 enables you to execute VTAM commands and view their output in one of two ways:

- **VTAM Display Application**

  The `VTAM Display` applications provided in the `Application Bank` operate either on the entire domain or on a specific VTAM resource.

- **Command Interface**

  OVO operators can process user-specified commands with the `Command Interface` application if it has been assigned to them.

To Use the VTAM Display Application

To use the `VTAM Display` application, follow these steps:

1. Locate the domain or VTAM resource:

   - *Domain*

     If the VTAM command applies to the domain, locate the Mainframe (S/390) icon in the `Node Bank`.

   - *Resource*

     If the VTAM command applies to individual VTAM resources (for example, `PU` or `LU`) then locate the resource on the `VP390SNA Map`.

     To locate the resource, select `Find` from the `Edit` menu.

2. Select the `VP390 Tools` icon.

   The `Application Group: VP390 Tools` window opens.

3. Select the `VTAM Display` icon.

   The `Application Group: VTAM` Display window opens.

4. Process the command on the domain or resource:

   - *Domain*

     Click and drag the Mainframe (S/390) icon from the `Node Bank` window and drop it on the VTAM application that you want to process.

   - *Resource*

     Click and drag the VTAM Resource icon from the VP390SNA submap and drop it on the VTAM application that you want to process.

   The `Output of Application` window opens. This window contains the output from that VTAM command.

To Use the Command Interface Application

To use the `Command Interface` application, follow these steps:

1. Locate the Mainframe (S/390) icon in the `Node Bank`.
2. Double-click`VP390 Tools` icon in the `Application Bank`.

   The `Application Group: VP390 Tools` window opens.
3. Click and drag the Mainframe (S/390) icon from the `Node Bank` window and drop it on the `Command Interface` application.

   A window opens. You can enter commands and view command output from this window.

Using OSINFO System Information API Commands

The OSINFO subtask of the VP 390 agent task will gather various S/390 operating system statistics and present the data in a format that can be parsed by a script on the OVO server. OSINFO data are requested using destination 46 of the VP390hostcmd utility. (See the `man` page for general syntax of `vp390hostcmd`.) The OSINFO data are requested by specifying a two-digit code followed by a vertical bar and additional parameter information depending on the selected code. For example, to gather DASD information (code 01) for a volume named:"Disk99" on mainframe "s390.mysite.com," the vp390hostcmd syntax is:

```
vp390/bin/vp390hostcmd 46 01|DISK99.s390.mysite.com
```

Keep in mind that if this command is entered on a Unix command line or script, the vertical bar will need to be escaped with a backslash (\)character.

Output lines will be returned with values separated by a vertical bar. One line will be generated for each record found, representing one job, device, etc. The last line will be the text "EOF".

Note: Codes 06, 07, and 08 require SDSF to be running on the mainframe agent, and will require the extra DD cards `ISFIN` and `ISFOUT` to be uncommented in the VP390 startup JCL.

The available OSINFO codes are:

01 DASD Utilization Statistics

Collects DASD volume statistics. The DASD must be online at the time of the request.

Parameters: DASD volume name, or a regular expression to look for multiple volumes, or * for all volumes.

Output: One line for each DASD volume found, in the format:

-VolSer

-Number of tracks

-Tracks per cylinder

-Free extents

-Free tracks

-Largest free extent

-Percent used

-DSCBs

02 RMF Address Space Resource Statistics

Collects statistics from RMF for a specified address space(s). RMF must be running on the system for this option to collect.

Parameter: Address space name, or a prefix of address space with an * to find multiple address spaces with the same starting characters.

Output: one line for each address space found in the format:

-Job Name

-Device connect time in milliseconds

-Number of fixed frames located below the 16M real line

-Number of non-LSQA fixed frames

-LSQA pages in real storage

-Total TCB time for this step in milliseconds

-Total CPU time consumed on behalf of this address space in milliseconds

-EXCP count for this step

03 Current CPU Snapshot for System and a Specific Address Space

Collects CPU and memory usage for the system and a specific address space by scheduling an SRB to execute in the target address space.

Parameter: Address space name.

Output: One line of values in the format

-Current total LPAR CPU utilization percentage

-Percentage of CPU used by specified address space

-Total CPU time used by address space in seconds

-Real storage used by address space in kilobytes

-Extended stage used by address space in kilobytes

-Region size requested in kilobytes

-Private storage allocated under the 16M line

-Private storage allocated above the 16M line

-Private storage used under the 16M line

04 Current Active Jobs

Collects a list of active address spaces.

Parameter: Regular expression filter of address space names to be displayed, or "*" for all.

Output: One line for each address space found, in the format:

-Job name

-Step name

-Proc step

-Job ID

-Owner

-Position

-Performance Group number

-Priority

-Current real storage usage in frames

05 System statistics from RMF

Collects current system statistics as reported by RMF type 79 subtype 3, subtype 4, and subtype 9 records. RMF must be running to get a valid output.

Parameter: none

Output: One line of output in the format:

-System CPU utilization percentage

-System demand paging rate

-Number of system common (:PA+CSA) pages in

-Number of swaps (out)

-Number of pages swapped in

-Number of pages swapped out

-Number of private pages swapped in

-Number of private pages swapped out

-High UIC count

-System LPA pages in

-Number of pages to extended storage

-Number of extended storage slots available and not in use

-Number of pages migrated from extended storage to auxiliary storage

-Number of available frames

-I/O activity rate: average I/O requests per second

-I/O response time: average milliseconds needed to complete an I/O request

-ISOQ time: average milliseconds an I/O request must wait on an IOS queue

-Number of fixed SQA frames

-Number of common (LPA+CSA) frames

-Number of private non-LSQA fixed frames

-Number of address spaces in storage

-Number of total LPA frames

-Number of total CAS frames

-Number of LPA fixed frames

-Number of CSA fixed frames

-Number of fixed LSQA frames

-Number of address spaces logically swapped out

06 JES2 Input Queue

Collects a list of jobs on the JES2 Input Queue. See the note above for extra requirements to run this option.

Parameter: Job name, or a prefix of a job name with an * to find multiple jobs with the same starting characters.

Output: One line for each job found, in the format:

-Job name

-Job ID

-Owner

-JES2 input queue priority

-JES2 input class

-Position within JES2 input queue class

-Print designating name

-Print routing

-Print node

-System affinity (if any)

07 JES2 Output Queue

Collects a list of jobs on the JES2 Output Queue. See the note above for extra requirements to run this option.

Parameter: Job name, or a prefix of a job name with an * to find multiple jobs with the same starting characters.

Output: One line for each job found, in the format:

-Job name

-Job ID

-Owner

-JES2 output group priority

-JES2 output class

-Output form number

-Print destination name

-Output total record count (lines)

-Output creation due

08 JES2 Held Queue

Collects a list of jobs on the JES2 Held Queue. See the note above for extra requirements to run this option.

Parameter: Job name, or a prefix of a job name with an * to find multiple jobs with the same starting characters.

Output: One line for each job found, in the format:

-Job name

-Job ID

-Owner

-JES2 output group priority

-JES2 output class

-JES2 output disposition

-Print destination name

-Output total record count (lines)

-Output creation date

Using RMF Performance and DASD Data Collection

RMF Performance or DASD volume statistics data is stored using either the OpenView Agent performance sub-agent (CODA) or the OpenView Performance Agent (formerly known as the MeasureWare agent) on the OpenView Management Server.

The PERF subtask of the VP390 agent task collects a subset of RMF performance data and forwards this data to the OV OS/390 message server (`vp390mms`). The message server stores the performance data in either the CODA performance sub-agent data store, or in the configured logfile set of the OpenView Performance Agent (OVPA).

To collect DASD statistics you must configure collection using a Scheduled Action template as described in Chapter 1.

It is important to note that data from individual OS/390 systems (LPARs) will be stored in a separate data classes. To access the data for an individual system you must specify the data class for that system. The data class naming conventions are:

`OS390_`*`HOSTNAME`*`:OS390_`*`HOSTNAME`*
for RMF performance data, where *`HOSTNAME`* will be the actual hostname.

`DASDUMM_`*`HOSTNAME`* `:DASDSUMM:`*`HOSTNAME`*
for summary DASD statistics, where *`HOSTNAME`* will be the actual system hostname.

`DASDSTAT_`*`HOSTNAME`* `:DASDSTAT_`*`HOSTNAME`*
for detailed DASD statistics, where *`HOSTNAME`* will be the actual system hostname.

Stored data is available for reporting/graphing through the OpenView Performance Manager product. Stored data may also be used with the OpenView Reporter product. See Appendix C for available data classes.

3 Troubleshooting OV OS/390

This chapter describes how to troubleshoot problems with HP OpenView Operations OS/390 Management (OV OS/390).

General Troubleshooting

Before you troubleshoot a particular problem you run into when installing, configuring, or using OV OS/390, you should verify that your environment is correctly installed and configured.

Correct installation and configuration of VP390 ensures, among other things, that messages are processed correctly:

- **Message Generation**

  Messages are generated by the OV OS/390 system.

- **Message Interception**

  Messages are intercepted by the OV OS/390 templates and monitors.

- **Message Browser**

  Messages appear in the OVO Mess`age Browser` in the form you expect.

Specific Troubleshooting

This section explains how to solve specific problems you may encounter when using OV OS/390.

If No Messages on the OVO Management Server

Symptom

No S/390 mainframe messages are arriving on the OVO management server.

Solution

1. Verify that the appropriate OV OS/390 server programs are running on the OVO management server by entering the following command:

   `vp390sv`

2. Verify that the OVO agent has been correctly installed and configured on the OVO management server.
3. Verify that the OVO agent processes (in particular, the control agent) are running.
4. Verify that the VP390 templates have been correctly assigned and distributed to the OVO management server.
5. Verify that the `VP390` node group has been assigned to the appropriate OVO operators.
6. Verify that the VP390 services have been assigned to the appropriate OVO operators.

If Automatic or Operator-initiated Actions Do Not Complete

Symptom

Messages appear in the `Message Browser` (with a status of `Running`), but automatic or operator-initiated actions do not complete.

Solution

1. Verify that the `vp390elli` process is running under OpenView by entering the following command:

   `ovstatus`

2. Recycle the `vp390elli` by entering the following commands:

   `ovstop vp390elli`

   `ovstart vp390elli`

A OS/390 Console Commands

This chapter explains OS/390 console commands used to display and change maintenance information about the mainframe job.

In this Appendix

OV OS/390 has several OS/390 console commands that enable operators to display and change maintenance information about the present mainframe job. Commands are sent from an OS/390 console to OV OS/390 using the `MODIFY` command.

If the OV OS/390 job name is `VP390`, the syntax for a console command is:

`MODIFY VP390,`*`command`*

This appendix explains the following types of OS/390 commands:

-SHOW commands

-Subtask control commands

-FILTER commands

-SUPPRESS commands

-PERF commands

About SHOW Commands

SHOW commands display the requested information in a formatted table.

SHOW TASK

Displays each of the defined subtask, their status, number of times the subtask was restarted, maximum number of automatic restart attempts for the subtask, and any unique information for the subtask.

Subtask Status

UP	Subtask is active and can accept messages.
DOWN	Subtask is down and is not restarting.
DOWNR	Subtask is down but is restarted after a delay.
INIT	Subtask is initializing.
QUIES	Subtask is in a quiescent state, cleaning up outstanding allocated memory before going into the DOWN or DOWNR state.

Example

```
MODIFY VP390,SHOW TASK
EVO595  Command entered: SHOW TASK
EVO600  TNUM TASKNAME STATUS RESTARTS/LIMIT  SPECIFIC
EVO600    1  PPO        UP          0     5  EVOPPO1
EVO600    2  CNM        UP          0     5  DSICRTR
EVO600    3  SPO-0      UP          0     5  EVOSPO1
EVO600    4  SEC        UP          0     5  EVRACF
EVO600    6  TCP-0      UP          0   100  6106,6107
EVO600    7  TCP-1     DOWNR        4   100  6116,6117
EVO600    8  MVS        UP          0     5  EVOCONS1,03000001
EVO600    9  CMD-0      UP          0     5  EVOCONS2,01000002
EVO695  VP390 SHOW command processed
```

SHOW ADDR

Displays the memory address of each defined subtask internal header control block, subtask control block, and OS/390 Task Control Block. This information is useful if you anticipate making an address space dump.

Parameters

None

Example

```
MODIFY VP390,SHOW ADDR
EVO595  Command entered: SHOW ADDR
EVO603  TNUM TASKNAME  ADDRESS    HEADER     TCB
EVO603    0  MAINTASK  00000000  05A350C8  00000000
EVO603    1  PPI       05A1BF6C  05A3F008  008F10E0
EVO603    2  SPO-0     05A1BF88  05A410C8  008DDD90
EVO603    3  SEC       05A1BFC0  05A53748  008CD658
EVO603    5  TCP-0     05A1C014  05A7B808  008CDE88
EVO603    6  MVS       05A1C04C  05A817C8  008C5E88
EVO603    7  CMD-0     05A1C068  05A837C8  008C5C58
EVO695  VP390 SHOW command processed
```

SHOW VERSION

Displays the version of VP390 running and the compile date of each subtask.

Parameters

None

Example

```
MODIFY VP390,SHOW VERSION
EVO595  Command entered: SHOW VERSION
EVO607  VP/390 V3.0 Copyright 2001 EView Technology, Inc.
EVO608  TASKNAME     DATE         TIME
EVO608  MAINTASK  Dec 29 2000   15:08:22
EVO608  SPO-0     Dec 29 2000   15:52:23
EVO608  SEC       Dec 29 2000   15:19:18
EVO608  MVS       Dec 29 2000   15:44:21
EVO695  VP390 SHOW command processed
```

SHOW FLOW

Displays the number of messages for each subtask on the input and output queues, the total number of messages that flowed in and out of the subtask, and the number of memory allocations currently outstanding.

Parameters

None

Example

```
MODIFY VP390,SHOW FLOW
EVO595  Command entered: SHOW FLOW
EVO605  TNUM TASKNAME  INPUTQ  OUTPUTQ  INFLOW  OUTFLOW   MC
EVO605    1  PPI            0        0       3        3    0
EVO605    2  SPO-0          0        0     729       35    0
EVO605    3  SEC            0        0       3        3    0
EVO605    5  TCP-0          0        0      11      249    0
EVO605    6  TCP-1          0        0       0        0    0
EVO605    7  MVS            0        0     366      366    0
EVO605    8  CMD-0          0        0       0        0    0
EVO695  VP/390 SHOW command processed
```

SHOW SUPPRESS

Displays a list of VP390 message IDs that were suppressed from printing using the `SUPPRESS SYSIN` command or the `SUPPRESS Modify` command.

Parameters

None

Example

```
MODIFY VP390,SHOW SUPPRESS
EVO595  Command Entered: SHOW SUPPRESS
EVO615  Suppressed message IDs:
EVO615  002, 902, 905
```

About Subtask Control Commands

Subtask control commands allow you to manually control the status of the subtask. VP390 subtask start automatically when the job is started, and the subtask restart automatically if brought down by some anomaly.

NOTE

For more information on automatic subtask restarts, see the description of the DELAY and RESTART input parameter cards in the *HP OpenView OS/390 Management Installation Guide.*

INIT

Activates a defined subtask that is in a `DOWN` state. This command can also be used when a subtask is in the `DOWNR` state to skip the rest of the timed delay and force the re-initialization to continue immediately. The `INIT` command can only activate tasks that are listed in the `SHOW TASK` table.

Parameters

`subtaskname`

Example

```
MODIFY VP390,INIT TCP-0
EVO595  Command entered: INIT TCP-0
EVO002  TCP subtask initialized for 6106,6107
```

KILL

Forces the termination of a defined subtask. When a subtask is terminated with this command, it does not attempt any automatic restarts. The command resets the count of number of automatic restarts that are attempted. The command can also be used to stop a subtask in the DOWNR state from attempting any more restarts.

Parameters

subtaskname

Example

```
MODIFY VP390,KILL SPO-1
EVO595  Command entered: KILL SPO-1
EVO902  SPO-1 subtask terminated, RC = 0
```

TERM

Stops all subtask and then stops the main task, terminating the VP390 job. This command is identical to the OS/390 STOP command.

Parameters

None

Example

```
MODIFY VP390,TERM
EVO595  Command entered: TERM
EVO690  VP390 STOP Command accepted
EVO901  Stopping subtask #1: PPI
EVO901  Stopping subtask #2: SPO-0
EVO901  Stopping subtask #3: SEC
EVO901  Stopping subtask #4: TCP-0
EVO901  Stopping subtask #5: MVS
EVO695  VP390 STOP command processed
EVO902  PPI subtask terminated, RC = 0
EVO902  SPO-0 subtask terminated, RC = 0
EVO902  SEC subtask terminated, RC = 0
EVO902  TCP-0 subtask terminated, RC = 0
STOPMN JOBNAMES
STOPMN SESS
IEA631I  OPERATOR EVOCONS1 NOW INACTIVE, SYSTEM=MVS01,
LU=EVOMESG
EVO902  MVS subtask terminated, RC = 0
EVO904  All VP390 subtasks complete
IEF404I VP390 - ENDED - TIME=17.30.08
$HASP395 VP390 ENDED
```

About FILTER Commands

The FILTER commands listed below make use of the VP390 feature that restricts the flow of OS/390 and VTAM messages and alerts to the OVO Server.

SHOW FILTER

Displays all message IDs in the message filter table and all alerts in the alert filter table. The message filter table is checked by the PPO or PPI subtask as VTAM messages are received, and it is checked by the MVS subtask as OS/390 messages are received. If the message table has any entries, then only the messages listed in the message table are forwarded to the workstation. The alert filter table is checked by the CNM or PPI subtask as alerts are received. In contrast to the message table, only alerts matching the ones listed in the alert table are blocked from being passed to the OVO server.

Parameters

None

Example

```
MODIFY VP390,SHOW FILTER
EVO595 Command entered: SHOW FILTER
EVO609 Message filters:
EVO609 IST105I IST093I
EVO609 Alert filters:
EVO609 8  41038D
EVO609 120  'LINE'
EVO695 VP390 SHOW command processed
```

FILTER ADD

Adds a message ID or a CNM alert to the appropriate filter table. The message table holds up to 2,000 message IDs, and each message ID can be up to 15 characters long, and may be specified with a period (.) as a placeholder for any character within the message ID, or with an asterisk(*) at the end of any message ID to match any message with the same beginning character(s).

Multiline messages can be delivered to the OVO server for display and logging in three possible formats:

1. By default, the multiple lines will be concentrated together into one single message as seen by the OVO server.
2. Prepend a plus sign “+” to the front of the message ID to send the multiple lines as separate messages to the OVO server, but with the original message ID prepended to the beginning of all secondary message lines.
3. Prepend a minus sign “-” to the front of the message ID to send the multiple lines as separate messages to the OVO server, but without the original message ID added to the beginning of the secondary message lines.

Messages can also be filtered based on the job name or job ID of the job (if any) which issued the message. Jobnames and job IDs may be specified with a period(.) as a placeholder for any character within the value, or with an asterisk(*) at the end of any value to match any jobname or job ID with the same beginning character(s). If job name(s) or job ID(s) are specified for a message ID, then the message ID will only pass the filtering if the given job name(s) or job IDs are attached to that message.

The alert table holds up to 2,000 alert filter definitions, and the data portion of the alert definition can be up to 50 bytes long. The offset can be expressed in decimal, or in hexadecimal if preceded by an X. The data can be expressed in text if it is enclosed in single quote marks, or in hexadecimal. Hexadecimal data must be an even number of characters (2 characters per byte).

Syntax

```
FILTER ADD MSG msgid [JOBNAME=(jobname[,jobname]...)]
    [JOBID=(jobid[,jobid]...)]

FILTER ADD CNM offset data
```

Examples

* Permit message IST105I to be passed on the the OVO server:

MODIFY VP390,FILTER ADD MSG IST105I

```
EVO595 Command entered: FILTER ADD MSG IST105I
EVO610 Message filter IST105I Added
```

*Permit any message that has “HASP” in the second through fifth character of the message ID:

MODIFY VP390,FILTER ADD MSG .HASP*

```
EVO595 Command entered: FILTER ADD MSG .HASP*
EVO610 Message filter .HASP* Added
```

*Permit any message that begins with “QWE” and was issued by a job with a name that begins with “BB” or “CC”

MODIFY VP390,FILTER ADD MSG QWE* JOBNAME=(BB*,CC*)

```
EVO595 Command entered: FILTER ADD MSG QWE* JOBNAME=(BB*,CC*)
EVO610 Message filter QWE* Added
```

*Restrict any alerts which have the string “LINE6” anywhere in the alert:

MODIFY VP390, FILTER ADD CNM * 'LINE6'

```
EVO595 Command entered: FILTER ADD MSG .HASP*
EVO610 Alert filter 'LINE6' Added
```

*Restrict any alerts which have the hexadecimal values “41038D” at offsets 8 to 10 within the alert:

MODIFY VP390, FILTER ADD CNM X8 41038D

```
EVO595 Command entered: FILTERD ADD CNM Z8 41038D
EVO610 Alert filter 41038D Added
```

FILTER DEL

Deletes a message ID from the message filter table, or deletes a CNM alert from the alert filter table. Specifying ALL deletes all filters from both the message and filter tables.

Syntax

FILTER DEL MSG *msgid*

FILTER DEL CNM *offset data*

FILTER DEL ALL

Examples

MODIFY VP390,FILTER DEL CNM 120 'LINE'

EVO595 Command entered: FILTER DEL CNM 120 'LINE'

EVO610 Alert filter deleted

MODIFY VP390,FILTER DEL MSG IST105I

EVO595 Command entered: FILTER DEL MSG IST105I

EVO610 Message filter IST105I deleted

MODIFY VP390,FILTER DEL ALL

EVO595 Command entered: FILTER DEL ALL

EVO613 All message and alert filters deleted

About SUPPRESS Commands

The SUPPRESS commands enable you to control the printing of VP390 messages that announce subtask connections, disconnections, or other changes internal to the VP390 address space. Message suppression may be useful in reducing the number of VP390 messages written to the system log.

SUPPRESS

Suppresses the display of the named VP390 message ID. The *msgid* variable must be the numerical extension of the message ID (no more than three digits).

Syntax

SUPPRESS *msgid*

Example

To suppress the printing of VP390 message EVO038:

```
MODIFY VP390,SUPPRESS 38
EVO595 Command entered: SUPPRESS 38
EVO616 Added suppression of 38
```

About DOM Commands

The DOM command allows you to control the delivery of MVS Delete Operator messages to the OVO server, which can be used to automatically acknowledge messages on the OVO browser when they are deleted from the MVS console. This command provides the same function as using the “DOM” parameter on the MVS SYSIN parameter card, but you do not need to restart the VP390 address space to have it take effect. This command is only valid when a MVS subtask is defined in the VP390 job. The information in a DOM is conveyed to the OVO server in an EVO211 message to allow message correlation. The EVO211 message ID does not need to be listed in the mainframe message filter table.

Syntax

DOM switch

where *switch* is:

ON	allow the MVS DOMs to be passed to the OVO server
OFF	suppress the delivery of DOMs to the OVO server

Examples

```
MODIFY VP390,DOM ON
EVO595 Command entered: DOM ON
EVO214 DOM flag updated
MODIFY VP390,DOM OFF
EVO595 Command entered: DOM OFF
EVO214 DOM flag updated
```

About PERF Commands

The PERF commands enable you to dynamically change the performance data gathering interval or update the RMF cycle time. The PERF command is only relevant when the PERF subtask is initialized (for reporting interval updates and RMF cycle updates) and/or the OSINFO subtask is initialized (for RMF cycle updates). Using the PERF command will only affect the currently running VP390 job; you must edit the initialization SYSIN cards RMFCYCLE or PERF to retain the new values when the VP390 job is recycled. (See the *OV OS/390 Installation Guide* for syntax of the RMFCYCLE and PERF cards.) When updating INTERVAL, the new interval will take effect after the current interval time has elapsed.

PERF

Change the performance collection interval or the RMF cycle time. (This command only affects statistic gathering by VP390. It does not make any modifications to RMF.)

Syntax

```
PERF INTERVAL=min
PERF CYCLE=msec
```

where:

min	number of minutes to wait before sending and RMF statistics report. Valid values are 1 through 35791394.
msec	number of milliseconds in an RMF cycle. Valid values are 50 through 9999.

Examples

To change the PERF subtask to send a system statistics report every 12 minutes:

```
MODIFY VP390,PERF INTERVAL=12
EVO595 Command entered: PERF INTERVAL=12
EVO215 PERF parameter updated
```

If the CYCLE parameter of RMF has been changed to 2500 milliseconds, update VP390 to use the new value in its statistical calculations:

```
MODIFY VP390,PERF CYCLE=2500
EVO595 Command entered: PERF CYCLE=2500
EVO215 PERF parameter updated
```

B VP390 Mainframe Messages

This chapter describes all messages generated by the `OV OS/390` job (VP390) running on the S/390 mainframe.

Messages

EVO002 *type* subtask initialized for *feature*

Message Variables

type Type of subtask

feature A specific attribute that this subtask is intialized for:

Subtask	Attribute description
CMD	extended MCS console name
CNM	VTAM resource contacted
NOMATCH	dataset name (and member) to be written
MVS	extended MCS console name
OSI	OS/390 system name
PERF	OS/390 system name
PPI	“PPI”
PPO	VTAM resource contacted
SEC	defined security application name
SPO	VTAM resource contacted
TCP	initialized TCP/IP port numbers

Message Description

The VP390 subtask is successfully initialized. This message will be issued for each of the defined subtasks of the VP390 main task.

System Action

Processing continues.

User Action

None.

EVO008 Invalid input parameter card on line *number*

Message Variables

number Line number of SYSIN

Message Description

VP390 read a line from SYSIN that it did not understand.

System Action

The invalid card is skipped. Processing continues with the next

SYSIN card.

User Action

Correct the input card on the given line number of SYSIN. Valid syntax for SYSIN cards are listed in the *HP OpenView Operations OS/390 Management Installation Guid*e. All other lines must begin with an asterisk (*) to denote a comment line.

EVO009 Duplicate *subtask* card on line *number* ignored

Message Variables

subtask Type of subtask

number Line number of SYSIN

Message Description

VP390 read a definition card from SYSIN for a subtask that has already been defined.

System Action

The invalid card is skipped. Processing continues with the next SYSIN card.

User Action

Correct or remove the input card on the given line number of SYSIN. For names of input parameter cards that may be defined multiple times, see the *HP OpenView Operations OS/390 Management Installation Guid*e.

EVO010 Maximum number of *subtask* cards reached; ignoring line *number*

Message Variables

subtask Type of subtask, or subtask

number Line number of SYSIN

Message Description

VP390 has reached the maximum number of subtasks of the type named. The definition card on the named line is not processed. If *subtask* is “subtask”, VP390 has reached the maximum number of total subtasks that can be defined, and all SYSIN parameter cards

from the current line number forward are ignored.

System Action

The parameter cards are skipped and processing continues.

User Action

Decrease the number of `SYSIN` parameter cards of the type named.

EVO011 Maximum number of *type* filter entries reached; new entry ignored

Message Variables

`type` Type of filter entry

Message Description

VP390 has reached the maximum number of filter entries allowed. Currently, VP390 will accept up to 2000 message ID entries, and up to 2000 CNM alert entries.

System Action

The new filter entry is discarded and processing continues.

User Action

Decrease the number of filter entries, possibly by combining multiple entries using wildcard characters.

EVO018 VTAM ACB generation for *subtask acb* failed, RC = *rcnumber*

Message Variables

`subtask` Type of subtask

`acb` Name of failing ACB

`rcnumber` Return code from the Get VTAM ACB routine

Message Description

An attempt by an initializing subtask to get a VTAM ACB failed.

System Action

The VP390 subtask terminates with a condition code 8.

User Action

Verify that the ACB is available. Use the `INIT` command to restart the subtask.

EVO019 VTAM *subtask* **open for** *acb* **failed, RC =** *rcnumber*, **error =** *enumber*

Message Variables

subtask Type of subtask

acb Name of failing ACB

rcnumber Return code from the Open VTAM ACB routine

enumber Error code within ACB

Message Description

An attempt by an initializing subtask to open a VTAM ACB failed.

System Action

The VP390 subtask terminates with a condition code 8.

User Action

Verify that the ACB name*acb* is correctly defined. If *rcnumber* = 8, then the subtask may be restarted using the `INIT` command. If *rcnumber* = 12, then there is a serious VTAM error which will not allow a re-issue of the ACB open command; check the status of VTAM and recycle the VP390 job. If *enumber* = 88, then resource *acb* is already in use by another program. (Remember that the `PPO` and `CNM` subtasks should not be used if NetView is running.) For descriptions of other error codes, see the section for the OPEN macro-instruction in the IBM manual *VTAM Programming*.

EVO020 *subtask* **is currently in use**

Message Variables

subtask Type of subtask

Message Description

This message follows immediately after the `EVO019` message if an exclusive subtask ACB is already in use by another program.

System Action

The VP390 subtask terminates with a condition code 8.

User Action

Verify that the ACB is not taken by another program on the mainframe, such as NetView/390 or SOLVE:NETMASTER.

EVO021 Unsolicited *msgtype* data is unavailable

Message Variables

msgtype Type of message

Message Description

This message follows the EVO019 message to alert you that VP390 is not able to receive unsolicited data because it was unable to access an ACB.

System Action

The VP390 subtask terminates with a condition code 8.

User Action

Correct the problem identified by the EVO019 message, then restart the subtask.

EVO026 Unexpected *subtask* return code, RC = *rcnumber*

Message Variables

subtask Type of subtask

rcnumber Return code from Receive routine

Message Description

The subtask Receive routine received an unexpected return code while attempting to receive messages.

System Action

The VP390 subtask terminates with a condition code 9.

User Action

Check the mainframe job output log for additional messages. Use the INIT command to restart the subtask.

EVO033 VP390 COMMAND = *command*

Message Variables

command Command text

Message Description

The command issued through VP390 is logged to SYSLOG.

System Action
Processing continues.
User Action
None.

EVO034 Initialization of SPO *name* failed in *reqtype* processing, RC1 = *addr* RC2 = *size*

Message Variables
name Name of the SPO subtask
reqtype Type of request being processed
addr Returned address from Get RPL routine
size Returned size from Get RPL routine
Message Description
The SPO subtask failed calling the VTAM RPL routine.
System Action
The VP390 SPO subtask terminates with a condition code 13.
User Action
Use the INIT command to recover subtask.

EVO035 SPO Warning: Failure retrieving command responses, max retries reached.

Message Variables
None.
Message Description
The VP390 Secondary Program Operator interface subtask encountered a failure while attempting to retrieve the command responses from an issued VTAM command. Not all responses were retrieved.
System Action
Processing continues.
User Action
Re-issue the VTAM command. If the proper responses are still not returned, contact Hewlett-Packard support.

EVO036 SPO command queue depth exceeded maximum

Message Variables
None.
Message Description
A VTAM SPO command could not be placed on the VP390 queue of waiting SPO commands because that queue has reached its maximum size.
System Action
The command is discarded.
User Action
Re-issue the VTAM command. If this message appears frequently, consider defining additional SPO subtasks to handle the load (VP390 allows up to ten SPO subtasks to be defined in the `SYSIN` cards).

EVO038 VTAM command support unavailable

Message Variables
None.
Message Description
VP390 is not able to process the VTAM command because there are no active Secondary Program Operator subtasks.
System Action
The command is discarded.
User Action
Add one or more `SPO` cards to the `SYSIN` deck, identifying a valid VTAM appl resource to be used as a SPO. Verify that the named VTAM resource is active before restarting VP390.

EVO039 Unable to route message (type=*type*)

Message Variables
type Invalid message type
Message Description
VP390 was unable to route an incoming message to any of its subtasks because the message type was unrecognized.

System Action

The invalid message is dumped to SYSPRINT immediately after this message.

User Action

Capture the SYSPRINT information and contact Hewlett-Packard support.

EVO091 PPI initialization failed, step = *stepnum* RC = *rcnumber*

Message Variables

stepnum Initialization step that failed:

1 SSI is not running

2 Attempt to get ASCB value failed

3 Attempt to register receiver failed

rcnumber Return code from call to CNMNETV

Message Description

An attempt by the PPI subtask to access the CNMNETV module failed.

System Action

The PPI subtask terminates with a condition code 6.

User Action

If *stepnum* = 1, check the status of the SSI address space. If *stepnum* = 2, use the NetView DISPLAY PPI modiFy command to verify that the NetView program-to-program interface is active. If *stepnum* = 3, verify that no other application is attached to the NetView/390 or NETMASTER PPI.

EVO095 VP390 PPI buffer size error, RC = *rcnumber*

Message Variables

rcnumber Return code from PPI call

Message Description

A Receive request for the PPI failed because the allocated buffer size was not large enough to hold the incoming data.

System Action

The VP390 PPI subtask terminates with a condition code 31.

User Action

Use the INIT command to restart the subtask.

EVO096 VP390 PPI interface failed, ID = *requestid*, RC = *rcnumber*

Message Variables

requestid ID of task request

rcnumber Return code from PPI call

Message Description

A Receive request for the PPI failed.

System Action

The VP390 PPI subtask terminates with a condition code 11.

User Action

For explanations of return codes, see the *TME 10 NetView for OS/390 Application Programmer's Guid*e. If *requestid* = 22 and *rcnumber* = 25, then add BUFLEN=40 to the PPI card in SYSIN.

EVO119 *count* messages queued on *subtask*. Command rejected: *cmd*

Message Variables

count Number of messages

subtask Subtask name

cmd Command entered

Message Description

Subtask *subtask* does not process the command issued from the workstation because there is a backlog of *count* messages waiting to be sent to the workstation.

System Action

The command *cmd* is discarded. Processing continues on the remaining messages in the subtask queue.

User Action

Wait until the existing backlog of messages is processed, then re-issue the command. Use the mainframe VP390 modiFy command `SHOW TASK` to view the number of messages in the Output Queue of the subtask.

EVO121 MVS console *name* could not obtain a migration ID

Message Variables

name Name of console to be defined

Message Description

The MVS console being defined requested a one-byte migration ID, but the console initialization routine was unable to provide one.

System Action

Initialization of the console continues.

User Action

None.

EVO122 MVS console *name* initialization failed, RC = *rcnumber*

Message Variables

name Name of console to be defined

rcnumber Return code from initialization routine

Message Description

The initialization of the MVS console failed.

System Action

The VP390 `MVS` subtask terminates with a condition code 8.

User Action

Verify that all the parameters on the MVS `SYSIN` card conform to the syntax rules. If *rcnumber* = 4, then a console *name* is already running. If you are running multiple OV OS/390 agents on mainframes or LPARs in a sysplex, then one mainframe image may be able to see another's consoles. Use a unique name for each agent's MVS card in its `SYSIN` deck.

EVO126 Unable to open MSGCATLG message file

Message Variables
None.
Message Description
The VP390 main task could not find or open the messages file, which is identified by the MSGCATLG DD card in the VP390 startup JCL.
System Action
The VP390 task terminates.
User Action
Verify that the MSGCATLG DD card is defined in the VP390 started task JCL and points to a readable message file. Restart the VP390 job.

EVO127 Too many messages in MSGCATLG message file

Message Variables
None.
Message Description
The VP390 messages file, identified by the MSGCATLG DD card in the VP390 startup JCL, contained more lines than expected for a valid messages file.
System Action
The VP390 task terminates.
User Action
Verify that the MSGCATLG file does not contain extra non-blank lines which could be misinterpreted for message lines. Comment lines beginning with an asterisk and blank lines in the file are ignored. Restart the VP390 job.

EVO128 Unable to find message ID *msg* in MSGCATLG file

Message Variables
msg Message ID to be written
Message Description
VP390 attempted to issue a message with the message ID *msg* but could not find this message ID in the MSGCATLG file.

System Action
Processing continues.
User Action
Verify that the file identified by the MSGCATLG DD in the VP390 startup JCL contains message text for the ID *msg*. In the MSGCATLG file, message IDs must start in the first column of each line. Restart the VP390 job to re-read the messages file.

EVO130 Unrecognized command option: *code*

Message Variables
code Option number
Message Description
The `vp390hostcmd` utility on the OVO server sent a type 46 command with an option code that the mainframe OSINFO subtask did not recognize.
System Action
Processing continues.
User Action
Consult the *OV OS/390 Administrator's Reference* for valid options for OSINFO system information. See the `man` page for correct syntax of the `vp390hostcmd` utility.

EVO131 Query failed, error code = *code*

Message Variables
code Error code
Message Description
The `vp390hostcmd` utility on the OVO server sent a type 46 command requesting information that could not be supplied by the OSINFO subtask on the mainframe.
System Action
Processing continues.
User Action
The *code* can have different meanings depending on the type 46

option that was requested. Identify what command request is being issued and contact Hewlett-Packard support.

EVO132 Query returned no lines

Message Variables

None.

Message Description

The `vp390hostcmd` utility on the OVO server sent a type 46 command that returned no output. This can be caused by improper syntax on the 46 command, or by specifying a non-existent task name or DASD volume.

System Action

Processing continues.

User Action

Check the syntax and parameters of the `vp390hostcmd` which was sent to the mainframe.

EVO133 Unable to collect *queue* queue data: error accessing *source*, rc=*code*

Message Variables

`queue` Queue name to gather information from: “INPUT”, “OUTPUT”, or “HELD”

`source` Resource that could not be accessed: “ISFIN”, “ISFOUT”, or “SDSF”

`code` Return code

Message Description

The `vp390hostcmd` utility on the OVO server sent a type 46 command requesting information from one of the JES2 queues that could not be supplied.

System Action

Processing continues.

User Action

If *`source`* is “ISFIN” or “ISFOUT”, verify that the `ISFIN` and `ISFOUT` DD cards are correctly defined in the VP390 startup JCL.

The *code* can have different meanings depending on the type 46 option that was requested. Identify what command request is being issued, and contact Hewlett-Packard support.

EVO150 TCP/IP communications: *function* for workstation *component* agent failed with errno *value*

Message Variables

function Failing communication function

component Workstation component that detected the failure

value Integer error value

Message Description

A TCP/IP communications error occurred. The error could have occurred while TCP/IP communication was being established or while a message was sent or received by the mainframe or specified agent.

System Action

The VP390 TCP subtask terminates with a condition code 1.

User Action

Verify the availability of TCP/IP communications between the workstation and the mainframe, and verify the mainframe TCPIP job's high-level qualifier is specified correctly on the TCP card in the VP390 SYSIN deck. Use the INIT command to recover the TCP subtask, or recycle the VP390 job if the SYSIN needs modification.

EVO151 VP390 failure in communication to TCP/IP

Message Variables

None.

Message Description

VP390 received an error while attempting to receive data from a TCP/IP socket or
ECB.

System Action

The TCP subtask terminates.

User Action

Use the INIT command to recover the subtask.

EVO152 Default TCP/IP *function* failed

Message Variables

function Failing communication function

Message Description

The setup of a default TCP/IP environment failed which performing *function*.

System Action

Processing continues, but initialization of subsequent TCP subtasks may fail.

User Action

Verify the mainframe TCPIP job's high-level qualifer is specified correctly on the TCP card in the VP390 SYSIN deck. Recycle the VP390 job if the SYSIN needs modification.

EVO153 Message length exceeds send buffer allocation

Message Variables

None.

Message Description

The TCP subtask could not send out a block of data because it was longer than the standard VP390 data buffer could hold.

System Action

The message is discarded.

User Action

Note the system message and alert activity at the time this message was issued, and contact Hewlett-Packard support.

EVO154 *server* Server connection lost on port *number*

Message Variables

server OV OS/390 server process on the OVO server

number Port number

Message Description

The mainframe agent lost its connection to the OVO server.

System Action

The port *number* is reset to allow re-connections. If message buffering is active, mainframe messages will be written to the buffer file until the connection to the OVO server is re-established.

User Action

Use the `vp390sv` command on the OVO server to verify the OV OS/390 processes are running.

EVO155 *server* Server connection established on port *number*

Message Variables

server OV OS/390 server process on the OVO server

number Port number

Message Description

The mainframe agent has made a connection to the *server* process on the OVO server.

System Action

Processing continues.

User Action

None.

EVO160 Console command return code = *rcnumber*

Message Variables

rcnumber Return code from command `Send` subroutine

Message Description

An MVS command request completed with a non-zero return code.

System Action

Processing continues.

User Action

If expected command response is not received, record the return code

and contact
Hewlett-Packard support.

EVO162 No valid DD names for message logging subtask

Message Variables
None.
Message Description
No valid log file DD names were specified for the NOMATCH subtask.
System Action
The NOMATCH subtask is terminated.
User Action
Add appropriate DD names to the NOMATCH line in SYSIN, and verify that the DD names are defined in the VP390 startup JCL. Recycle the VP390 job.

EVO163 Unable to open message logging file *ddname*

Message Variables
ddname DD name of the file
Message Description
The NOMATCH subtask was unable to open the logging dataset *ddname* named on the SYSIN card for the NOMATCH initialization.
System Action
The NOMATCH subtask attempts to open the next dataset in the list.
User Action
Verify that the DD name given on the SYSIN card has a matching DD card in the VP390 startup JCL. Verify that the dataset named for that DD name is defined with the DCB values stated in the *OV OS/390 Installation Guide*.

EVO164 Message logging is closing *dataset*

Message Variables
dataset Log dataset name. If the log is a PDS member, the mem-

ber name will be appended to the dataset name in parentheses.

Message Description

The NOMATCH subtask is closing the dataset logging dataset, either because of subtask termination or because an attempt to write to the dataset failed (usually because the dataset has been filled.)

System Action

If the dataset closing was due to a write failure, the NOMATCH subtask attempts to open the next dataset in its list of defined DDs.

User Action

None.

EVO165 Message logging is wrapping to the first file

Message Variables

None.

Message Description

The NOMATCH subtask has reached the end of its list of valid logfile DD names.

System Action

The NOMATCH subtask wraps back to re-open the first DD in the list. The existing data in that logfile will be purged and overwritten.

User Action

None.

EVO170 Unable to open message buffering file *ddname*

Message Variables

ddname DD name of the file

Message Description

The message buffering facility was unable to open the dataset *ddname* for buffering messages while the TCP/IP connection to the OVO server is down.

System Action

No message buffering will occur while the TCP/IP connection is down.

User Action

Verify that the DD name on the TCP SYSIN card for message buffering has a matching DD card in the VP390 startup JCL. Verify that the dataset named for that DD name is defined with the DCB values stated in the *OV OS/390 Installation Guide*. Recycle the VP390 job if any changes are made to the SYSIN cards or the startup JCL.

EVO205 MVS console *name* reached memory limit. Data lost

Message Variables

name VP390 console name

Message Description

The extended console defined for VP390 filled all available cells in the data space. The incoming message is not queued.

System Action

Processing continues.

User Action

Check the status of the extended console with the DISPLAY CONSOLES,CN=*name*

command. If messages do not resume queuing to the extended console, recycle VP390, making sure the console shuts down without any problems. You may need to define a new console with a larger message data space.

EVO206 MVS console *name* reached queue limit, data lost

Message Variables

name VP390 console name

Message Description

The extended console defined for VP390 reached its maximum queue depth.

System Action

The incoming message is not queued. Processing continues.

User Action

Check the status of the extended console with the DISPLAY CONSOLES,CN=*name*

command. If messages do not resume queuing to the extended console,

recycle VP390, making sure the console shuts down without any problems. Use the QL parameter on the MVS `SYSIN` card to increase the queue size of the console. See the definition of the MVS Parameter Card in the *OV OS/390 Installation Guide.*

EVO207 MVS console *name* stopped by internal error

Message Variables

name VP390 console name

Message Description

The extended console defined for VP390 received an error while processing its message queues.

System Action

VP390 deactivates the console and stops the MVS subtask.

User Action

Recycle the subtask, then issue a `DISPLAY CONSOLES,CN=`*name* command to check the status of the *name* console.

EVO208 MVS console *name* reached alert percentage

Message Variables

name VP390 console name

Message Description

The number of messages queued to the extended console reached a pre-specified alert percentage of the maximum queue depth.

System Action

Processing continues.

User Action

Verify that desired MVS messages are being sent to the OVO server. Check the status of the extended console with the `DISPLAY CONSOLES,CN=`*name* command. If the queue shortage is not relieved shortly, recycle VP390, making sure the console shuts down without any problems. Use the QL parameter on the MVS `SYSIN` card to increase the queue size of the console. See the definition of the MVS Parameter Card in the *OV OS/390 Installation Guide.*

EVO209 MVS console *name* suspended by request

Message Variables

name VP390 console name

Message Description

A condition developed in the extended console defined for VP390 that caused the operating system to request console deactivation.

System Action

VP390 deactivates the console and stops the MVS subtask.

User Action

Recycle the subtask, then issue a `DISPLAY CONSOLES,CN=`*name* command to check the status of the *name* console.

EVO210 MVS console *name* alert ECB posted for unknown reason

Message Variables

name VP390 console name

Message Description

The extended console defined for VP390 is posted with an alert indicating a problem, but no error flags are set in the console status area.

System Action

Processing continues.

User Action

Check the condition of the console with the `DISPLAY CONSOLES,CN=`*name* command.

EVO211 DOM *source key*

Message Variables

source message deletion type, either “MSGKEY” or “TOKEN”

key identifying number of the original message

Message Description

The operating system has issued a Delete Operator Message notification that a previous message (identified by a MSGKEY) or group of messages (identified by a TOKEN) have been deleted from the console.

System Action

Processing continues.

User Action

If DOM processing is active in VP390 (activated by the "DOM" option on the MVS parameter card in `SYSIN`), then this message will be sent to the OVO server. It can be used for automatically acknowledging an existing message on the OVO browser. See "Using DOM Information" in Chapter 2 for more information.

`EVO214 DOM flag updated`

Message Variables

None.

Message Description

In response to a `MODIFY` command, VP390 has changed its processing of operating system DOM messages. See "About DOM Commands" in Appendix A.

System Action

Processing continues.

User Action

None.

`EVO215 PERF parameter updated`

Message Variables

None.

Message Description

In response to a `MODIFY` command, VP390 has updated its timing intervals for performance data gathering. See "About PERF Commands" in Appendix A of the *OV OS/390 Administrator's Reference*.

System Action

Processing continues.

User Action
None.

EVO302 *name* **: VP390 PPI TASK INITIALIZED**

Message Variables
name Name of NetView/390 PPI subtask

Message Description
The program-to-program interface subtask for VP390 is successfully initialized in the NetView/390 address space.

System Action
Processing continues.

User Action
None.

EVO303 *name* **: VP390 PPI TASK TERMINATED**

Message Variables
name Name of NetView/390 PPI subtask

Message Description
The program-to-program interface task for VP390 is terminated in the NetView/390 address space.

System Action
Processing continues, but VP390 no longer receives unsolicited VTAM messages and alerts from NetView/390.

User Action
Restart NetView/390 if it is terminated. If only the PPI subtask is terminated, restart the subtask from a NetView/390 operator session with the START TASK=*name* command.

EVO304 *name* **: DSIFRE FAILED FOR USER STORAGE**

Message Variables
name Name of NetView/390 PPI subtask

Message Description

The VP390 PPI program running in the NetView address space received an error return code from the NetView/390 macro `DSIFRE` while attempting to free the 4K work area of memory during subtask shutdown.

System Action

Subtask shutdown processing continues.

User Action

Notify the system programmer that a potential memory leak exists in the currently running NetView/390.

EVO305 *name* : DSIFRE FAILED FOR QUEUED STORAGE

Message Variables

name Name of NetView/390 `PPI` subtask

Message Description

The VP390 PPI program running in the NetView address space received an error return code from NetView/390 macro `DSIFRE` while attempting to free all remaining subtask memory during subtask shutdown.

System Action

Subtask shutdown processing continues.

User Action

Notify the system programmer that a potential memory leak exists in the currently running NetView/390.

EVO306 *name* : DSIFRE FAILED FOR MQS BUFFER

Message Variables

name Name of NetView/390 `PPI` subtask

Message Description

The VP390 PPI program running in the NetView address space received an error return code from NetView/390 macro `DSIFRE` while attempting to free the memory allocated for the private message queue.

System Action

Processing continues.

User Action

Notify the system programmer that a potential memory leak exists in the currently running NetView/390.

EVO307 *name* : DSIGET FAILED FOR USER STORAGE

Message Variables

name Name of NetView/390 PPI subtask

Message Description

The VP390 PPI program running in the NetView address space failed to get a 4K block of memory for use during processing.

System Action

Task termination flag is set.

User Action

Notify the system programmer that a potential memory shortage exists in the currently running NetView/390. The region size of the NetView/390 address space may need to be increased.

EVO308 *name* : ENQ ERROR

Message Variables

name Name of NetView/390 PPI subtask

Message Description

An ENQ on the NetView/390 TVB chain failed.

System Action

If not already in termination processing, the task termination flag is set.

User Action

Notify the system programmer. Restart the subtask.

EVO309 *name* : DEQ ERROR

Message Variables

name Name of NetView/390 PPI subtask

Message Description

A DEQ on the NetView/390 TVB chain failed.

System Action

If not already in termination processing, the task termination flag is set.

User Action

Notify the system programmer. Restart the subtask.

EVO310 *name* : TASK ALREADY EXISTS

Message Variables

name Name of NetView/390 PPI subtask

Message Description

The VP390 PPI subtask attempted to add itself to the NetView/390 TVB chain, but found another task with the same name already on the chain.

System Action

The task termination flag is set.

User Action

Verify that another instance of the subtask is not already running under this NetView/390. Restart the subtask.

EVO311 *name* : LOAD OF CNMNETV COMPLETE

Message Variables

name Name of NetView/390 PPI subtask

Message Description

The loading of the CNMNETV module into NetView virtual storage completed successfully.

System Action

Processing continues.

User Action

None.

EVO312 *name* : UNABLE TO LOAD CNMNETV

Message Variables

name Name of NetView/390 PPI subtask

Message Description

The loading of the CNMNETV module into virtual storage failed.

System Action

The subtask terminates.

User Action

Verify that load module CNMNETV exists in a NetView/390 STEPLIB dataset.

Restart the subtask.

EVO313 *name* **: NETVIEW INTERFACE FAILURE, RC=***rcnumber*

Message Variables

name Name of NetView/390 PPI subtask

rcnumber Hexadecimal return code from CNMNETV call

Message Description

A call to the CNMNETV interface routine failed.

System Action

The message is discarded.

User Action

For explanations of return codes, see the *TME 10 NetView for OS/390 Application Programmer's Guide*.

EVO314 *name* **: NETVIEW COMMAND RECEIVED**

Message Variables

name Name of NetView/390 PPI subtask

Message Description

A message was successfully received from the PPI interface routine.

This message is used for debugging purposes only. It is not displayed unless the subtask is reassembled with the CMDREC lines uncommented.

System Action

Processing continues.

User Action

None.

EVO315 *autotask* COMMAND EXECUTION FAILED

Message Variables

autotask Name of NetView/390 autotask that executes the command

Message Description

A failure occurred in a command that was to be executed under NetView/390 on behalf of VP390.

System Action

The command is discarded.

User Action

Verify that the autotask defined under NetView/390 during VP390 installation is active. Verify that the `NTICMD` and `NTIMVS` command lists are present in a NetView/390 `DSICLD` dataset. Verify that the name in the `EVOCMD_OPERATOR` field on the OVO server (which was filled in while running the Add Node program) matches the autotask name defined under NetView/390.

EVO595 Command entered: *cmdtxt*

Message Variables

cmdtxt Text of command entered

Message Description

VP390 received a command from a console.

System Action

Processing continues with the execution of the command.

User Action

None.

EVO600 TNUM TASKNAME STATUS RESTARTS/LIMIT SPECIFIC

Message Variables
None.
Message Description
This message is the header of a table which is generated in response to a SHOW TASK console command. Additional EVO600 messages will follow with data for each subtask.
System Action
Processing continues.
User Action
None.

EVO603 TNUM TASKNAME ADDRESS HEADER TCB

Message Variables
None.
Message Description
This message is the header of a table which is generated in response to a SHOW ADDR console command. Additional EVO603 messages will follow with data for each subtask.
System Action
Processing continues.
User Action
None.

EVO605 TNUM TASKNAME INPUTQ OUTPUTQ INFLOW OUTFLOW MC

Message Variables
None.
Message Description
This message is the header of a table which is generated in response to a SHOW FLOW console command. Additional EVO605 messages will follow with data for each subtask.
System Action
Processing continues.
User Action
None.

EVO608 TASKNAME DATE TIME

Message Variables
None.
Message Description
This message is the header of a table which is generated in response to a SHOW VERSION console command. Additional EVO608 messages will follow with data for each subtask.
System Action
Processing continues.
User Action
None.

EVO609 *type* **filters:**

Message Variables
type Filter type, either “Message” or “Alert”.
Message Description
This message is the start of a list of filter table entries which is generated in response to a SHOW FILTER console command. Additional EVO609 messages will follow with lists of filter table entries. Message IDs (include entries) will be listed four per line after the EVO609. CNM alerts (exclude entries) will be listed with the offset followed by the hexadecimal data.
System Action
Processing continues.
User Action
None.

EVO610 *type* **filter** *data action*

Message Variables
type Filter type, either “Message” or “Alert”.
data User-entered data

action Command action, either “added” or “deleted”

Message Description

Verification message to indicate that the message of filter table action entered from a VP390 `MODIFY` command has completed successfully.

System Action

Processing continues.

User Action

None.

EVO611 *type* filter *data* not found

Message Variables

type Filter type, either “Message”, “Alert”, “JOBNAME”, or “JOBID”

data User-entered data

Message Description

A VP390 `MODIFY` command could not find the *data* entry when attempting to delete it from the message or alert table.

System Action

Processing continues.

User Action

None.

EVO612 No *type* filters defined

Message Variables

type Filter type, either “message” or “alert”

Message Description

A VP390 `MODIFY` command could not any filters of the type *type* to display.

System Action

Processing continues.

User Action

None.

EVO613 All *type* filters deleted

Message Variables

type Filter type: “message and alert”

Message Description

A FILTER DEL ALL command has successfully deleted all message and alert filter table entries.

System Action

Processing continues.

User Action

None.

EVO614 No suppressed messages

Message Variables

None.

Message Description

The VP390 message suppression table has no entries to display as a result of a SHOW SUPPRESS command.

System Action

Processing continues.

User Action

None.

EVO615 Suppressed message IDs:

Message Variables

None.

Message Description

This message is the header of a table which is generated in response to a SHOW SUPPRESS console command. Additional EVO615 messages will follow with a list of VP390 message IDs, eight per line, that should not be sent to the console.

System Action

Processing continues.

User Action
None.

EVO616 *action* suppression of *msgid*

Message Variables
action Suppression action, either “Added” or “Removed”.

msgid VP390 message ID.

Message Description
Verification message to indicate that the action to suppress or unsuppress a VP390 message ID from printing on the system console has completed successfully.

System Action
Processing continues.

User Action
None.

EVO617 Message ID *msgid* not found in suppression table

Message Variables
msgid VP390 message ID.

Message Description
An attempt to UNSUPPRESS a message ID in the VP390 message suppression table failed.. The message ID given was not found in the table.

System Action
Processing continues.

User Action
Use the SHOW SUPPRESS command to see the list of message IDs currently in the table. Use only the 3-digit suffix of the message ID when issuing an UNSUPPRESS command.

EVO690 VP390 STOP Command accepted

Message Variables

None.

Message Description

The VP390 task has received a STOP command.

System Action

Processing continues with shutdown of any active subtasks, then ends the main task.

User Action

None.

EVO695 VP390 *cmdtype* command processed

Message Variables

cmdtype Command type

Message Description

VP390 completed the initial processing of a console command. Additional messages may be sent, depending on whether additional work is being done by subtasks.

System Action

None.

User Action

None.

EVO698 Subtask *task* is already *status*

Message Variables

task Subtask name

status Current subtask status, either “active” or “inactive”

Message Description

A request to activate or deactivate a VP390 subtask was not processed because the subtask is already in that state.

System Action

None.

User Action

Use the SHOW TASK command to verify the status of the VP390 subtasks.

EVO699 Invalid operator command entered

Message Variables
None.

Message Description
An invalid MODIFY command was sent to the VP390 task.

System Action
None.

User Action
See Appendix A of the *OV OS/390 Administrator's Reference* for syntax rules of MODIFY commands.

EVO701 Starting subtask *#idnum* for *info*

Message Variables
idnum Numerical ID for the newly started subtask

info Information sent to the ATTACH macro

Message Description
VP390 attached a subtask with the information provided in *info*.

System Action
Processing continues with the ATTACH attempt.

User Action
None.

EVO702 Buffer size = *size*M, Queue depth = *totalmsg*, Maximum = *maxmsg*

Message Variables
size Size (in megabytes) allocated for messages

totalmsg Total message queue depth

maxmsg Maximum message queue depth permitted

Message Description
A message queuing problem occurred for an MCS console defined for VP390. This message should be displayed only in the VP390 job log.

Additional message(s) giving more detailed information about the problem may appear on the system console at the same time.

System Action

Processing continues. The MCS console may be terminated, depending on the severity of the queuing problem.

User Action

Monitor the VP390 job log and system console for the next message and necessary action..

EVO703 Console *name* is utilizing *pct*% of message queue

Message Variables

name Name of defined extended console

pct Percentage of console queue in use

Message Description

This message is generated when the extended console for gathering MVS messages has a backlog of messages on its queue to be processed by the VP390 task. *pct* tells what percentage of the console's queue is in use. This message is only generated when using the QLP option of the MVS SYSIN card.

System Action

Processing continues.

User Action

The extended console *name* may need to be re-defined with a larger queue size. See the QL and QLP options of the MVS parameter card in the *OV OS/390 Installation Guide*.

EVO704 Console *name* queue backlog has been relieved

Message Variables

name Name of defined extended console

Message Description

This message is generated after an EVO703 message is issued to announce that the console message queue shortage has been relieved. This message is only generated when using the QLP option of the

MVS SYSIN card.

System Action

Processing continues.

User Action

The extended console *name* may need to be re-defined with a larger queue size. See the QL and QLP options of the MVS parameter card in the *OV OS/390 Installation Guide*. This message can be used for automatically acknowledging an existing EVO703 message on the OVO browser.

EVO778 RMF data not available, rc=*code*

Message Variables

code Return code

Message Description

VP390 encountered an error while attempting to collect system data from the mainframe Resource Measurement Facility (RMF) for a vp390hostcmd 46 option 02 call.

System Action

The OSINFO subtask will send an EVO131 error message in response to the vp390hostcmd explaining that the command had failed to complete.

User Action

The meaning of the return code *code* can be looked up in Chapter 1 of the IBM *Resource Measurement Facility Programmer's Guide* under the section of "Return Codes" for the ERBSMFI command.

EVO901 Stopping subtask #*number*: *name*

Message Variables

number Subtask number

name Subtask name

Message Description

This message is issued in response to a STOP command. One message is issued for each VP390 subtask.

System Action
A termination command is sent to each of the existing subtasks.
User Action
None.

EVO902 *name* **subtask terminated, RC =** *rcnumber*

Message Variables
name Name of subtask
rcnumber Return code from termination call
Message Description
The named subtask is terminated.
System Action
Any queues or memory allocated for the subtask are freed.
User Action
None.

EVO903 *name type* **queue freed, RC =** *rcnumber*

Message Variables
name Name of subtask
type Queue type, either “Input” or “Output”
rcnumber Return code from `Free` call
Message Description
An allocated message queue for the named subtask has been cleared during subtask termination.
System Action
Processing continues.
User Action
None.

EVO904 All VP390 subtasks completed

Message Variables
None.

Message Description
VP390 completed the shutdown of all subtasks.
System Action
Processing continues with main task shutdown.
User Action
None.

EVO905 Restart #*num* of subtask *name* will be attempted in *sec* seconds

Message Variables
num Count of number of restarts for this subtask
name Name of subtask
sec Number of seconds until next automatic restart attempt
Message Description
The subtask *name* has been terminated, but will be automatically restarted in *sec* seconds.
System Action
Processing continues.
User Action
None.

EVO906 No auto restart for *name* - Use INIT command to restart

Message Variables
name Name of subtask
Message Description
The subtask *name* has terminated and will not restart because it has exceeded the number of automatic restarts allowed.
System Action
Processing continues.
User Action
Use the console INIT command to restart the subtask. See Appendix A of the *OV OS/390 Administrator's Reference* for the syntax of the INIT command. Use the console command SHOW TASK to see how many restarts are allowed for each subtask. To change the number of

automatic restarts that a subtask is allowed, add a RESTART card to the `SYSIN` deck just prior to the *`name`* subtask parameter card. See the “RESTART Parameter Card” in the *OV OS/390 Installation Guide* for the syntax of the RESTART card.

C Resource Monitoring Classes

These are the data source classes for OV OS/390 performance and DASD data collection. Use these lists as a reference to identify what metrics are collected.

RMF Performance Data Class

```
ASS OS390_HOSTNAME=10040
LABEL "HOSTNAME RMF Stats"
INDEX BY DAY
MAX INDEXES 62
ROLL BY MONTH;

TRICS
VP390_HOSTNAME=101
LABEL "Mainframe Hostname"
TYPE TEXT LENGTH 256;

CPU_UTIL = 102
LABEL "Interval CPU Util"
PRECISION 0;

PAGING_RATE = 103
LABEL "Interval Paging Rate"
PRECISION 0;

SYS_COMMON_PAGES_IN = 104
LABEL "Sys Common pages in"
PRECISION 0;

SWAP_OUTS = 105
LABEL "Number of swaps out"
PRECISION 0;

PAGES_SWAPPED_IN = 106
LABEL "Pages swapped in"
PRECISION 0;

PAGES_SWAPPED_OUT = 107
LABEL "Pages swapped out"
PRECISION 0;

PRIVATE_PAGES_IN = 108
LABEL "Private pages in"
PRECISION 0;

PRIVATE_PAGES_OUT = 109
LABEL "Private pages out"
PRECISION 0;
```

```
GH_UIC_COUNT = 110
LABEL "High UIC Count"
PRECISION 0;

SYS_LPA_PAGES_IN = 111
LABEL "Sys LPA Pages In"
PRECISION 0;

PAGES_TO_EXT = 112
LABEL "Pages to ext stor"
PRECISION 0;

EXT_SLOTS_AVAIL = 113
LABEL "Extended slots avail"
PRECISION 0;

PAGES_MIG_AUX = 114
LABEL "Pages migrated aux"
PRECISION 0;

AVAIL_FRAME_COUNT = 115
LABEL "Avail Frame Count"
PRECISION 0;

DEV_ACTIVITY_RATE = 116
LABEL "Device Activity Rate"
PRECISION 0;

DEV_RESP_TIME = 117
LABEL "Device Resp Time"
PRECISION 0;

DEV_IOSQ_TIME = 118
LABEL "Device queue time"
PRECISION 0;

FIXED_SQA_FRAMES = 119
LABEL "Fixed SQA Frames"
PRECISION 0;
```

```
MMON_FIXED_FRAMES = 120
LABEL "Common Fixed Frames"
PRECISION 0;

PRIVATE_FIXED_FRAMES = 121
LABEL "Private Fixed Frames"
PRECISION 0;

AS_IN_STORAGE =  122
LABEL "AS in Storage"
PRECISION 0;

LPA_FRAMES = 123
LABEL "LPA Frames"
PRECISION 0;

CSA_FRAMES = 124
LABEL "CSA Frames"
PRECISION 0;

LPA_FIXED_FRAMES = 125
LABEL "LPA Fixed Frames"
PRECISION 0;


CSA_FIXED_FRAMES = 126
LABEL "CSA Fixed Frames"
PRECISION 0;

LSQA_FIXED_FRAMES = 127
LABEL "LSQA Fixed Frames"
PRECISION 0;

AS_LOG_OUT = 128
LABEL "AS Logically Out"
PRECISION 0;
```

DASD Statistics Data Class

```
ASS DASDSTAT_HOSTNAME=10040
LABEL "HOSTNAME DASD Stats"
INDEX BY DAY
MAX INDEXES 62
ROLL BY MONTH;

TRICS
VP390_HOSTNAME=101
LABEL "Mainframe Hostname"
TYPE TEXT LENGTH 256;

VOLSER = 102
LABEL "Volume Serial"
TYPE TEXT LENGTH 6;

TOTAL_TRACKS = 103
LABEL "Total Tracks"
PRECISION 0;

TRACKS_PERL_CYL = 104
LABEL "Tracks/cylinder"
PRECISION 0;

FREE_EXTENTS = 105
LABEL "Free Extents"
PRECISION 0;

FREE_TRACKS = 106
LABEL "Free Tracks"
PRECISION 0;

LARGEST_FREE_EXTENT = 107
LABEL "Largest Free Extent"
PRECISION 0;

DASD_UTILIZATION = 108
LABEL "DASD Utilization"
PRECISION 0;

FREE_DSCBS = 109
LABEL "Free DSCBs"
PRECISION 0;
```

DASD Summary Statistics Class

```
ASS DASDSUMM_HOSTNAME=10040
LABEL "HOSTNAME DASD Summary"
INDEX BY DAY
MAX INDEXES 62
ROLL BY MONTH;

TRICS
VP390_HOSTNAME=101
LABEL "Mainframe Hostname"
TYPE TEXT LENGTH 256;

AVG_FREE_EXTENTS = 102
LABEL "Average Free Extents"
PRECISION 0;

AVG_FREE_TRACKS = 103
LABEL "Average Free Tracks"
PRECISION 0;

LARGREST_FREE_EXTENT = 104
LABEL "Largest Free Extent"
PRECISION 0;

AVERAGE_UTILIZATION = 105
LABEL "Average Utilization"
PRECISION 0;

AVERAGE_FREE_DSCBS = 106
LABEL "Average Free DSCBs"
PRECISION 0;
```

A

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.

C

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.*

D

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.

H

high-level qualifier *See HLQ.*

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

HP OpenView Windows *See OVW.*

I

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 work day or after a system malfunction. 3. Process of loading system programs and preparing a system to run jobs.

J

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.*

L

Legacy Link Interface *See LLI.*

LLI Legacy Link Interface. VPO option that allows external processes to connect to VPO 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.

M

MCS Multiple Console Support. Method of programatically defining an OS/390 console for command and message support.

Message Queuing Series *See Message Queuing Series.*

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. *See also* IEBCOPY.

N

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.

Program to Program Interface

See NTIPPI.

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 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 Program to Progam Interface. VP390 message and command interface to NetView/390. *See also* NetView/390*;* PPI.

O

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.

P

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.

R

RACF Resource Access Control Facility. Licensed IBM program providing user and resource authorization security. *See also* ACF2.

Resource Access Control Facility

See RACF.

S

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. 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.

T

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 supports 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.

V

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.