EView/400i IBM i (iSeries-AS/400) Management for Micro Focus Operations Manager i (OMi)

Administrator’s Reference

Software Version: 7. 0

April 2018

Legal Notices

Warranty

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

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

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

www.eview-tech.com

Copyright Notices

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.

AS/400, OS/400, and iSeries, IBM i are trademarks of International Business Machines Corporation.

NetView/390 is a product of IBM Corporation.

OpenView®, Operations Manger (OM), Operations Manager i (OMi) are registered trademarks of Micro Focus.

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

UNIX® is a registered trademark of the Open Group.

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

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

Printing History

The printing date of the manual indicates the edition of the manual; the printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date, and manual updates may be issued between editions to correct errors or document product changes. The printing date will be updated if extensive changes are made to the manual or upon the release of new versions of the software.

First Edition:	Version 6.3	December 2015
Second Edition:	Version 7.0	February 2018

Support

Visit the EView Technology web site at:

`http://www.eview-tech.com/`

This web site provides contact information and details about the products, services, and support that EView Technology offers.

You can also contact EView Technology support via e-mail. See the web site for contact information.

Table of Contents

1

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 *EView/400i 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 the [Apply] button.
Menu Items	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows.	Select Actions:Utilities->Reports ...

2

Documentation

EView/400i IBM i (iSeries-AS/400) Management for OMi (EView/400i) provides a set of manuals that help you use the product and understand the concepts underlying the product. This section describes what information is available and where you can find it.

In addition to EView/400i documentation, related Micro Focus products provide a comprehensive set of manuals that help you use the products and improve your understanding of the underlying concepts.

EView/400i Printed Manuals

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

EView/400i Management for Micro Focus OMi Concepts Guide

Explains EView/400i features, functions, architecture, and data flow. Describes EView/400i agent and server components, process management, and message windows.

EView/400i Management for Micro Focus OMi Installation Guide

Explains how to install, de-install, and configure EView/400i. Also includes how to upload installation files to the EView proxy server, and start and stop EView/400i.

EView/400i Management for Micro Focus OMi Administrator's Reference

Explains how to customize and use EView/400i. Also includes detailed troubleshooting procedures and explanations of EView/400i system messages.

EView/400i Online Information

The following information is available online.

- *EView/400i Management for Micro Focus OMi Concepts Guide*
- *EView/400i Management for Micro Focus OMi Installation Guide*
- *EView/400i Management for Micro Focus OMi Administrator's Reference*

3

Configuring EView/400i

This chapter describes how to configure EView/400i and distribute the configurations to the iSeries (AS/400) agents.

Phase 1: Add, Modify, and Distribute Agent Parameters and Filters

In this section, iSeries nodes are defined to the OMi component of EView/400i, and new nodes are added to the OMi Node configuration. Agent parameters are distributed to the iSeries agent(s) using the EView/400i Configuration Interface.

Starting the iSeries Agent Configuration Interface

The iSeries node configuration interface provides a launching pad for all EView/400i configuration utility programs (See Figure 3-1).

The EView/400i configuration server is configured to start automatically after installation and after booting the EView proxy server. If the configuration server is not running, start it as follows:

On Linux servers run the following command to start the configuration server:

```
/etc/init.d/vp400conf start
```

On Windows servers, the configuration server runs as a service. To start the configuration service, open the Windows server manager and locate the EView 400 configuration service and right click on the service name and choose Start.

Start the “EView/400i Configurator” application from a web browser pointing to the EView Proxy server where the EView/400i is installed with the following URL:

http://*servername*:9850

By default, the EView/400i Configurator listens on port 9850 on the installed server. This port number may be customized by changing the EV400_CONFIG_PORT value in the `/etc/opt/OV/share/conf/vp400/vp400info` file on the server on Linux and on Windows in the `vp400info` file in the `conf` subdirectory of the EView/400 installation path.

Access to the configuration interface may be restricted to specific IP addresses or subnets by adding the EV400_CONFIG_ALLOWED_IP_NETWORK value to this `vp400info` file, for example:

To allow a specific IP address to connect:

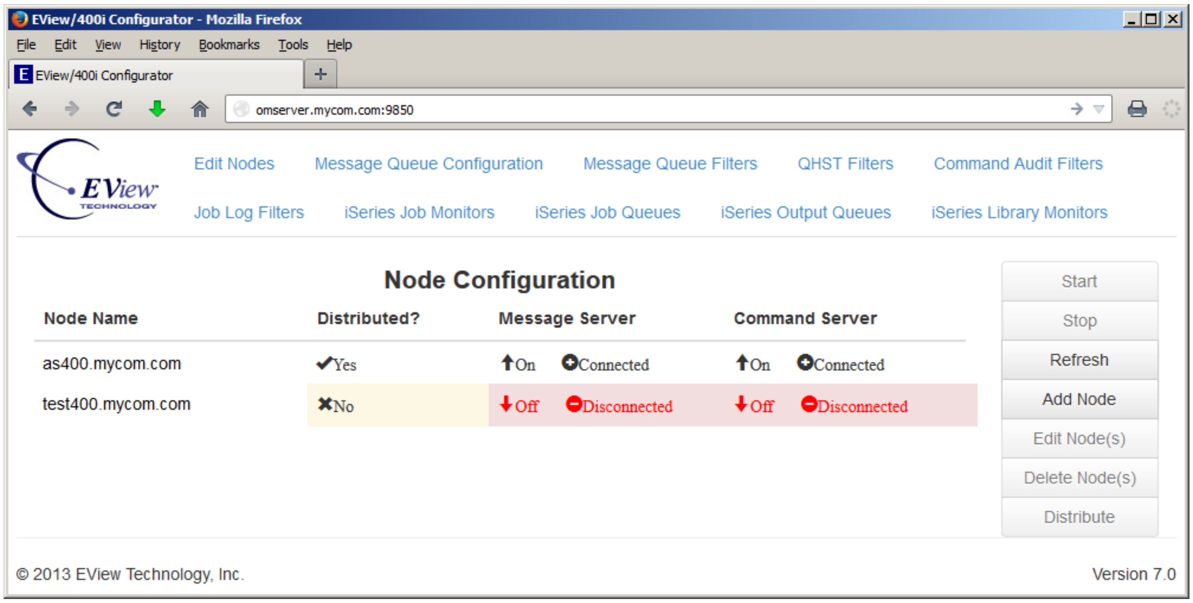
```
EV400_CONFIG_ALLOWED_IP_NETWORK  192.168.0.222
```

To allow a range of IP subnets to connect (using either CIDR prefix or dotted decimal subnet mask):

```
EV400_CONFIG_ALLOWED_IP_NETWORK  10.5.0.0/16
EV400_CONFIG_ALLOWED_IP_NETWORK  192.168.7.0/255.255.255.0
```

Multiple EV400_CONFIG_ALLOWED_IP_NETWORK entries are allowed in the `vp400info` file.

Figure 3-1: EView/400i Configuration Launchpad



Node Configuration

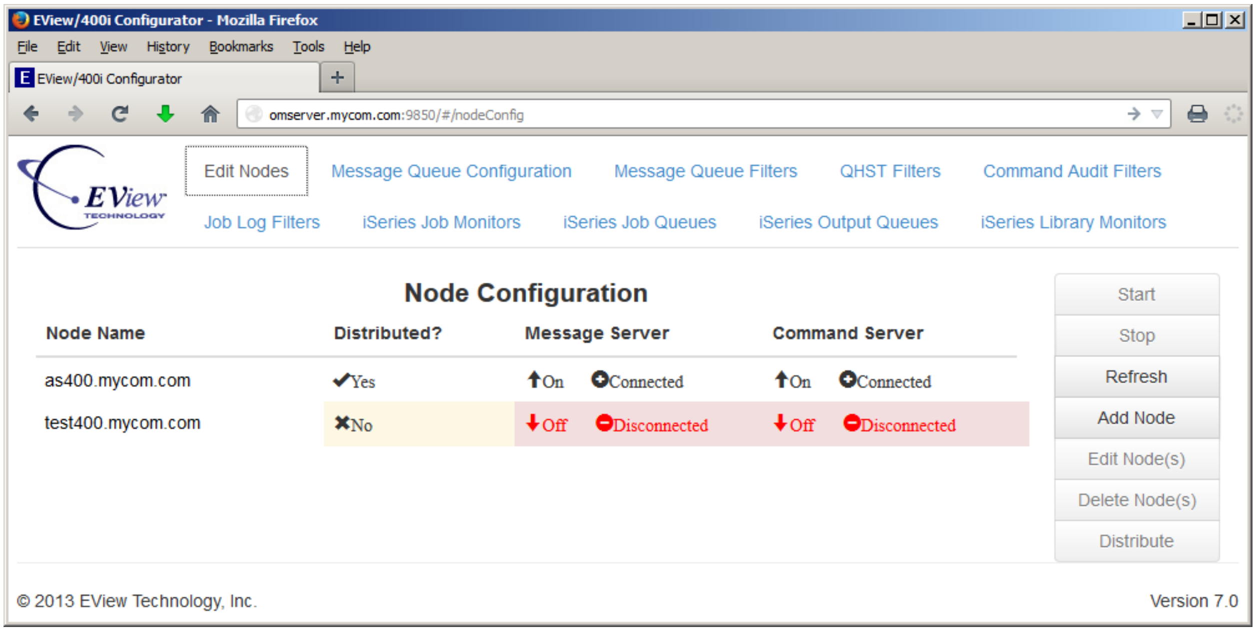
Node Name	Distributed?	Message Server		Command Server	
as400.mycom.com	✓Yes	↑On	⊕Connected	↑On	⊕Connected
test400.mycom.com	✖No	↓Off	⊖Disconnected	↓Off	⊖Disconnected

Add iSeries Nodes and Runtime Parameters

The Node Configuration utility will define an iSeries node to be monitored by EView/400i.

1. Start the Node Configuration application (Figure 3-2) by clicking the [Edit Nodes] link on the Configurator page.

Figure 3-2: EView/400i Node Configuration Utility



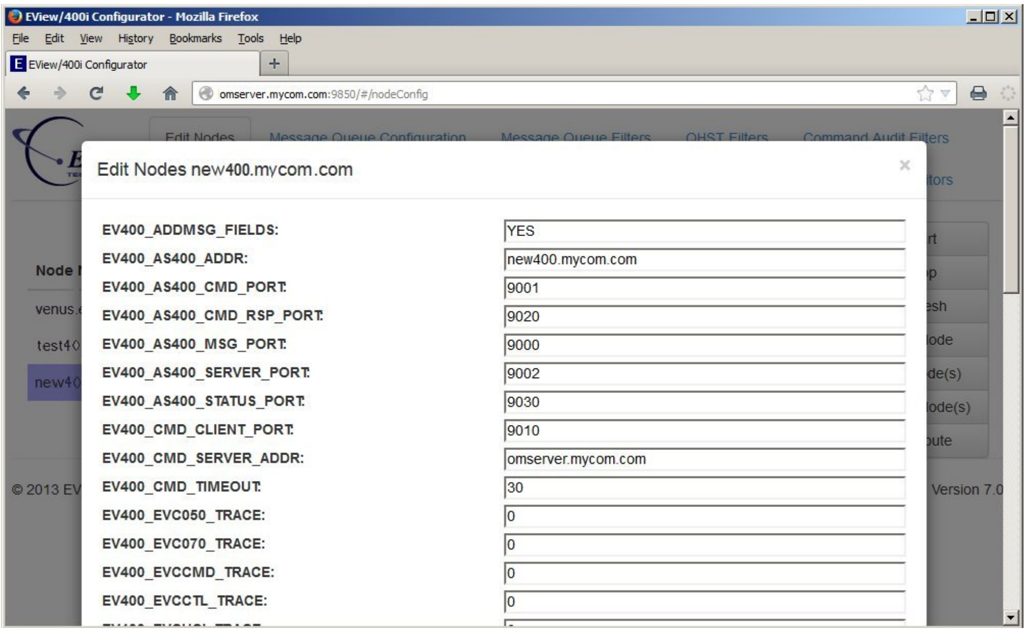
Node Configuration

Node Name	Distributed?	Message Server		Command Server	
as400.mycom.com	✓Yes	↑On	⊕Connected	↑On	⊕Connected
test400.mycom.com	✖No	↓Off	⊖Disconnected	↓Off	⊖Disconnected

2. Select [Add Node] and enter the fully qualified name of the iSeries node to be monitored. The name must be able to be resolved through the existing name service (for example, DNS or an entry in the OM server's `/etc/hosts` file).

3. Select the new node name in the list and click [Edit Node] to modify any of the configuration parameters.

Figure 3-3: EView/400i Node Configuration Parameters



Edit Nodes new400.mycom.com

EV400_ADDMSG_FIELDS:	YES
EV400_AS400_ADDR:	new400.mycom.com
EV400_AS400_CMD_PORT:	9001
EV400_AS400_CMD_RSP_PORT:	9020
EV400_AS400_MSG_PORT:	9000
EV400_AS400_SERVER_PORT:	9002
EV400_AS400_STATUS_PORT:	9030
EV400_CMD_CLIENT_PORT:	9010
EV400_CMD_SERVER_ADDR:	omserver.mycom.com
EV400_CMD_TIMEOUT:	30
EV400_EVC050_TRACE:	0
EV400_EVC070_TRACE:	0
EV400_EVCCMD_TRACE:	0
EV400_EVCCTL_TRACE:	0

The standard configuration parameters are:

EV400_AS400_ADDMSG_FIELDS

Indicates whether EView/400i will send the Program Name and Message Type fields in the messages that are sent to the EView Proxy server.
Default Value
YES
Valid Values
YES – The EView/400i message server will send the Program Name and Message Type fields in its messages sent to the EView Proxy Server. These fields are new in the EView/400i releases and will need to be accommodated in any existing template conditions that were written for the OV OS/400 product.
NO – Use this option if you are using message template conditions from the OV OS/400 and do not wish to modify those existing templates to utilize the new fields.

EV400_AS400_ADDR

Fully qualified network name of the iSeries system where the EView agent component is installed.

Default Value
None
Valid Values
Name of the managed node

EV400_AS400_BIND_ADDR

Address on the agent that the EVSBS subsystem should bind to when opening its TCP/IP listening ports (useful when the AS/400 has multiple IP addresses defined).
Default Value
0.0.0.0 – the INADDR_ANY default
Valid Values
IPV4 dotted decimal address in the format *nnn.nnn.nnn.nnn*. The value must be a defined address on the AS/400 system.

EV400_AS400_CMD_PORT

TCP/IP port number assigned to the EView/400i Command Server process.
Default Value
9001
Valid Values
Any unused port number on the iSeries agent between 1024 and 49151.

EV400_AS400_MSG_PORT

TCP/IP port number assigned to the EView/400i Master Message Server process.
Default Value
9000
Valid Values
Any unused port number on the iSeries agent between 1024 and 49151.

EV400_AS400_SERV_ADDR

Address (or address range) of the EView Proxy server(s) that are allowed to connect to this AS/400 agent. Use a "/" followed by a CIDR prefix or subnet mask to specify a range of allowed addresses.
Default Value
0.0.0.0 – Any address may connect to the listening EView/400i ports.
Valid Values
IPV4 dotted decimal address in the format *nnn.nnn.nnn.nnn*, optionally followed by a slash and either a dotted decimal address representation of a subnet mask , or a number (0-32) representing the number of bit positions to use for the mask.

EV400_AS400_SERVER_PORT

A TCP port number reserved for inter-process communications on the AS/400 agent.
Default Value
9002
Valid Values

Any unused port number on the iSeries agent between 1024 and 49151.

EV400_CMD_CLIENT_PORT

A TCP port number used by the Command Server process to communicate with the Master Message Server process. This port number must be unique on the EView Proxy server where the Command Server and Master Message Server processes are running.
Default Value
8003. This number will be incremented automatically when new nodes are added so that the port numbers remain unique.
Valid Values
Any unused port number on the EView Proxy server between 1025 and 65535.

EV400_CMD_SERVER_ADDR

The name of the UNIX server where the Command Server process is to run.
Default Value
The EView Proxy server name.
Valid Values
A UNIX server name.

EV400_CMD_TIMEOUT

The amount of time to wait for an iSeries command response (in seconds).
Default Value
30
Valid Values
An integer greater than or equal to 1 (second).

EV400_CMD_USER

The user profile that will be used for executing command requests initiated by actions run on the EView proxy server. If the default user is not used, the user profile specified must be created. It is important that the user profile created have sufficient authority to execute required commands. For example, to allow operators to change status of jobs or job queues, this profile needs a special authority of *JOBCTL. This parameter should be used in environments where security requirements require that command execution authority be controlled at the iSeries level.
Default Value
EVUSER
Valid Values
An existing iSeries user profile.

EV400_LICKEY

The license key for this iSeries managed node. See the section "Obtaining License Keys" in the *EView/400i Installation Guide* for information.
Default Value

None

Valid Values

48-character string

EV400_MONITOR_QHST

Indicates whether the EView/400i agent should monitor for messages that are sent to the AS/400 QHST system history log. If set to "YES", then verify that the EV400_QHST_MON_FREQ field is greater than 0.

Default Value

YES

Valid Values

YES or NO

EV400_MON_AUDJRNL

A list of two-character entry types from the QAUDJRN that should be forwarded from the iSeries agent. Entry types may be separated by commas or spaces. See iSeries documentation (such as the *iSeries Security Reference*) for descriptions of journal entry types. See Appendix E for sample messages that will be sent to the OMi browser for each of these entry types.

Default Value

NONE

Valid Values

AD AF AU CA CD CO CP DO DS NA OW PA PG PW ST SV VA VP VU ZC ZR
ALL – All of the above
NONE – None of the above

EV400_MON_RESOURCES

Indicates whether the EView/400i agent should monitor the status of iSeries resources (lines, controllers, and devices). If set to "YES", then verify that the EV400_MON_RSRCE_FREQ field is greater than 0.

Default Value

YES

Valid Values

YES or NO

EV400_MON_RESOURCES_FREQ

Frequency (in seconds) that EView/400i agent polls the status of the iSeries agent's resources (lines, controllers, and devices). Only necessary when the EV400_MON_RESOURCES parameter is "YES".

Default Value

30

Valid Values

An integer greater than or equal to 1 (second)

EV400_MSG_DISTRIB

Should the iSeries agent send its collected messages to all EView Proxy servers that are in contact with it? (If “NO”, then specify in the EV400_PRIMARY_SERVER field which EView Proxy server is the primary recipient of messages.)
Default Value
YES
Valid Values
YES - Send unsolicited iSeries messages to all EView/400i servers that are in contact with this agent.
NO - Send unsolicited messages only to the primary server.

EV400_MSG_SERVER_ADDR

The name of the UNIX server where the Master Message Server process is to run.
Default Value
The EView Proxy server
Valid Values
A UNIX server name

EV400_PATH

The EView/400i installation directory on the EView Proxy server.
Default Value
`/opt/OV/vp400`
Valid Values
EView/400i home directory

EV400_PERF1

Specifies whether the performance gathering function will be activated on the iSeries agent to gather the data for performance group 1. See Appendix D for the list of metrics collected in group 1.
Default Value
NO
Valid Values
YES – Activate the performance gathering function on the iSeries agent. See "Phase 8: Configuring Nodes for Performance Data Collection" on page 38 for information on how to receive the performance data on the EView Proxy server and send it to OVPA or CODA.
NO – Do not activate performance data gathering for group 1.

EV400_PERF1_CLASS

Specifies an alternate name for the group 1 performance class in CODA or OVPA. The default will send performance data to class “OS400_PERF1_*nodename*” where *nodename* is the short name of the iSeries node. If this short name results in a class name that is longer than 20 characters, specify a new class name here and when defining the performance class to CODA/OVPA (see "Phase 8: Configuring Nodes for Performance Data Collection" on page 38).
Default Value
*DEFAULT
Valid Values

A valid OVPA/CODA class name less than 20 characters.

EV400_PERF1_INT

The interval, in minutes, at which group 1 performance data is collected on the iSeries agent and forwarded to the EView Proxy server. This field is only needed if EV400_PERF1 is set to "YES".
Default Value
5
Valid Values
An integer greater than or equal to 1 (minute).

EV400_PERF2

Specifies whether the performance gathering function will be activated on the AS/400 agent to gather the data for performance group 2. See Appendix D for the list of metrics collected in group 2.
Default Value
NO
Valid Values
YES – Activate the performance gathering function on the iSeries agent. See "Phase 11" on page 38 for information on how to receive the performance data on the EView Proxy server and send it to OVPA or CODA.
NO – Do not activate performance data gathering for group 2.

EV400_PERF2_CLASS

Specifies an alternate name for the group 2 performance class in CODA or OVPA. The default will send performance data to class "OS400_PERF2_*nodename*" where *nodename* is the short name of the iSeries node. If this short name results in a class name that is longer than 20 characters, specify a new class name here and when defining the performance class to CODA/OVPA (see "Phase 8: Configuring Nodes for Performance Data Collection" on page 38).
Default Value
*DEFAULT
Valid Values
A valid OVPA/CODA class name less than 20 characters.

EV400_PERF2_INT

The interval, in minutes, at which group 2 performance data is collected on the iSeries agent and forwarded to the EView Proxy server. This field is only needed if EV400_PERF2 is set to "YES".
Default Value
30
Valid Values
An integer greater than or equal to 1 (minutes).

EV400_PRIMARY_SERVER

The fully qualified name of the primary EView Proxy server to receive messages from this agent. Although multiple EView Proxy servers may be connected to the iSeries agent at the same time, only the server named here will receive unsolicited OS/400 messages. This field is only necessary when the EV400_MSG_DISTRIB parameter is “NO”.
Default Value
null
Valid Values
An EView/400i UNIX server name

EV400_QHST_MON_FREQ

Frequency, in seconds, that the EView/400i agent collects new messages from the QHST system history log. This field is only necessary when the EV400_MONITOR_QHST parameter is “YES”.
Default Value
30
Valid Values
An integer greater than or equal to 1 (second)

EV400_WORK_AREA

Specifies where EView/400i places temporary work files on the EView Proxy server.
Default Value
`/var/opt/OV/share/tmp/vp400`
Valid Values
Any existing directory on the EView Proxy server

EV400_VP400MMS_TRACE

Set tracing level for the master message server (vp400mms)
Default Value
0 - No tracing output enabled
Valid Values
The values below may be added together in hexadecimal to combine multiple tracing options:
0001 - general program tracing enabled
0002 - internal tracing enabled
0004 - program detail tracing enabled
0008 - warning messages enabled
0010 - error tracing enabled
0020 - dump output enabled
0040 - loop tracing enabled
0080 - verify tracing enabled
0100 - log messages sent to OM
0200 - log performance records

EV400_VP400CS_TRACE

Set tracing level for the command server (vp400cs)
Default Value
0 - No tracing output enabled
Valid Values
The values below may be added together in hexadecimal to combine multiple tracing options:
0001 - general program tracing enabled
0002 - internal tracing enabled
0004 - program detail tracing enabled
0008 - warning messages enabled
0010 - error tracing enabled
0020 - dump output enabled
0040 - loop tracing enabled
0080 - verify tracing enabled

EV400_VP400SM_TRACE

Set tracing level for the status manager (vp400sm)
Default Value
0 - No tracing output enabled
Valid Values
The values below may be added together in hexadecimal to combine multiple tracing options:
0001 - general program tracing enabled
0002 - internal tracing enabled
0004 - program detail tracing enabled
0008 - warning messages enabled
0010 - error tracing enabled
0020 - dump output enabled
0040 - loop tracing enabled
0080 - verify tracing enabled

EV400_VP400HOSTCMD_TRACE

Set tracing level for the host command client (vp400hostcmd)
Default Value
0 - No tracing output enabled
Valid Values
The values below may be added together in hexadecimal to combine multiple tracing options:
0001 - general program tracing enabled
0002 - internal tracing enabled
0004 - program detail tracing enabled
0008 - warning messages enabled
0010 - error tracing enabled
0020 - dump output enabled
0040 - loop tracing enabled
0080 - verify tracing enabled

EV400_EVCMSG_TRACE

Set tracing level for the agent message TCP task (EVCMSG)
Default Value
0 - Tracing disabled
Valid Values
The values below may be added together in hexadecimal to combine multiple tracing options:
0001 - general program tracing enabled
0002 - internal tracing enabled
0004 - program detail tracing enabled
0008 - warning messages enabled
0010 - error tracing enabled
0020 - dump output enabled
0040 - loop tracing enabled
0080 - verify tracing enabled

EV400_EVCHCI_TRACE

Set tracing level for the agent message transfer process (EVCHCI)
Default Value
0 - Tracing disabled
Valid Values
The values below may be added together in hexadecimal to combine multiple tracing options:
0001 - general program tracing enabled
0002 - internal tracing enabled
0004 - program detail tracing enabled
0008 - warning messages enabled
0010 - error tracing enabled
0020 - dump output enabled
0040 - loop tracing enabled
0080 - verify tracing enabled

EV400_EVC050_TRACE

Set tracing level for the agent command processor (EVC050)
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_EVCQSCAN_TRACE

Set tracing level for the agent message queue monitor (EVCQSCAN)
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_EVCCTL_TRACE

Set tracing level for the API interface process (EVCCTL)
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_EVPERFM_TRACE

Set tracing level for the agent performance monitor process (EVPERFM)
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_EVC070_TRACE

Set tracing level for the agent resource monitor (EVC070)
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_EVCCMD_TRACE

Set tracing level for the agent command TCP process (EVCCMD)
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_EVHSTPGM_TRACE

Set tracing level for the agent history log (QHST) monitor
Default Value
0 - Tracing disabled
Valid Values
0 - Tracing disabled
1 - Tracing enabled

EV400_VP400MMS_LOGSIZE

Set the maximum log size, in kilobytes, for the master message server (vp400mms)
Default Value
3000
Valid Values

1-99999

EV400_VP400CS_LOGSIZE

Set the maximum log size, in kilobytes, for the command server (vp400cs)
Default Value
3000
Valid Values
1-99999

EV400_VP400SM_LOGSIZE

Set the maximum log size, in kilobytes, for the status manager (vp400sm)
Default Value
3000
Valid Values
1-99999

4. Save the parameters for this agent. The Node Configuration program will save the parameters locally on the EView Proxy server.

5. Multiple nodes may be edited at once to edit the same parameters that are not required to be unique. Select the node names while holding down the **Ctrl** key then click the [Edit Node(s)] button.

6. Select any nodes in the list of defined nodes that have the "Distributed?" box marked with a "No" and click the [Distribute...] button to send the configuration parameters to the AS/400 agents. Multiple lines may be selected at one time. (Any time a node's configuration parameters are changed using the [Add Node] or [Edit Node] functions of the Node Configuration program, the "Distributed?" field will be marked with a "No" to remind you that there are changes that need to be distributed to the iSeries agent.)

7. If a node is deleted using the [Delete Node] button, the node will be removed from the EView/400i list of defined nodes, but it will remain in the OMi Node list. Use the OMi GUI to remove deleted nodes from OMi.

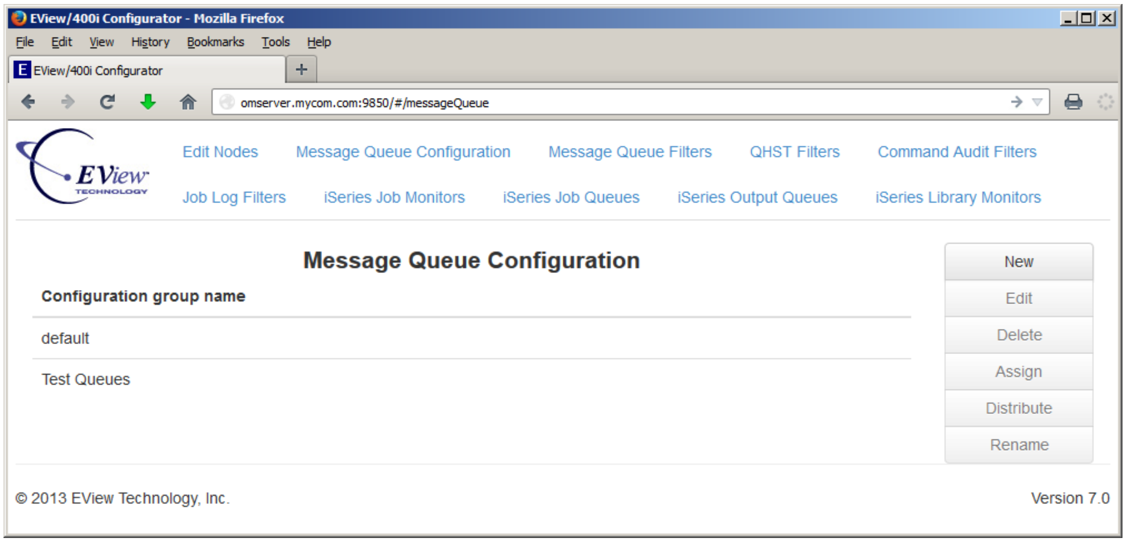
Phase 2: Add, Modify, and Distribute Message Queues and Message IDs

iSeries messages can be captured from any message queue or the QHST message log. This section explains how to identify which queues are to be monitored and which messages should be captured and passed from the EView/400i agent to the server.

Configure Message Queues

1. Start the Message Queue Configuration utility from the EView/400i Configurator (Figure 3-1) by clicking on the "Message Queue Configuration" button.

Figure 3-4: Message Queue Configurator



Message Queue Configuration

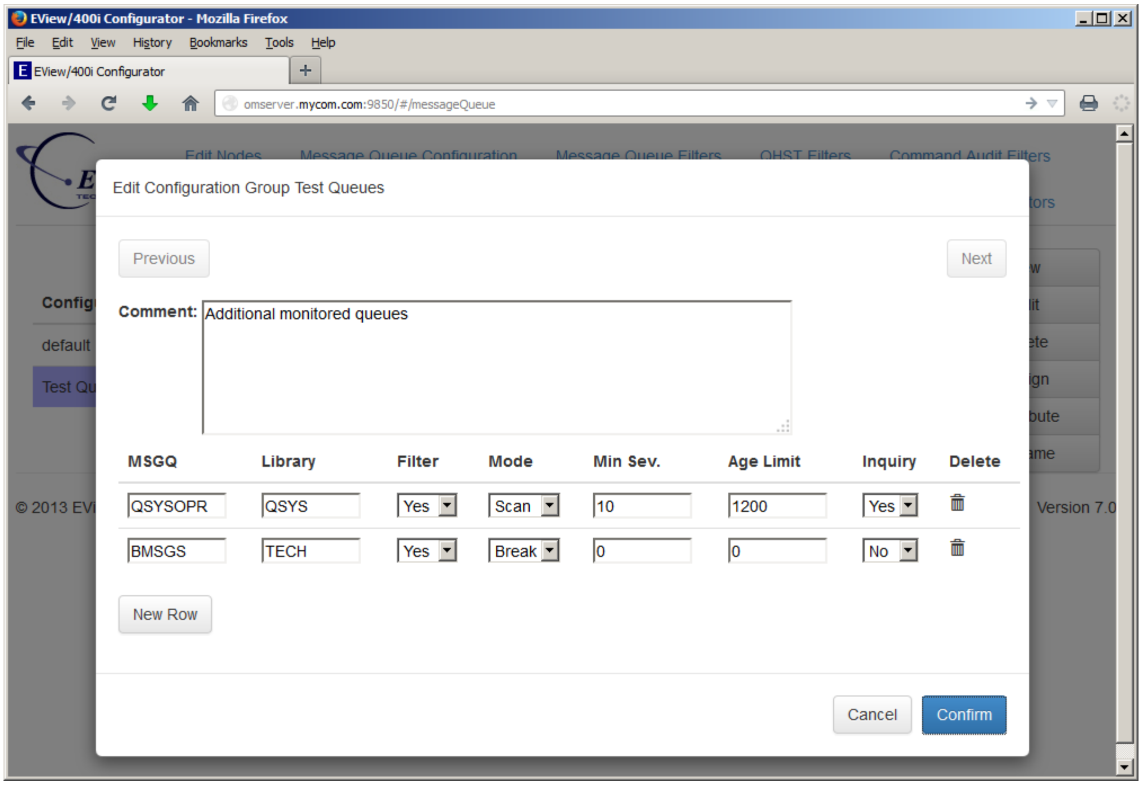
Configuration group name
default
Test Queues

© 2013 EView Technology, Inc.

Version 7.0

2. To change the message queues being monitored, add a new configuration group using the [New] button or edit an existing group using the [Edit] button.

Figure 3-5 Editing a Message Queue Group


EView/400i Configurator - Mozilla Firefox
File Edit View History Bookmarks Tools Help
EView/400i Configurator
omserver.mycom.com:9850/#/messageQueue
Previous
Next
New Row
Cancel
Confirm

Edit Configuration Group Test Queues

Comment: Additional monitored queues

MSGQ	Library	Filter	Mode	Min Sev.	Age Limit	Inquiry	Delete
QSYSOPR	QSYS	Yes	Scan	10	1200	Yes	🗑
BMSGS	TECH	Yes	Break	0	0	No	🗑

3. The QSYSOPR/QSYS queue is listed by default in a group. Use the [New Row] button to add another line for additional queues to be added to this group. To delete a listed queue, check the trash can icon to the right of the line. The options for each queue are:

- In the **MSGQ** field, enter the name of the message queue to be monitored.
- In the **Library** field, enter the name of the library where the message queue resides.
- Set the **Filter** option to one of the following:
  - "Yes" : if the message ID filters should be applied to messages coming from this queue, restricting which messages will be forwarded to the EView Proxy server.
  - "No" : allow messages to be passed on to EView Proxy regardless of their message ID.
  - "Sev" : Allow any non-inquiry messages with a severity equal to or greater than the Min Sev. field to be forwarded to the EView Proxy server regardless of the message ID. Messages with a severity less than the Min Sev. value will be forwarded only if the message ID is in the message ID filter table.
- Set the **Mode** option to "Break" to allow EView/400i to set the queue in *BREAK mode. EView/400i provides a break message-handling program that will be called each time a new message is written to the queue. Set the option to "Scan" to have EView/400i scan the queue on the interval (by default, every 5 seconds) to check for new messages.
  - Break Mode advantage: instant processing of incoming messages
  - Scan Mode advantage: does not require a lock on the message queue and can co-exist with other message queue monitoring programs.
- Set the **Min Sev.** field to a numeric value 0-99 indicating the necessary minimum severity of an incoming message. Messages with a lower severity will not be passed on to Eview Proxy, even if matched to a message ID filter. Enter "0" to allow all messages to be processed, regardless of severity.
- In the **Age Limit** field, enter a time limit (in seconds) of how old a message can be and still be passed on to the EView Proxy server. This field is only used for queues that are monitored with the "Scan" mode option (see above). This is useful during startup of the subsystem on the iSeries agent. When the subsystem is started for the first time (or if it has been brought down for any length of time), the Age Limit prevents the agent from sending a flood of old unnecessary messages to the EView Proxy server.
- If the **Inquiry** field is set to "Yes" then all messages in that queue with a Message Type of Inquiry (messages that ask for a reply) will be forwarded to the EView Proxy server, regardless of the message ID if the Filter option is set to "Yes".

4. Click the [Confirm] button when all message queues are added to the configuration group.

5. Click the [Assign] button to assign queue configuration groups to iSeries agents. The same configuration group may be assigned to multiple agents.

6. Select a queue configuration and click the [Distribute] button to send the list of monitored queues to the iSeries agent. The EView/400i EVSBS subsystem must be running on the agent at the time of the distribution.

When the EView/400i agent subsystem is running, it will begin monitoring message queues defined with Scan mode immediately after the distribution is completed. Queues defined with Break mode monitoring will begin monitoring after the next time the subsystem is restarted.

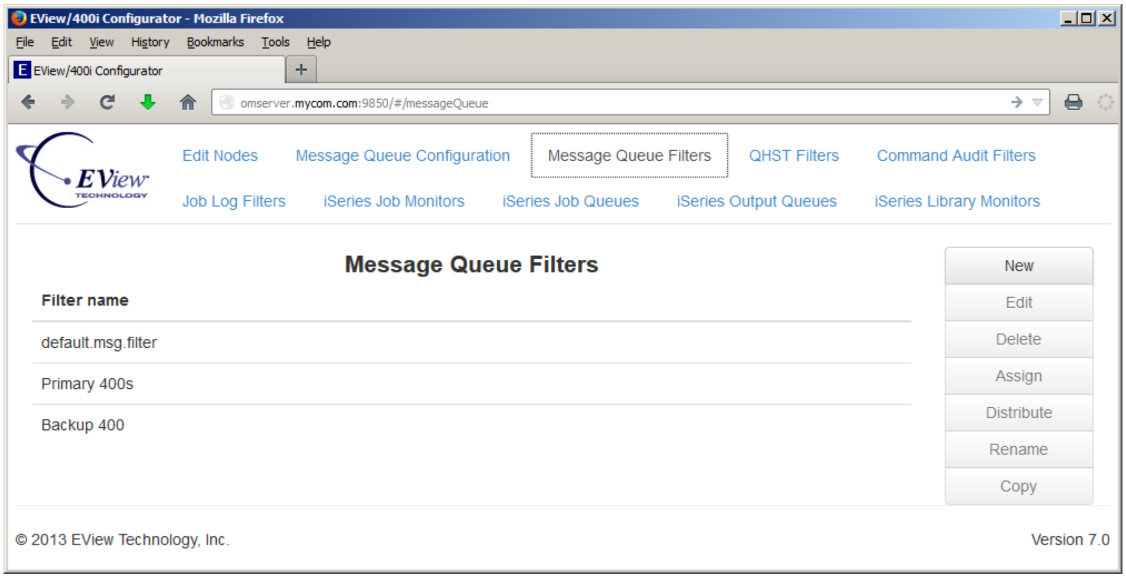
Configure Message ID Filters

Message ID filters restrict the number of messages that are sent from the iSeries agent to the EView Proxy server and save the server from receiving a flood of unnecessary messages. Each iSeries agent has two message filters, one for message queues and one for the QHST message log.

Message Queue Filters

Start the Message Queue Filters application from the EView/400i Configurator (Figure 3-1) by clicking on the [Message Queue Filters] button.

Figure 3-6: Message Queue Filters

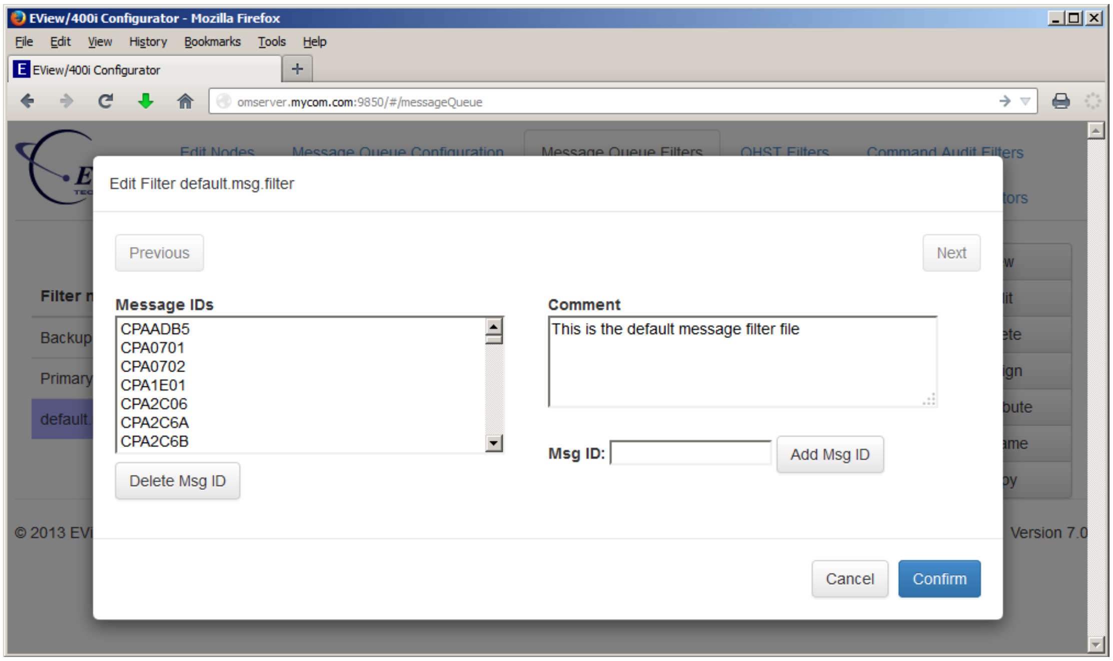


Message Queue Filters

Filter name
default.msg.filter
Primary 400s
Backup 400

1. To change the list of message IDs that are sent to the EView Proxy server, add a new filter group using the [New] button or edit an existing one using the [Edit] button. New filters may also be created by copying an existing filter or the supplied default filter (default.msg.filter) by selecting an existing filter and using the [Copy] button.

Figure 3-7: Editing a Message Queue Filter



Message IDs:

CPAADB5
CPA0701
CPA0702
CPA1E01
CPA2C06
CPA2C6A
CPA2C6B

Comment: This is the default message filter file

2. Enter new message IDs to the list in the open field and click the [Add Msg ID] button. To delete from the list, click the message ID(s) to remove and click the [Delete Msg ID] button.

   Message IDs must be no more than seven alphanumeric characters, but any message ID entered can contain the special period character (.) to indicate that any character in that position should match. If the message ID is terminated with an asterisk (*), matching will only occur on characters preceding the asterisk. See the following examples:

Table 3-1: Message Filter Examples

To forward the following messages:	Enter the following in the list of Message IDs:
All messages	`*`
ABC1234	`ABC1234`
All messages beginning with "ABC"	`ABC*`
Any 7-character message beginning with "ABC" and ending with "9"	`ABC...9`

3. Click [Confirm] button when all message IDs are added to the filter group.

4. Click the [Assign] button to assign filter groups to iSeries agents. The same filter group may be assigned to multiple agents.

5. Select a filter group name and click the [Distribute] button to send the list of message IDs to the iSeries agent. The EView/400i EVSBS subsystem must be running on the agent at the time of the distribution.

The EView/400i agent subsystem will begin monitoring with the new message ID filters immediately after the distribution is completed.

QHST Filters

Start the QHST Filters utility from the EView/400i Configurator by clicking on the [QHST Filters] button.

1. To change the list of message IDs that are sent to the EView Proxy server, add a new filter group using the [New] button or edit an existing one using the [Edit] button.
2. Enter new message IDs to the list in the open field and click the [Add Msg ID] button. To delete from the list, click the message ID(s) to remove and click the [Delete Msg ID] button.
3. Click the [Confirm] button when all message IDs are added to the filter group.
4. Click the [Assign] button to assign filter groups to iSeries agents. The same filter group may be assigned to multiple agents.
5. Select a filter group name and click the [Distribute] button to send the list of message IDs to the iSeries agent. The EView/400i EVSBS subsystem must be running on the agent at the time of the distribution.

The EView/400i agent subsystem will begin monitoring with the new message ID filters immediately after the distribution is completed.

Phase 3: Identify Job Log Messages

The Job Log message filter allows you to set up a file of message IDs that will be searched by the EVJLSCAN job log-scanning program on the iSeries agent. Any matching messages found in specified jobs will be returned. This can be useful in scanning the job log of a completed program for error messages.

It is important to know the name of the job that is being scanned because the EVJLSCAN program reads in message ID filters based on the job name.

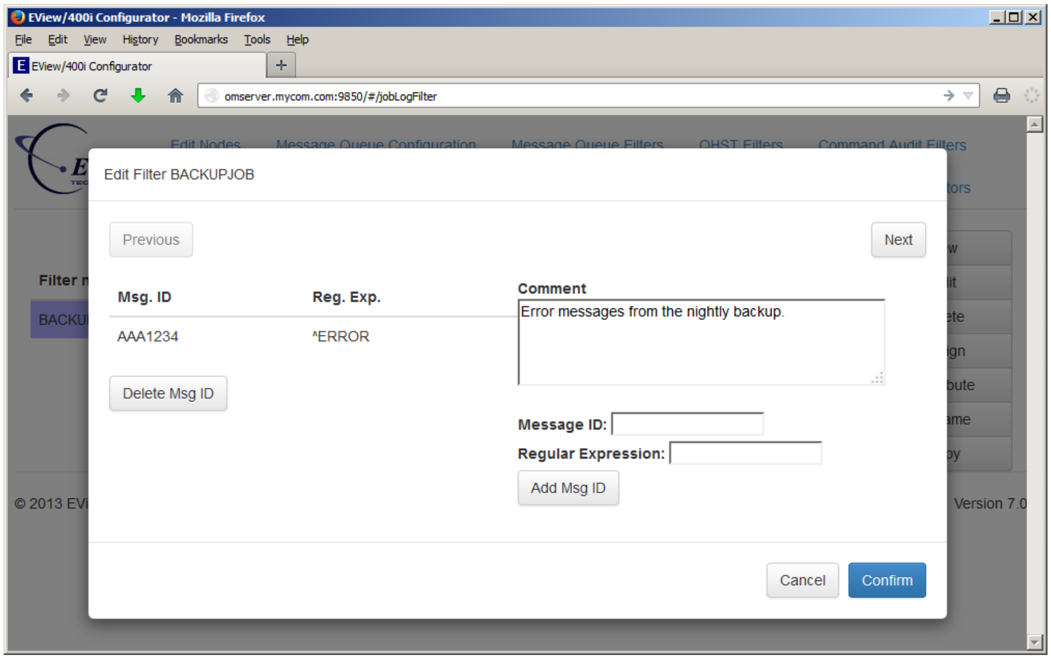
1. Start the Job Log Message Configurator utility from the EView/400i Configurator by clicking on the [Job Log Filters] button.

2. To change the list of message IDs that scanned from a specific job’s log, add a new filter group using the [New] button or edit an existing one using the [Edit] button.

   ➤ The name of the filter must match the job name that will be scanned on the iSeries system.

3. To add or edit message IDs in this group, use the [Add Message ID] and [Delete Msg ID] buttons. Enter the message ID in the “Message ID:” field. Optionally, enter a Unix-style regular expression to search for in the text of the message. For example, in the entry below, the joblog for the BACKUPJOB job will be scanned for message “AAA1234” that has the word “ERROR” at the beginning of the message.

Figure 3-8: Edit Job Log Message Criteria


EView/400i Configurator - Mozilla Firefox
File Edit View History Bookmarks Tools Help
EView/400i Configurator
omserver.mycom.com:9850/#/jobLogFilter
Edit Nodes Message Queue Configuration Message Queue Filters QHST Filters Command Audit Filters
Previous
Next
Delete Msg ID
Add Msg ID
Cancel
Confirm
© 2013 EVi
Version 7.0

Edit Filter BACKUPJOB

Msg. ID	Reg. Exp.
AAA1234	^ERROR

Comment: Error messages from the nightly backup.

Message ID:

Regular Expression:

4. Click the [Confirm] button when all message criteria are added to the job's filter list.

5. Click the [Assign] button to assign job name groups to iSeries agents. The same job name may be assigned to multiple agents. Adding agents to the list will highlight the Distribution box, indicating that assignments have been made but not yet sent to the agents.

6. Select a job name group and click the [Distribute] button to send the list of message criteria to the iSeries agent. The EView/400i EVSBS subsystem must be running on the agent at the time of the distribution.

After the job log group is distributed, it will be used the next time the EVJLSCAN program is called on that agent. See "Using the EVJLSCAN Program" on page 62 for information on how to scan job logs.

Phase 4: Identify Active Jobs to Monitor

The AS/400 Job Monitor works with deployed OMi policies to periodically verify that specified jobs or subsystems on the AS/400 are running. Policies can also be defined to check the count of running jobs with a specific name. Jobs found not to be running or under a specified count will generate an alert to the OMi event browser.

1. Start the Job Monitor Configurator utility from the EView/400i Configurator by clicking on the [Job Monitors] button.

2. Click the [Add Monitor] button to open a blank row to fill in a job or subsystem description.

3. Enter the AS/400 job name or subsystem name to be monitored in the Name field. Optionally, if the entry is for a job, enter the name of the job's subsystem in the Subsystem field. (If this entry is for a subsystem, leave the Subsystem field blank.)
4. The Monitor Type field can be used to optionally check for additional status:
   - "Active Job" for the default action (check that the job or subsystem is active).
   - "Job Count" to count the number of jobs with the specified name that are running.
   - "MSGW in Subsystem" to check if any jobs in the named subsystem are in a Message Wait (MSGW) status. (Use this keyword only when the specified name is a subsystem name.)
   - "MSGW on Job" to check if a specific job is in a Message Wait (MSGW) status.
   - "LCKW on Job" to check if a specific job is in a Lock Wait (LCKW) status.
   - "MTXW on Job" to check if a specific job is in a Mutex Wait (MTXW) status.
   - "TIMW on Job" to check if a specific job is in a Timed Interval Wait (TIMW) status.
   - "CMTW on Job" to check if a specific job is in a Commitment Control Wait (CMTW) status.
5. In the Host Name field, enter the fully qualified name of the AS/400 where the job or subsystem is running.
6. (Optional) In the Interval field, enter the time of day (HH:MM:SS) that the monitoring of this job should start. In the Duration field, enter the length of time that the monitoring should run (Duration is specified in hours, minutes, or seconds. To specify four hours, enter "4h". To specify 20 minutes, use "20m".) In the Days field, select which days of the week the job's active state should be monitored. (If no time period options are selected, monitoring is done at all times.)
7. (Optional) To remove a job monitor from the list, click the trash can icon. To temporarily deactivate a monitor without removing it from the list, check the "Inactive" checkbox.
8. Click the [Save] button to save the list of monitored jobs and subsystems.

To use these job and subsystem monitors, deploy the AS400_JOBMON policy in the EV400 policy group on OMi. To use the Job Count monitor, deploy the AS400_JOBCNT policy and specify the number of jobs to be counted in the Threshold Limit (Minimum) field.

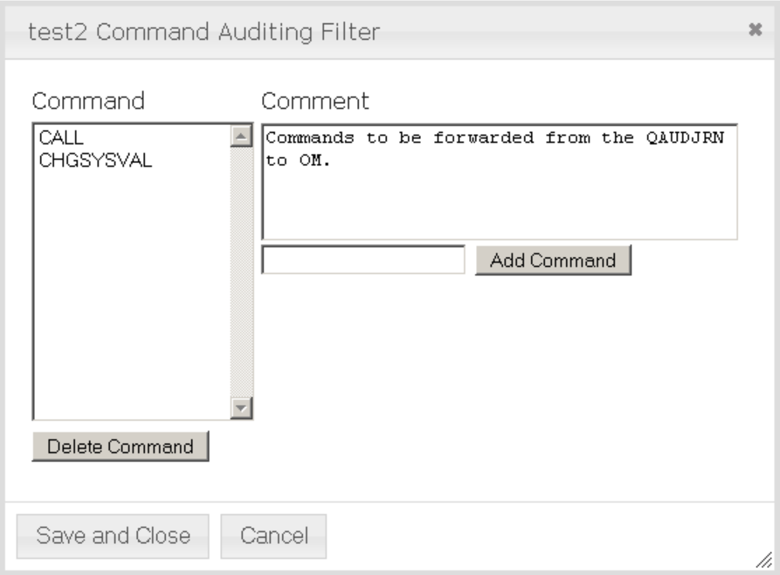
Phase 5: Identify Command Audit Filters

The Command Audit Filters work with the iSeries' QAUDJRN audit journal to determine which audit entries of type CD (Command) will be forwarded to the EView Proxy server. If an iSeries user's profile is set up (using CHGUSRAUD) to journalize the user's issued commands, the Command Audit Filters can be used to reduce the number of journal entries that are forwarded to the EView Proxy server.

Note that this section is only necessary if the “CD” value is specified in the EV400_MON_AUDJRNL parameter for this node.

1. To change the list of commands that are sent to the EView Proxy server, add a new filter group using the [New] button or edit an existing one using the [Edit] button.
2. Enter new commands to the list in the open field and click the [Add Command] button. To delete from the list, click the command(s) to remove and click the [Delete Command] button.
3. Click the [Save and Close] button when all commands are added to the filter group.

Figure 3-10: Editing the Command Audit Filters



test2 Command Auditing Filter

Command

CALL
CHGSYSVAL

Comment

Commands to be forwarded from the QAUDJRN to OM.

4. Click the [Assign] button to assign filter groups to iSeries agents. The same filter group may be assigned to multiple agents.
5. Select a filter group name and click the [Distribute] button to send the list of commands to the iSeries agent. The EView/400i EVSBS subsystem must be running on the agent at the time of the distribution.

➤ See Appendix E for the displayed format of the CD and other audit journal command types.

Phase 6: Assigning and Distributing the EView/400i Policies

In this phase of the EView/400i configuration process, you assign and distribute the EView/400i policies to the Micro Focus Operations Agent on the EView Proxy server, which then acts as the agent for the iSeries (AS/400) system.

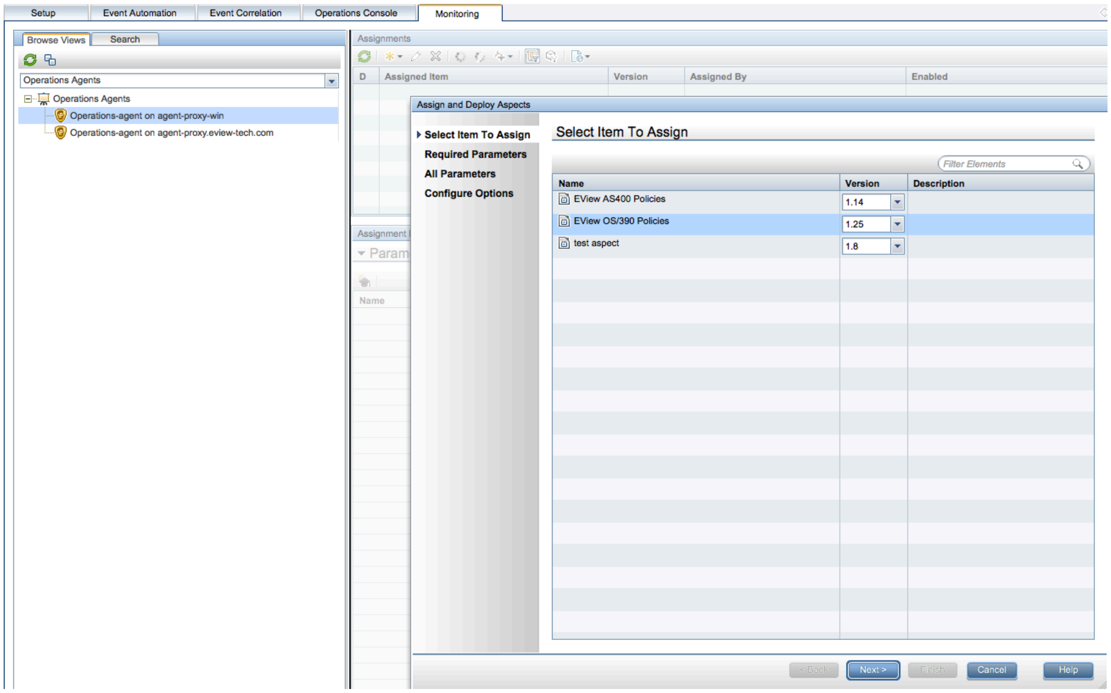
You must assign EView/400i policies to the Operations agent on the EView Proxy.server.

To Assign EView/400i Policies

To assign all of the EView/400i policies to the agent component of the Operations Manager server, follow these steps:

1. Start the OMi Administration GUI and login with a user ID that has administrative authority.
2. Navigate to the “Assignments and Tuning” item in the “Monitoring” tab.
3. Right-click in the Assignments pane and select “Add Aspect”.
4. In the Selector window, choose the EView AS400 Policies and click Next and then Finish in the next window.

Figure 3-11: Assigning the EV400 Policies



Select Item To Assign

Name	Version	Description
EView AS400 Policies	1.14	
EView OS/390 Policies	1.25	
test aspect	1.8	

To assign only selected EView/400i policies to the agent component on the EView proxy server, follow these steps.

1. Start the BSM Administration GUI and login with a user ID that has administrative authority.
5. Navigate to the “Policy Templates” item in the “Monitoring” tab.
6. Select the EV400 Policy Group from the list on the left.
7. Choose the policy to be assigned and deployed and right click on the policy and select “Assign and Deploy Policy Template”.

8. In the Selector window choose the EView proxy server node CI and click Next then Finish in the next window.

Phase 7: Starting the EView/400i Processes

After the iSeries agent configuration parameters have been customized and distributed to the agents, start the EView/400i processes on the EView Proxy server and the EVSBS subsystem on the iSeries agent (if not already running):

1. On the iSeries, enter one of the following commands to start the EVSBS subsystem for EView/400i:

```
CALL EVIEW/EVINIT
```

or

```
STRSBS EVIEW/EVSBS
```

Use the WRKACTJOB command to verify that the EVSBS subsystem is active.

Optional PARM values are available for the EVINIT command:

ALL	**Start all jobs in the subsystem that have been configured. This is the default option.**
CLEARQ	Clear any buffered messages from the EView message queues before starting the subsystem jobs.
TEST	Instead of starting the jobs, EVINIT will display the SBMJOB commands that would be used to start the jobs. This may be useful to verify that the jobs are being started with the desired options.
VERSION	Display the version of the installed agent software.
jobname	Start specific job(s) in the subsystem. If a job has fallen into a Message Wait status, use ENDJOB OPTION(*IMMED) to stop the individual job, then restart it by specifying the specific process name in the PARM when calling EVINIT. Specifying individual job names is only valid if the EVSBS subsystem is already running. Job names are listed in Appendix C.

Examples:

To clear the agent's internal data queues before starting the EVSBS agent subsystem:

```
CALL EVIEW/EVINIT  PARM(CLEARQ)
```

If the EVSBS subsystem is running, to restart only the EVSTCPPROC and EVTCTLPROC
jobs:

```
CALL EVIEW/EVINIT PARM('EVSTCPPROC EVTCTLPROC')
```

2. Use the "Edit Nodes" tab from the Configurator web page to start the server processes for the iSeries nodes, or enter the following Unix command:

   ```
   /opt/OV/vp400/bin/vp400sv -start [domain]
   ```

   where ***domain*** is the name of the iSeries node. If you use the iSeries short name, the vp400sv command will attempt a lookup to obtain the fully qualified name. If the managed iSeries node uses a different domain than the domain name configured for the name service, the fully qualified name must be used. If you omit ***domain***, vp400sv will start the EView/400i processes for all configured iSeries nodes.

To verify that EView/400i processes are running, click the [Refresh] button of the "Edit Nodes" tab on the Configurator web page, or enter the following Unix command:

```
/opt/OV/vp400/bin/vp400sv -status
```

Stopping the EView/400i Processes

1. To stop the EView/400 processes for an iSeries node, select the "Edit Node" tab from the Configurator web page and select the node(s) to stop, then click the `[Stop]` button, To terminate a running EView/400i subsystem on the iSeries agent, use the command:

   ```
   ENDSBS EVSBS *IMMED
   ```

➤ The EView/400i subsystem (EVSBS) must be ended prior to ending the TCPIP job or executing any save commands that would allocate an EView/400i object, such as performing a backup.

Phase 8: Configuring Nodes for Performance Data Collection

➤ Before performing this configuration you must have the Operations Manager Smart Plug-ins DSI-to-DDF wrapper utilities (DSI2DDF) component installed. This component is available on the Operations Manager i Smart Plug-in DVD.

Collecting performance metrics for iSeries nodes is an optional task and requires configuration steps to be performed on the iSeries agent and the EView proxy server. The iSeries agent must be running with the performance gathering job running under the EView/400i subsystem.

There are two sets of performance data metrics that can be collected at different intervals. The class specification files for these sets are listed in Appendix D. Browse these class specifications to determine which set(s) you want to collect.

The `vp400addperf.pl` script creates a performance class specification file for an iSeries node and updates the Operations agent performance data configuration.

To configure an iSeries node for performance data collection, follow these steps:

1. From a terminal or command window on the EView proxy server enter one of the following commands:

   For Linux enter the command:

   ```
   /opt/OV/vp400/bin/vp400addperf.pl node {1|2} [class]
   ```

   For Windows enter the command:

   ```
   perl "%EV400BIN%\ev400addperf.pl" node {1|2} [class]
   ```

   where:

   - ***node*** is the fully-qualified name of the iSeries system.

   - "**1**" or "**2**" represent whether to configure performance data set 1 or data set 2.

   - ***class*** can be optionally used to specify a class name for the specification under CODA or OVPA. If ***class*** is not specified, the default class name will be "OS400_PERF1_*nodename*" or "OS400_PERF2_*nodename*". (If the short name of the iSeries node is longer than eight characters, use the ***class*** parameter to specify a class name that will be less than the 20-character maximum name length required by OVPA/CODA.) Then use the Node Configurator application to change the EV400_PERF1_CLASS and/or the EV400_PERF2_CLASS parameter to the customized name (see "Add iSeries Nodes and Runtime Parameters" beginning on page 16).

   The `vp400addperf` script will create a class specification file for the iSeries node, and update either the CODA performance sub-agent or the OVPA. The script can be run twice to initialize both data set 1 and data set 2.

2. The Operations agent performance processes must be restarted.

3. Use the "Add/Edit Nodes" option of the EView/400i Configurator as explained in Phase 1 on page 15 to select an iSeries node, and then change the EV400_PERF1 and/or the EV400_PERF2 parameters to "YES". Change the EV400_PERF1_INT and/or the EV400_PERF2_INT parameters to the frequency (in minutes) that each of the data sets will be collected. Save the configuration parameters and distribute to the iSeries node. After the configuration parameters have been distributed, you must stop and restart the EVSBS agent subsystem to activate the new configuration.

4. The EView/400i server processes must be restarted to activate the interface to either the CODA sub-agent or OVPA. Use the Stop and Start options on the Add/Edit Nodes application from the EView/400i Configurator interface.

4

Using EView/400i

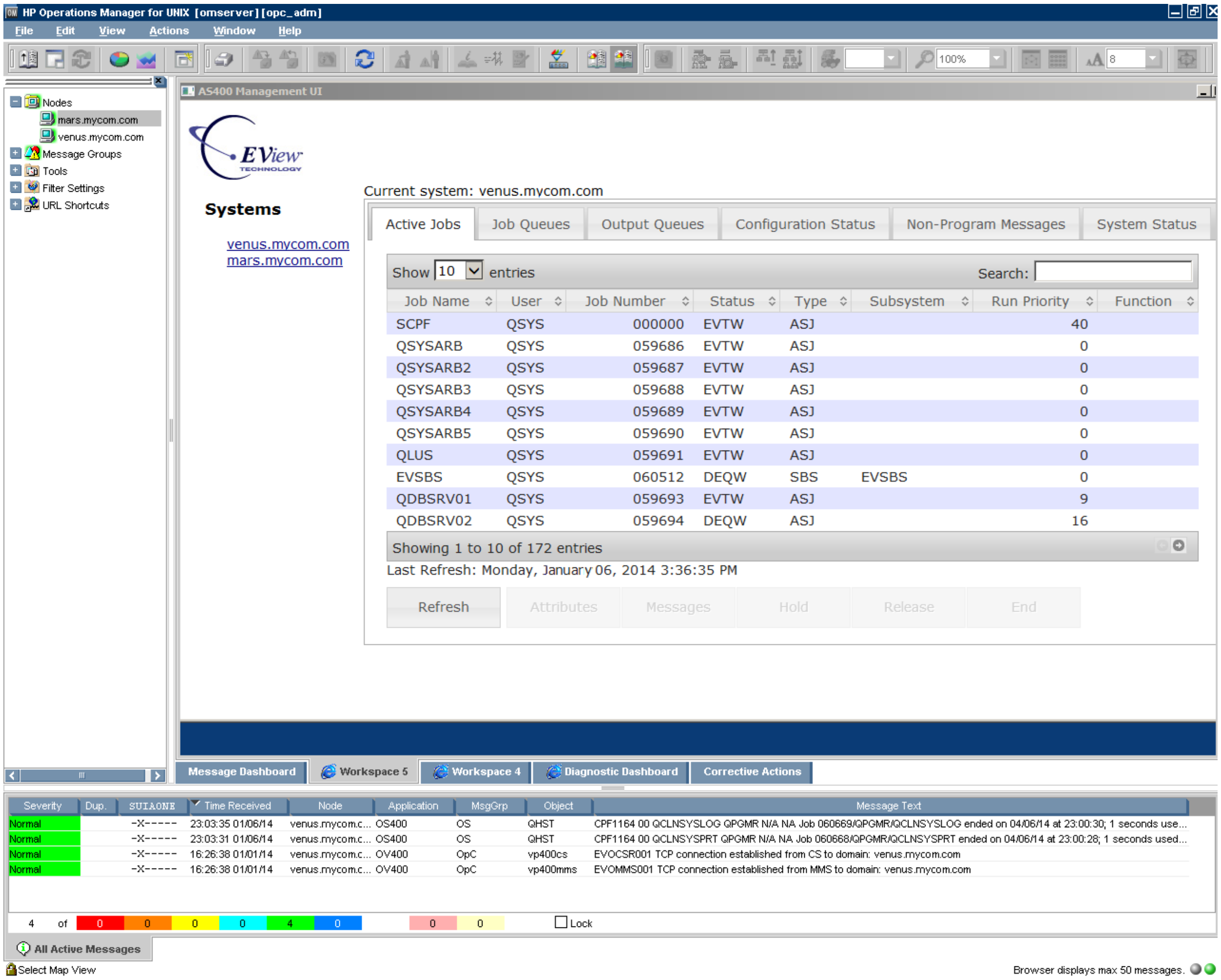
This chapter describes how to use EView/400i Management to perform daily tasks.

About the EView/400i Management UI

The EView/400i Management UI is an interactive tool that enables the operator to manage job queues, output queues, active jobs, and to monitor the system status.

Figure 4-1 shows the Management UI running in the Operations Java console.

Figure 4-1: EView/400i Management UI



Current system: venus.mycom.com

Show 10 entries Search:

Job Name	User	Job Number	Status	Type	Subsystem	Run Priority	Function
SCPF	QSYS	000000	EVTW	ASJ		40	
QSYSARB	QSYS	059686	EVTW	ASJ		0	
QSYSARB2	QSYS	059687	EVTW	ASJ		0	
QSYSARB3	QSYS	059688	EVTW	ASJ		0	
QSYSARB4	QSYS	059689	EVTW	ASJ		0	
QSYSARB5	QSYS	059690	EVTW	ASJ		0	
QLUS	QSYS	059691	EVTW	ASJ		0	
EVSBS	QSYS	060512	DEQW	SBS	EVSBS	0	
QDBSRV01	QSYS	059693	EVTW	ASJ		9	
QDBSRV02	QSYS	059694	DEQW	ASJ		16	

Showing 1 to 10 of 172 entries

Last Refresh: Monday, January 06, 2014 3:36:35 PM

Severity	Dup.	SUIAONE	Time Received	Node	Application	MsgGrp	Object	Message Text
Normal		-X-----	23:03:35 01/06/14	venus.mycom.c...	OS400	OS	QHST	CPF1164 00 QCLNSYSLOG QPGMR N/A NA Job 060669/QPGMR/QCLNSYSLOG ended on 04/06/14 at 23:00:30; 1 seconds use...
Normal		-X-----	23:03:31 01/06/14	venus.mycom.c...	OS400	OS	QHST	CPF1164 00 QCLNSYSPRT QPGMR N/A NA Job 060668/QPGMR/QCLNSYSPRT ended on 04/06/14 at 23:00:28; 1 seconds used...
Normal		-X-----	16:26:38 01/01/14	venus.mycom.c...	OV400	OpC	vp400cs	EVOCSR001 TCP connection established from CS to domain: venus.mycom.com
Normal		-X-----	16:26:38 01/01/14	venus.mycom.c...	OV400	OpC	vp400mms	EVOMMS001 TCP connection established from MMS to domain: venus.mycom.com

4 of 0 0 0 0 4 0 0 0 □ Lock

Browser displays max 50 messages.

Starting the Management UI Server

The AS400 Management UI server can be started from the EView proxy server on Linux with the command:

```
/etc/init.d/vp400mgmt start
```

On the EView proxy server on Windows, open the Windows Service manager and locate the EView/400 Management service and right click on the service and select Start.

(On subsequent reboots of the EView proxy server, the process will be started during the system initialization process.)

Using the EView/400i Management User Interface

Access the EView/400i Management user Interface using the following URL:

`http://eview-proxy:9851`

By default, the Management UI listens on port 9851 on the installed EView Proxy server. This port number may be customized by changing the EV400_MGMT_PORT value in the `/etc/opt/OV/share/conf/vp400/vp400info` file on the EView proxy server On Linux or the `vp400info` file in the `conf` subdirectory of the EView/400 installation path on Windows. If the port number is changed the EView/400 Management interface service must be restarted.

On Linux issue the commands:

`/etc/init.d/vp400mgmt stop`
and
`/etc/init.d/vp400mgmt start` (Solaris or Linux)

On the EView proxy server on Windows, open the Windows Service manager and locate the EView/400 Management service and right click on the service and select Stop and then Start.

Access to the management interface may be restricted to specific IP addresses or subnets by adding the EV400_MGMT_ALLOWED_IP_NETWORK value to this `vp400info` file, for example:

To allow a specific IP address to connect:

```
EV400_MGMT_ALLOWED_IP_NETWORK  192.168.0.222
```

To allow a range of IP subnets to connect (using either CIDR prefix or dotted decimal subnet mask):

```
EV400_MGMT_ALLOWED_IP_NETWORK  10.5.0.0/16
EV400_MGMT_ALLOWED_IP_NETWORK  192.168.7.0/255.255.255.0
```

Multiple EV400_MGMT_ALLOWED_IP_NETWORK entries are allowed in the `vp400info` file.

Monitor Active Jobs

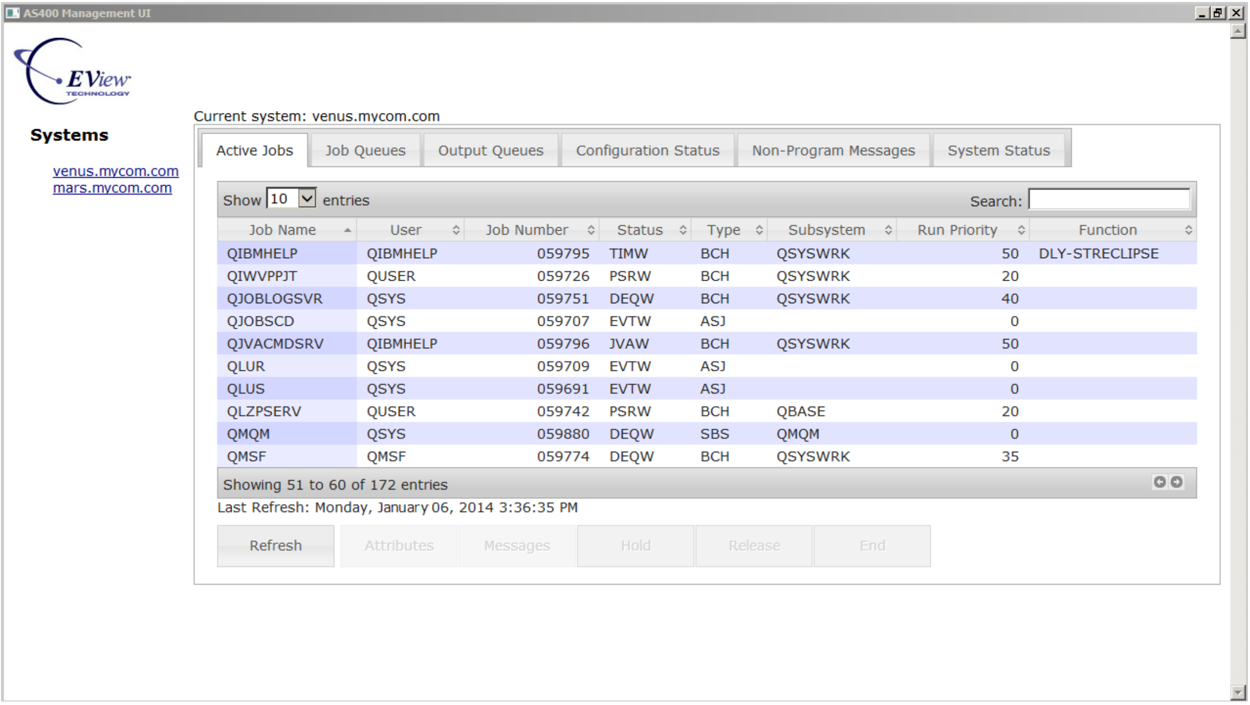
1. Select the "`Active Jobs`" tab.
2. To perform actions on a specific job select the job. By clicking the appropriate button you may perform the following:
    - [Attributes] – Displays attributes of the selected active job

- [Messages] – Displays the last 500 lines of the active job log
- [Hold] – Places the selected active job in hold status
- [Release] – Removes the selected job from hold status
- [End] – Ends job processing. A pop-up window will be displayed to confirm ending the job via a `[Controlled]` or `[Immediate]` option.

➤ Status changes in the display will not be updated until the [Refresh] button is pressed.

The output can be sorted by clicking on the column header. The initial sort direction is ascending. Subsequent clicking on the column header will reverse the sort direction.

Figure 4-2: Active Jobs



Current system: venus.mycom.com

Job Name	User	Job Number	Status	Type	Subsystem	Run Priority	Function
QIBMHELP	QIBMHELP	059795	TIMW	BCH	QSYSWRK	50	DLY-STRECLIPSE
QIWVPPJT	QUSER	059726	PSRW	BCH	QSYSWRK	20	
QJOBLOGSVR	QSYS	059751	DEQW	BCH	QSYSWRK	40	
QJOBSCD	QSYS	059707	EVTW	ASJ		0	
QJVACMDSRV	QIBMHELP	059796	JVAW	BCH	QSYSWRK	50	
QLUR	QSYS	059709	EVTW	ASJ		0	
QLUS	QSYS	059691	EVTW	ASJ		0	
QLZPSERV	QUSER	059742	PSRW	BCH	QBASE	20	
QMQM	QSYS	059880	DEQW	SBS	QMQM	0	
QMSF	QMSF	059774	DEQW	BCH	QSYSWRK	35	

Showing 51 to 60 of 172 entries

Last Refresh: Monday, January 06, 2014 3:36:35 PM

➤ When a button is gray the action is not available for the selected active job.

Manage Job Queues

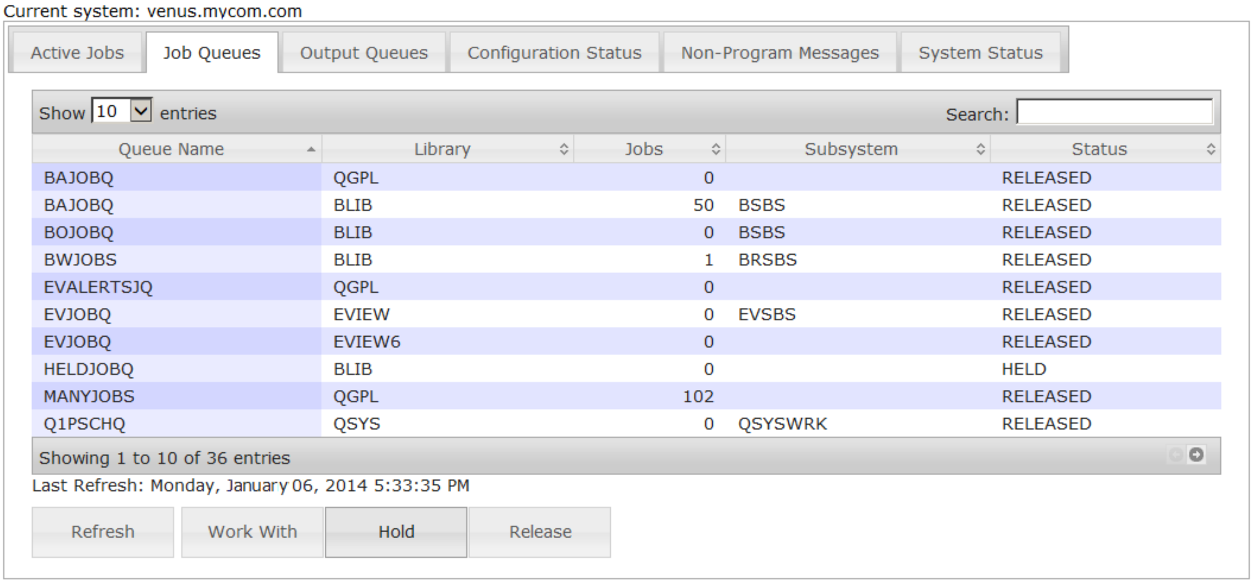
1. Select the `[Job Queues]` tab.
2. Select the Job Queue you wish to monitor.
3. By clicking the appropriate button you may perform the following:

- `[Work With]` – Displays a list of jobs on the selected queue.
- `[Hold]` – Places the selected job queue in a hold status.
- `[Release]` – Removes the hold that was placed on the selected job queue.

Status changes in the display will not be updated until the `[Refresh]` button is pressed.

4. The output can be sorted by clicking on the column header. The initial sort direction is ascending. Subsequent clicking on the column header will reverse the sort direction.

Figure 4-3: Job Queues



Current system: venus.mycom.com

Queue Name	Library	Jobs	Subsystem	Status
BAJOBQ	QGPL	0		RELEASED
BAJOBQ	BLIB	50	BSBS	RELEASED
BOJOBQ	BLIB	0	BSBS	RELEASED
BWJOBS	BLIB	1	BRSBS	RELEASED
EVALERTSJQ	QGPL	0		RELEASED
EVJOBQ	EVIEW	0	EVSBS	RELEASED
EVJOBQ	EVIEW6	0		RELEASED
HELDJOBQ	BLIB	0		HELD
MANYJOBS	QGPL	102		RELEASED
Q1PSCHQ	QSYS	0	QSYSWRK	RELEASED

Showing 1 to 10 of 36 entries

Last Refresh: Monday, January 06, 2014 5:33:35 PM

When a button is gray the action is not available for the selected submitted job.

Work With Selected Job Queue

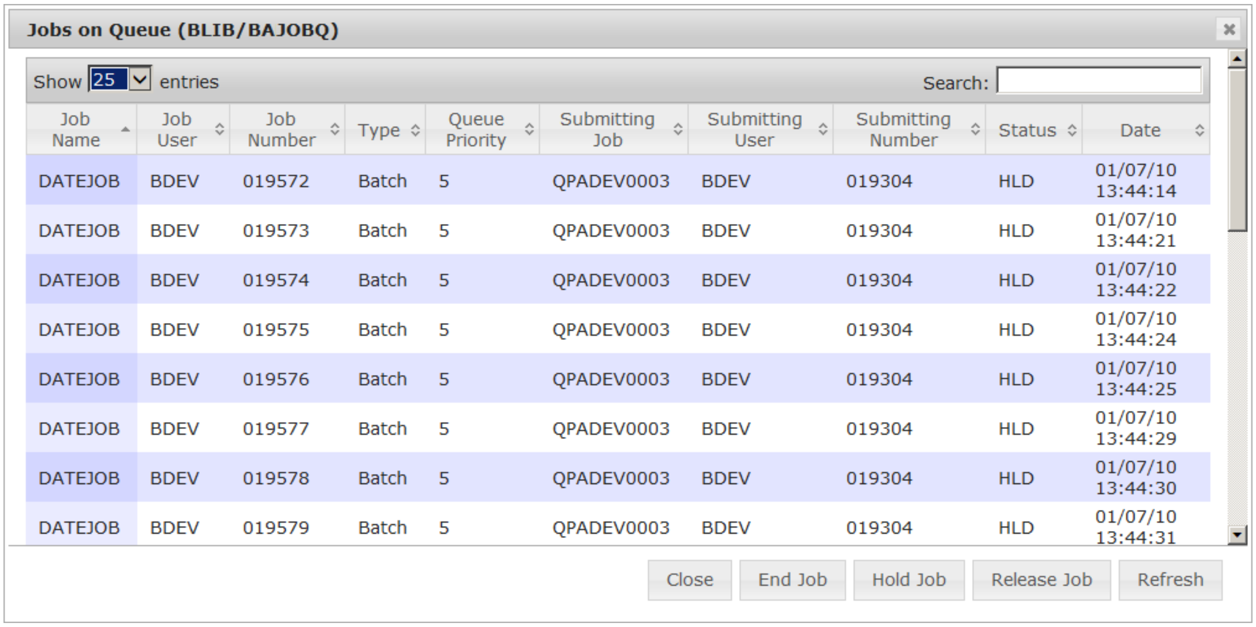
1. Select the Job Queue you wish to work with.
2. Click the [Work With] button.
3. A list of jobs in that queue will be displayed.
4. Select the job you wish to manipulate.

5. By clicking the appropriate button you may perform the following:
    - [Hold] – Places the selected job in a hold status.
    - [Release] – Removes the hold that was placed on the selected job.
    - [End] – Causes the selected job to end.

Status changes in the display will not be updated until the [Refresh] button is pressed.

6. The output can be sorted by clicking on the column header. The initial sort direction is ascending. Subsequent clicking on the column header will reverse the sort direction.

Figure 4-4: Work With Job Queue



Jobs on Queue (BLIB/BAJOBQ)

Show 25 entries Search:

Job Name	Job User	Job Number	Type	Queue Priority	Submitting Job	Submitting User	Submitting Number	Status	Date
DATEJOB	BDEV	019572	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:14
DATEJOB	BDEV	019573	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:21
DATEJOB	BDEV	019574	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:22
DATEJOB	BDEV	019575	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:24
DATEJOB	BDEV	019576	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:25
DATEJOB	BDEV	019577	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:29
DATEJOB	BDEV	019578	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:30
DATEJOB	BDEV	019579	Batch	5	QPADEV0003	BDEV	019304	HLD	01/07/10 13:44:31

When a button is gray the action is not available for the selected active job.

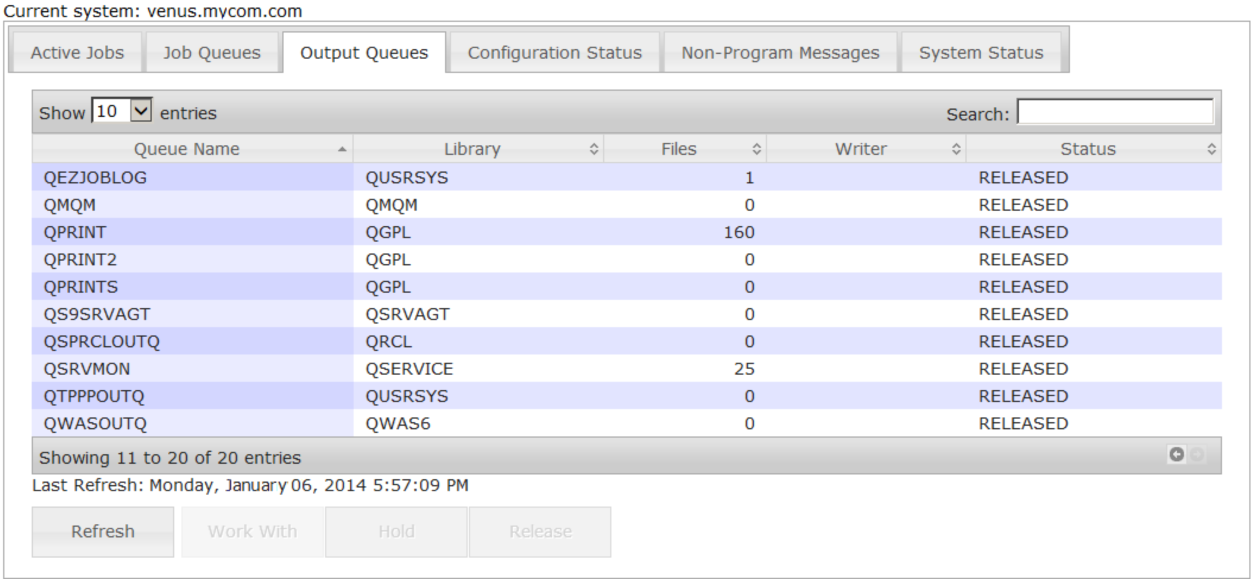
Manage Output Queues

1. Click on the "Output Queues" tab.
2. Select the Output Queue you wish to monitor.
3. By clicking the appropriate button you may perform the following:
    - [Work With] – Displays a list of files on the output queue.

- [Hold] – Places the selected output queue in hold status
- [Release] – Removes a hold that was placed on the selected output queue.

Status changes in the display will not be updated until the [Refresh] button is pressed.

Figure 4-5: Output Queues



Current system: venus.mycom.com

Queue Name	Library	Files	Writer	Status
QEZJOBLOG	QUSRSYS	1		RELEASED
QMQM	QMQM	0		RELEASED
QPRINT	QGPL	160		RELEASED
QPRINT2	QGPL	0		RELEASED
QPRINTS	QGPL	0		RELEASED
QS9SRVAGT	QSRVAGT	0		RELEASED
QSPRCLOUTQ	QRCL	0		RELEASED
QSRVMON	QSERVICE	25		RELEASED
QTPPPOUTQ	QUSRSYS	0		RELEASED
QWASOUTQ	QWAS6	0		RELEASED

Showing 11 to 20 of 20 entries

Last Refresh: Monday, January 06, 2014 5:57:09 PM

When a button is gray the action is not available for the selected submitted job.

Work With Selected Output Queue

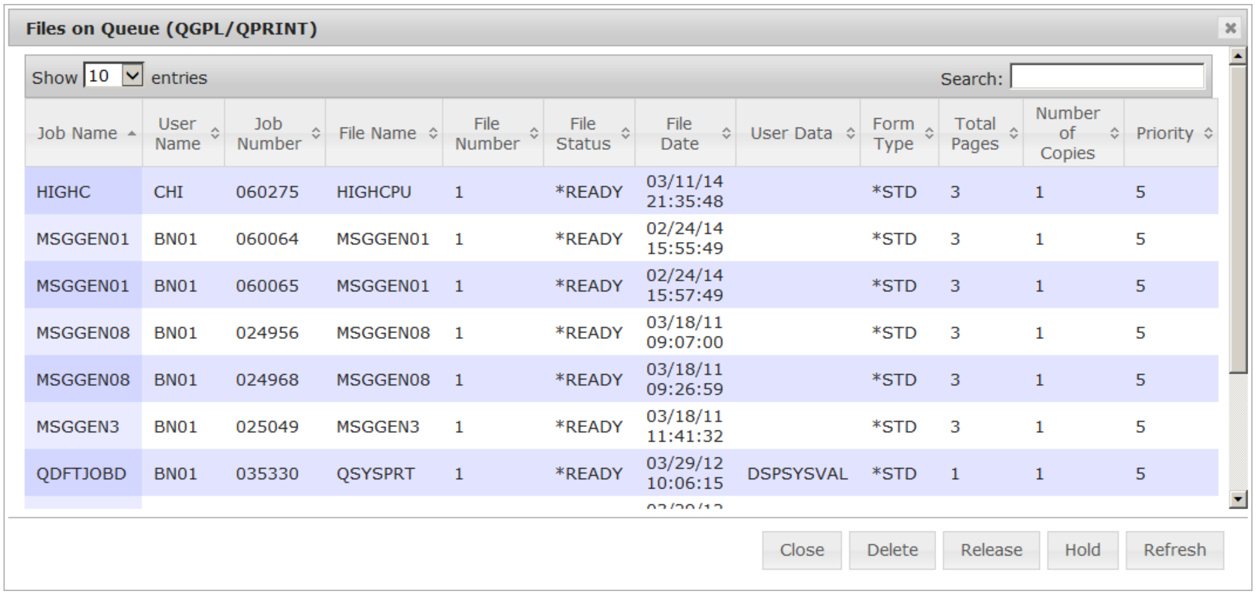
1. Select the Output Queue you wish to work with.
2. Click the [Work With] button.
3. The Work With Output Queue window appears in the Workspace pane with a list of spool files for the selected output queue.
4. Select the job you wish to manipulate.
5. By clicking the appropriate button you may perform the following:
    - [Delete] – Causes the selected spool file to be deleted.

- [Release] – Removes the hold that was placed on the selected spool file.
- [Hold] – Places the selected spool file in a hold status.

Status changes in the display will not be updated until the [Refresh] button is pressed.

6. The output can be sorted by clicking on the column header. The initial sort direction is ascending. Subsequent clicking on the column header will reverse the sort direction.

Figure 4-6: Work With Output Queues



Files on Queue (QGPL/QPRINT)

Show 10 entries Search:

Job Name	User Name	Job Number	File Name	File Number	File Status	File Date	User Data	Form Type	Total Pages	Number of Copies	Priority
HIGHC	CHI	060275	HIGHCPU	1	*READY	03/11/14 21:35:48		*STD	3	1	5
MSGGEN01	BN01	060064	MSGGEN01	1	*READY	02/24/14 15:55:49		*STD	3	1	5
MSGGEN01	BN01	060065	MSGGEN01	1	*READY	02/24/14 15:57:49		*STD	3	1	5
MSGGEN08	BN01	024956	MSGGEN08	1	*READY	03/18/11 09:07:00		*STD	3	1	5
MSGGEN08	BN01	024968	MSGGEN08	1	*READY	03/18/11 09:26:59		*STD	3	1	5
MSGGEN3	BN01	025049	MSGGEN3	1	*READY	03/18/11 11:41:32		*STD	3	1	5
QDFTJOBD	BN01	035330	QSYSPRT	1	*READY	03/29/12 10:06:15	DSPSYSVAL	*STD	1	1	5

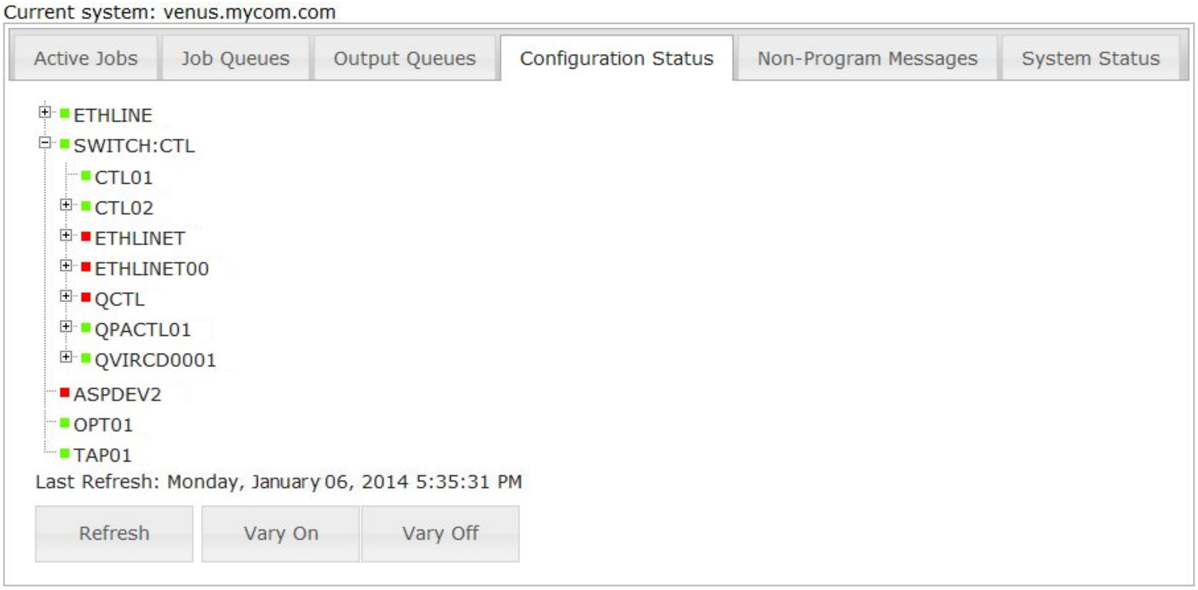
When a button is gray the action is not available for the selected active job.

Monitor Configuration Status

This tab shows the status of the defined lines, controllers, and devices of the system.

1. Select the "`Configuration Status`" tab.
2. By clicking the appropriate button you may perform the following:
   - [Vary On] – Activate an object.
   - [Vary Off] – Deactivate an object.

Figure 4-7: Work With Configuration Status



Current system: venus.mycom.com

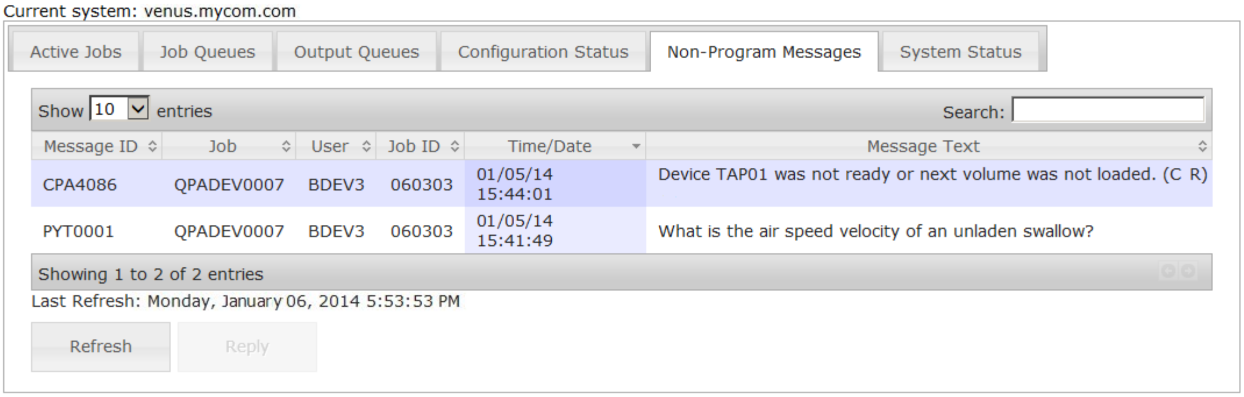
- 🟩 ETHLINE
- 🟩 SWITCH:CTL
  - 🟩 CTL01
  - 🟩 CTL02
  - 🟥 ETHLINET
  - 🟥 ETHLINET00
  - 🟥 QCTL
  - 🟩 QPACTL01
  - 🟩 QVIRCD0001
- 🟥 ASPDEV2
- 🟩 OPT01
- 🟩 TAP01

Last Refresh: Monday, January 06, 2014 5:35:31 PM

Manage Outstanding Inquiry Messages

1. Select the `"Non-Program Messages"` tab.
2. By clicking the appropriate button you may perform the following:
   - [Refresh] – Refreshes the list of outstanding inquiry messages.
   - [Reply] – Reply to the currently selected message. A dialog window will be displayed to accept the reply input.

Figure 4-8 Display Outstanding Inquiry Message



Current system: venus.mycom.com

Message ID	Job	User	Job ID	Time/Date	Message Text
CPA4086	QPADEV0007	BDEV3	060303	01/05/14 15:44:01	Device TAP01 was not ready or next volume was not loaded. (C R)
PYT0001	QPADEV0007	BDEV3	060303	01/05/14 15:41:49	What is the air speed velocity of an unladen swallow?

Showing 1 to 2 of 2 entries

Last Refresh: Monday, January 06, 2014 5:53:53 PM

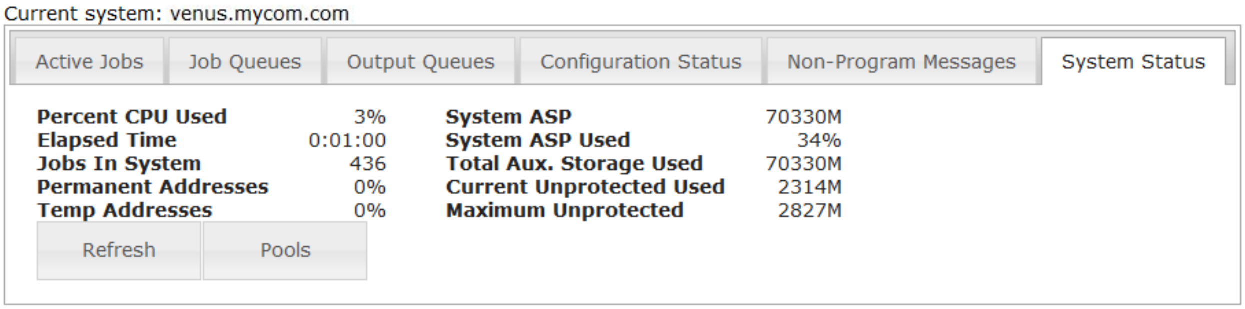
Monitor System Status

3. Select the "`System Status`" tab.

4. By clicking the appropriate button you may perform the following:

   - [Pools] – Displays system pool information.
   - [Refresh] – Refreshes the system status data as shown in Figure 4-9.

Figure 4-9: Monitor System Status



Current system: venus.mycom.com

Percent CPU Used	3%	**System ASP**	70330M
Elapsed Time	0:01:00	**System ASP Used**	34%
Jobs In System	436	**Total Aux. Storage Used**	70330M
Permanent Addresses	0%	**Current Unprotected Used**	2314M
Temp Addresses	0%	**Maximum Unprotected**	2827M

About the EView/400i Node Group

The installation of EView/400i creates a "400" node group which should be used to hold all AS/400 systems in the EView/400i environment.

By assigning the "400" node group and the OS and Network message groups to the OM operator responsible for the EView/400i environment, you ensure that messages relating to errors and potential problems with AS/400 systems appear in the appropriate operator's Message Browser.

About the EView/400i Tool Group: iSeries Tools

The installation of EView/400i creates a Tool Group named "iSeries Tools" which contains applications designed to help OM operators manage and monitor the iSeries environment. This group holds the tool that points to the Management UI URL. This group also contains two subgroups:

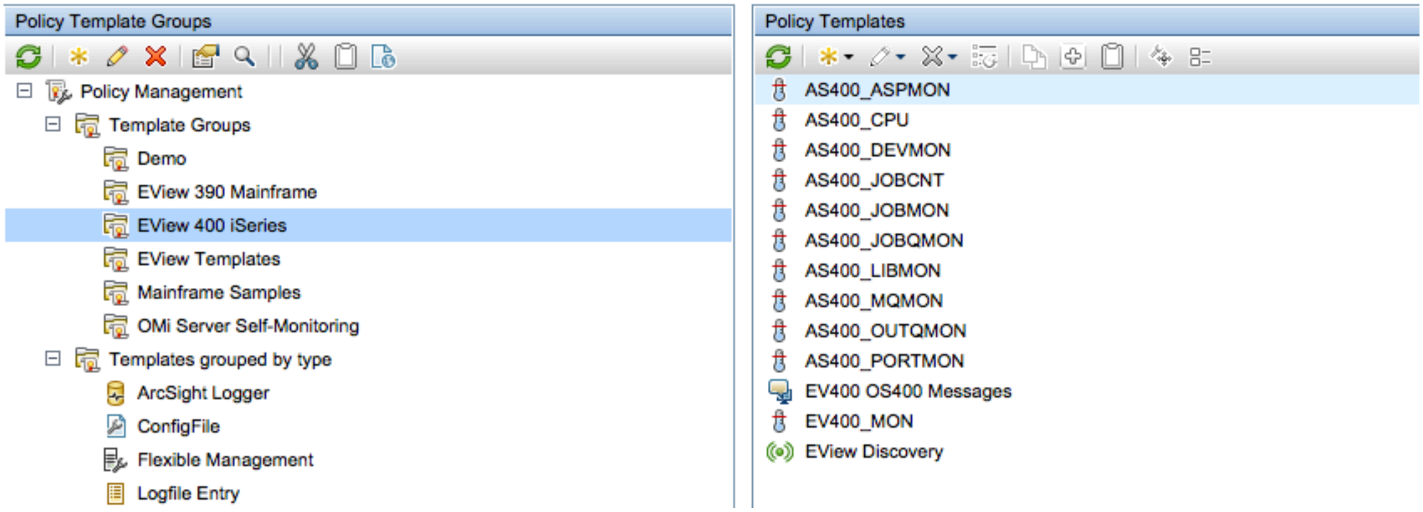
- "AS400 Configuration" contains administrative tools used to set up the iSeries (AS/400) monitoring options. This group should be assigned to administrative personnel.
- "AS400 Status" contains operator tools for displaying active iSeries (AS/400) job and queue statuses. This group should be assigned to general operators.

.

About EView/400i Policies

One policy group “EView 400 iSeries” contains all the default policies provided with EView/400i, as shown in Figure 4-9.

Figure 4-10: Default EView/400i Policies



Policy Template Groups

- Policy Management
  - Template Groups
    - Demo
    - EView 390 Mainframe
    - EView 400 iSeries
    - EView Templates
    - Mainframe Samples
    - OMi Server Self-Monitoring
  - Templates grouped by type
    - ArcSight Logger
    - ConfigFile
    - Flexible Management
    - Logfile Entry

Policy Templates

- AS400_ASPMON
- AS400_CPU
- AS400_DEVMON
- AS400_JOBCNT
- AS400_JOBMON
- AS400_JOBQMON
- AS400_LIBMON
- AS400_MQMON
- AS400_OUTQMON
- AS400_PORTMON
- EV400 OS400 Messages
- EV400_MON
- EView Discovery

Intercepting Messages

The EView/400i message source policies intercept messages that are generated on the iSeries managed node.

Use the default messages listed in the “EV400 OS400 Messages” policy as a guide for creating automatic and operator-initiated actions for other messages.

Any message listed must also exist in the message filter table on the managed node. See “Configure Message ID Filters” on page 30 to add message IDs to the agent’s message filter table.

Message Formats

Messages received from the iSeries system are passed through the OM message stream interface. If the EV400_ADDMSG_FIELDS option is set to “YES” (see description on page 17), then the format of the original message is:

<MsgId> <Severity> <JobName> <User> <ProgName> <MsgType> <MsgText>

If the EV400_ADDMSG_FIELDS option is set to “NO”, then the format of the original message is:

<MsgId> <Severity> <JobName> <User> <MsgText>

where the following are separated by spaces:

<MsgId> the 7 character message ID

<Severity> The 2 character severity code (00-99)

<JobName> The name of the job that generated the message

<User> The user profile that ***JobName*** is running under

<ProgName> The name of the program that issued the message. Note that for messages from the history log (QHST), the program name will be set to "N/A".

<MsgType> The message type. The possible values are:

01 – Completion
02 – Diagnostic
04 – Informational
05 – Inquiry
06 – Sender's Copy
08 – Request
10 – Request with prompting
14 – Notify
15 – Escape
NA – Not available

Messages from the history log (QHST) will have the ***MsgType*** set to "NA".

<MsgText> The actual text of the message

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

Table 4-1: Optional Variables

$MSG_OBJECT	The queue name that the message was received from
$OPTION(msgkey)	The unique message key assigned to the message on the agent
$OPTION(netid)	The agent netid
$OPTION(location)	The agent location (system) name
$OPTION(msgdate)	The agent system date when the message was created, formatted MM/DD/YY
$OPTION(msgtime)	The agent system time when the message was created, formatted HH:MM:SS
$OPTION(msgid)	The 7 character message ID
$OPTION(msgtype)	The 2 character message type: 01 – Completion 02 – Diagnostic 04 – Informational 05 – Inquiry 06 – Sender's Copy 08 – Request 10 – Request with prompting 14 – Notify 15 – Escape NA – Not Available
$OPTION(severity)	The 2 character severity code (00-99)
$OPTION(jobname)	The name of the job that generated the message

$OPTION(user)	The username that "jobname" is running under
$OPTION(jobid)	The 6 character job ID of the job that issued the message

These variables can be used in displayed message text as well as passed as arguments to automatic or operator-initiated commands.

Sending Commands to the Agent

Sending Native Commands Using vp400hostcmd

To send a native OS/400 command, use the vp`400hostcmd` program located in the `/bin` subdirectory of the EView/400i installation path. Any response from the agent is directed to stdout. The format of `vp400hostcmd` is:

`vp400hostcmd <`*`type`*`> <`*`command`*`>.<`*`iSeriesHostname`*`>`

where:

`<type>`	The command type. Use one of the following codes: **`80`** = The OS/400 command is issued and vp400hostcmd waits for the response from the agent. **`85`** = The OS/400 command is issued and vp400hostcmd ends immediately after receiving confirmation that the command has been sent. Use type 85 for commands that do not produce a response, or if the response can be ignored. **`86`** = A special EView/400 internal command that requests information on certain agent system resources. See "Using the EView/400i Agent Interface to System APIs" on page 62.
`<command>`	For type 80/85 commands: the OS/400 native command. The first period encountered marks the end of the command. If the *`command`* text has a period in it, enter two periods to signify that it is not the end of the command. For type 86 commands, enter the desired API code and any necessary parameters.
`<iSeriesHostname>`	The iSeries node where the command is to be executed. Use the node name that was used when defining the agent to EView/400i.

Examples:

1. Send an OS/400 command to the agent named myhost.mysite.com to display the active jobs:

```
vp400hostcmd 80 WRKACTJOB.myhost.mysite.com
```

2. Send a command to the agent named myhost.mysite.com to call a program named MY.PROGRAM that does not generate output (note that the program name contains a period, so use a double period in the command call to indicate that it is not marking the end of the command):

```
vp400hostcmd 85 CALL MYLIBRARY/MY..PROGRAM.myhost.mysite.com
```

3. Send a command to the agent named myhost.mysite.com to collect a list of all running jobs:

```
vp400hostcmd 86 01.myhost.mysite.com
```

Operator Action Replies

For operator-initiated replies to iSeries inquiry messages, use the following text in the policy’s Operator initiated command field for the message condition:

```
http://<$MSG_GEN_NODE_NAME>:9851/?system=<$MSG_NODE_NAME>&tab=4
```

which will open the Non-Program Messages tab of the EView/400i Management interface. (If the default port number 9851 has been changed (check the EV400_MGMT_PORT value stored in the /etc/opt/OV/share/conf/vp400info file on the management server), then use the modified value in the command.) From this browser interface, the operator can select the outstanding inquiry message and send a reply. See “Manage Outstanding Inquiry Messages” on page 49.

Automatically Acknowledging Replies

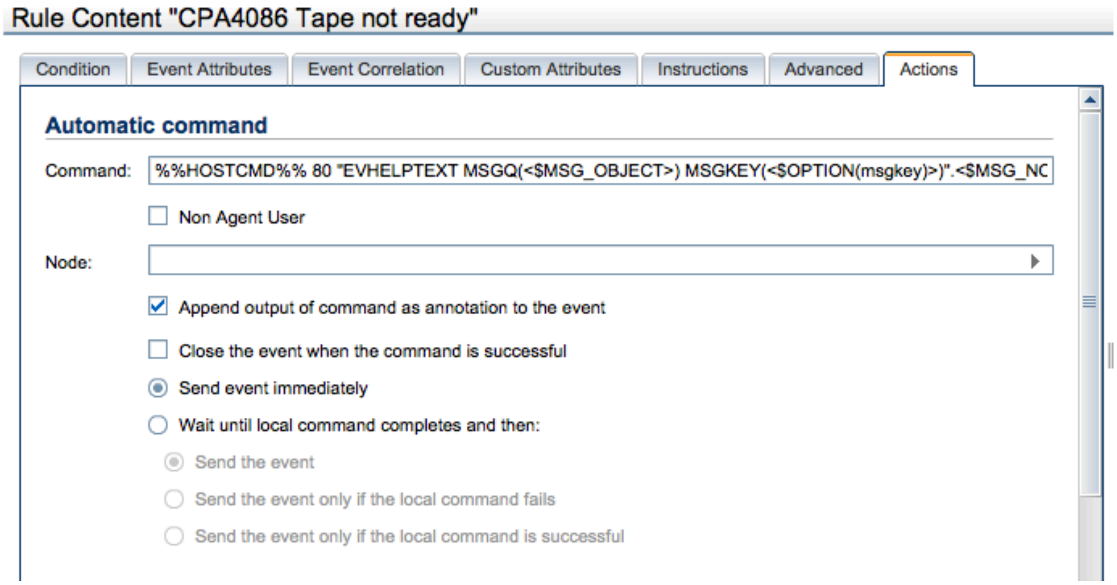
EView/400i generates an EVM5000 message on the iSeries agent when an outstanding inquiry message receives a response. The EVM5000 message ID is listed in the EView/400i QHST message filters, and the EVM5000 is sent to the EView Proxy server and can be used to acknowledge the inquiry message in the OM Active Messages browser. This acknowledgement will occur regardless of where the response was entered (either from an iSeries terminal session or an OM reply).

The default "EV400 OS400 Messages" policy has a rule defined to capture EVM5000 messages and correlate them to existing messages. When adding inquiry messages to the OS/400 message policy, enter the following key to use automatic acknowledgement:

```
<$MSG_NODE_NAME>:<$OPTION(msgid)>:<$OPTION(jobname)>:<$OPTION(user)>
:<$OPTION(jobid)>:<$OPTION(msgkey)>:INQUIRY
```

See Figure 4-11 for an example of adding the message key to the CPA4086 message condition to acknowledge it after a reply has been received.

Figure 4-11: Adding a Message Key to OS/400 Message Condition to Allow Auto-Acknowledgement After Message Has Received a Reply



Rule Content "CPA4086 Tape not ready"

Automatic command

Command: %%HOSTCMD%% 80 "EVHELPTEXT MSGQ(<$MSG_OBJECT>) MSGKEY(<$OPTION(msgkey)>)".<$MSG_NC

□ Non Agent User

Node:

☑ Append output of command as annotation to the event

□ Close the event when the command is successful

● Send event immediately

○ Wait until local command completes and then:

- ● Send the event
- ○ Send the event only if the local command fails
- ○ Send the event only if the local command is successful

Inquiry messages that receive an automatic reply using the message default reply will be formatted using the message ID “EVM5001”. To forward these replies to the OM server, add “EVM5001” to the list of QHST messages using the "QHST Filters" tab from the EView/400i Configuration application. Since messages which have an automatic system reply usually do not need to be seen by operators, the EVM5001 is not entered in the default QHST message filter table.

Retrieving Second Level (Help) Text for OS/400 Messages

A supplemental command EVHELPTEXT is available for the EView/400i agent that can be used to retrieve second level (help) message text for OS/400 messages that are received from a monitored message queue.

➤ This command cannot be used to retrieve second level text from messages that have been forwarded from the history log (QHST).

To automatically provide second level text as part of a received message’s annotations, the EVHELPTEXT command may be used in an automatic action of the desired messages. The syntax for executing the EVHELPTEXT command in an automatic action is:

```
EVHELPTEXT MSGQ(<$MSG_OBJECT>) MSGKEY(<$OPTION(msgkey)>)
```

Fill in the automatic action's Node field with **<$MSG_NODE_NAME>** and set the Annotations to "Yes".

Displaying Physical File Contents Using EVDSPPFM

The EVDSPPFM command in the EVIEW library can be called to display the contents of a physical file. This command differs from the operating system command DSPPFM in that it sends the output to *PRINT allowing the display of physical files using the `vp400hostcmd` interface from the EView Proxy server.

The command syntax is:

```
EVIEW/EVDSPPFM FILE(library/filename) MBR(membername)
```

where:

filename Name of the physical file

library Library where ***filename*** resides

membername Member name in physical file to list. For files with only one member, the ***membername*** is the same as the ***filename***.

Using the Default Monitors

EView/400i provides eleven different monitors in the installed product, which can be used or modified to suit your needs. These monitor policies can also be used as samples for creating additional monitors needed in your environment. These monitors provided during installation are available in the “EV400” policy group in OM:

ASP Monitor

Policy name: AS400_ASPMON

Monitors Auxiliary Storage Pool (ASP) utilization on all iSeries systems that have been configured on the EView Proxy server. The agent node name and ASP number are passed in the object field and can be used to perform pattern matching to create different conditions for specific nodes and Auxiliary Storage Pools. See the default condition in the ASP Monitor policy for the format of the object field.

CPU Monitor

Policy name: AS400_CPU

Monitors CPU utilization on all iSeries systems that have been configured on the EView Proxy server.. The agent node name is passed in the object field and can be used to perform pattern matching to create different conditions for specific nodes.

Device Busy Monitor

Policy name: AS400_DEVMON

Monitor the percent device busy and number of I/Os per second of individual disk units.

To initiate device monitoring for all managed iSeries nodes, set the desired thresholds in the AS400_DEVMON measurement threshold monitor and deploy this monitor to the Operations agent on the EView Proxy server.

The object field passed by the monitor script contains the node name, Disk Unit/ASP#, and a keyword indicator for the type of value being sent. Different thresholds may be set for different managed nodes by creating additional conditions in the policy that match on the specific node name and have different threshold levels.

➤ This monitor requires that performance data collection be active on the agent (EV400_PERF1 must be set to YES).

Job Count Monitor

Policy name: AS400_JOBCNT

Monitor the number of jobs with the same job name that are active. This is a minimum threshold monitor where the threshold is set for the minimum number of jobs of the same name that should be active.

To monitor the count of active jobs, the job must be entered in the job monitor configuration with the "Job Count" keyword selected.

The object field passed by the monitor script contains the node name, job name, subsystem name and the keyword "ACTIVECNT". Different thresholds for different jobs may be set by creating additional conditions in the policy that match the job name in the object parameter.

Active Job Monitor

Policy name: AS400_JOBMON

Monitors for subsystems and jobs which should be active. The jobs/subsystems to be monitored are specified using "iSeries Job Monitors" configuration screen in the EView/400i configuration web interface. (See "Phase 4: Identify Active Jobs to Monitor" on page 33.) This monitor can also monitor for other job or subsystem conditions by selecting different keywords in the configuration. The following keywords can be selected:

Active Job – if the named job (or any job in the named subsystem) is inactive, an alert will be generated.

MSGW in Subsystem – if any jobs in the subsystem are in a message wait an alert will be generated.

MSGW on Job – if the job is in a message wait an alert will be generated.

LCKW on Job – if the job is in a lock wait, an alert will be generated.

Optionally the time frame in which the monitoring is to occur for each job can be specified in the configuration. It is also possible to specify that specific jobs or subsystems should be monitored for message wait conditions or the total number of jobs in a message wait condition. The object field will contain the iSeries node name and other data that is specific to the type of condition that is being reported to the monitor. See the default message conditions in the policy for the format of the object field for different conditions.

Job Queue Monitor

Policy name: AS400_JOBQMON

Monitors number of jobs in job queues. To initiate job queue monitor for an iSeries managed node, enter the name of the managed node using the "iSeries Job Queues" configuration screen in the EView/400i configurator web interface. Each node to be monitored should also specify a minimum threshold for the number of jobs, which will cause the job queue to be selected for monitoring and the value sent to the monitor. The agent node name, Job Queue name and Library name are passed in the object field and can be used to perform pattern matching to create conditions for specific nodes and job queues. See the default condition in the monitor policy for the format of the object field.

Library Object Monitor

Policy name: AS400_LIBMON

Monitors the number of objects in a library and the total size of the objects in a library.

Use the “Library Monitors” tab on the EView/400i Configurator web page to identify which libraries are to be monitored. Regular expressions are allowed to identify multiple library names.

Memory Pool Monitor

Policy name: AS400_MEMMON

Monitor the number of database faults and non-database faults in each memory pool and the number of wait-ineligible transitions in each memory pool. Also monitor the total number of database and non-database faults for all memory pools combined.

To initiate memory pool monitoring for all managed iSeries nodes, set the desired threshold in the AS400_MEMMON measurement threshold policy and deploy this policy to the HP Operations agent on the EView Proxy server.

The object field passed by the monitor script contains the node name, pool name or *ALL for total faults and a keyword indicator of the type of value being sent. Different thresholds may be set for different managed nodes by creating additional conditions in the policy that match on the specific node name and have different threshold levels.

MQ Series Monitor

Policy name: AS400_MQMON

Monitor MQ Series channel status and queue depth.

Before deploying this monitor, the MQ Series commands to display the channels and queues to be monitored must be entered into members of the file QMQSC file in the EVIEW library on the iSeries system where MQ is active. See sample members in the EVIEW/QMQSC file. In the AS400_MQMON monitor policy definition, the Program field must be modified, adding three parameters to specify (1) the hostname of the iSeries system, (2) the member name in the QMQSC file containing the command(s) for channels to be monitored, and (3) the member name containing the command(s) for queues to be monitored.

If MQ Series managers on more than one iSeries agent are to be monitored, then another copy of the monitor policy must be created with a unique name and the appropriate changes made to the Program field parameters.

Output Queue Monitor

Policy name: AS400_OUTQMON

Monitors number of spool files in output queues. To initiate output queue monitoring for a managed node, enter the name of the managed node using the “iSeries Output Queues” configuration tab in the EView/400i configurator web interface. Each entry will contain the name of the node and a minimum threshold number of spool files, which will cause the output queues of that node to be selected for monitoring and the number of spool files sent to the monitor. The node name, output queue, and its library name are passed in the object field and can be used to perform pattern matching to create conditions for specific nodes and output queues. See the default condition in the monitor policy for the format of the object field.

TCP/IP Port Monitor

Policy name: AS400_PORTMON

Monitors all defined AS/400 agents for TCP/IP port activity.

This monitor scans all TCP/IP connections on an AS/400 agent and counts the number of established connections exist on each port. An alert is generated if the number of connections exceeds the defined policy threshold (default of five connections).

Agent Health Check Monitoring

Policy names: EV400_HCMON, EV400_HEALTHCHK_SCHED and EV400 OS400 Messages

Two policies are provided to perform a health check function of the EView/400 agent and proxy server functions. The EV400_HEALTHCHK_SCHED policy will send a request to the agent to execute a command that generates and EView/400 health check message to the system operator queue (QSYSOPR). The message queue filter must contain an entry for the message EVM0400 (or EVM*) to cause the agent to forward this message to the EView/400 proxy server.

When the message is received at the proxy server, the message condition for the EVM0400 message will run an automatic action to create a health check file with the time stamp of when the message is received. The EView supplied “EV400 OS400 Messages” policy contains a condition for the EVM0400 message.

The EV400_HCMON policy when deployed will check the time stamp in all of the health check files and send the difference between the current time and the health check time stamp in seconds. The default condition in the EV400_HCMON policy will compare the monitor value to a threshold and will generate an alert if the time difference is greater than the threshold. The default threshold is 900 seconds (15 minutes) which is also the default interval for the EV400_HEALTHCHK_SCHED policy and the EV400_HCMON policy. If the default interval is changed it should be changed in all health check policies and threshold conditions.

AS/400 Severity Mapping

Mapping of the AS/400 Severity Code (00-99) to OM severities is controlled by the file `/etc/opt/OV/share/conf/vp400/as400sev_map.conf` on Linux or the `as400sev_map.conf` file in the `conf` directory of the EView/400 installation path on Windows. This file maps the iSeries severity codes (in ranges) to one of five OM severities. The default assignments are:

Table 4-2: iSeries Severity Mapping

Severity Code	OVO Severity
00-20	Normal
21-40	Warning
41-60	Minor
61-80	Major
81-99	Critical

To change mapping, edit this file and restart the Master Message Server process for the agent(s).

Filtering Messages

Initial message filtering is performed on the iSeries agent based on the seven-character message ID. Details about message filtering are covered in the section "Configure Message ID Filters" on page 30.

Using Command Applications

The predefined applications provided in the Tool Bank operate on the entire domain (AS/400 node).

Using the EVJLSCAN Program

The EVJLSCAN program in the EVIEW library can be called to read the output job log from a completed job and scan the log for specified messages. The syntax for the program call is:

```
EVJLSCAN OUTQ(joboutq) OUTQLIB(outqlib) JOB(jobname) JOBID(jobnum)
USER(jobuser) MSGID1(id1 | *FILE) [MSGID2(id2) ...]
```

where:

joboutq	The name of the output queue where messages are written for this job
outqlib	The library where the *joboutq* resides
jobname	The name of the completed job
jobnum	The job ID number
jobuser	The user (owner) of the job
id1	A message ID to scan for or "*FILE"
id2 – id5	Additional messages IDs to scan for

If the *id1* value is "`*FILE`", EVJLSCAN will look for a list of message ID criteria which is created by the AS/400 Job Log Filter application, which allows for more than five message IDs to search for and check for regular expressions in the text of the messages. See "Phase 3: Identify Job Log Messages" on page 32 for information on how to set up a job log filter criteria file, and use ***jobname*** as the name when setting up a job log criteria file.

This program will normally be called as part of an automatic action after a CPF1164 message arrives announcing the completion of a job. In this case, the ***jobname***, ***jobnum***, and *jobuser* values can be retrieved from the message text.

Any messages found by the EVJLSCAN will be sent to the standard output stream.Using the EView/400i Agent Interface to System APIs

The EVCCTLPROC job of the EView/400i agent provides a direct interface for retrieving operating system information through the use of system APIs without the need to execute OS/400 commands. The output of information retrieved in this manner is presented in a format that can be parsed by a script on the OM management server. Access to this API interface is requested using destination 86 of the ev400hostcmd utility program. The type of data requested is specified by a two-digit code followed by a vertical bar (|) and additional parameter information depending on the selected code. The syntax of the ev400hostcmd with destination 86 is:

```
ev400hostcmd 86 XX[|parameters].<iSeriesHostname>
```

Keep in mind that if this command is entered on a Unix command line or script, the vertical bar(s) 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 if applicable, for example, a line representing each job. The last line will be the text “EOF”.

The available codes are:

01 Active Job Listing

Retrieves a list of active jobs.

Parameters: None

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

Job name
User name
Job Number
Internal Job ID (in printable hexadecimal)
Job Status (e.g. MSGW, DEQW, TIMW, EVTW, etc.)
Job Type (A=autostart, B=batch, I=interactive, M=subsystem monitor, R=spooled
  reader, S=system, W=spooled writer, X=start-control-program-function
  system job)
Subsystem Name
Run Priority
Function Type (the function that the job’s initial thread is currently performing:
  C=interactive command is running, D=job delayed, G=job suspended by
TFRGRPJOB
  command, I=an index is being rebuilt, L=history information is being logged,
M=job is a
  multiple requester terminal job, N=job is at a system menu, O=job is
performing I/O
  operations to a work station, P=a program is running, R=a procedure is
running, *=see
  Function Name field for current action, (space)=the system is not doing a
logged function)
Function Name

Sample Output:

```
QTFTP00349|QTCP      |043769|0050000100007B0087B67F222949C906|DEQW|B|QSYSWRK  |25| |
EVSBS     |QSYS      |043983|0050000100009F0087CA970D21D49A70|DEQW|M|EVSBS    |0| |
EVSCMDPROC|EVUSER    |044053|005000010000AB0087D1AA0A8F3185A6|DEQW|B|EVSBS    |10|P|EVC050
QPADEV0005|CHIP      |044054|005000010000AF0087D201903D4598D0|DSPW|I|QBASE    |20|C|STRSEU
EVACMDPROC|EVUSER    |043984|005000010000CB0087CA970D4BB7DBD2|SELW|B|EVSBS    |10|P|EVCCMD
```

02 Job Attributes

Retrieves additional attributes of the requested job.

Parameters: Internal Job ID in printable hexadecimal (available from the fourth field of the “01” active jobs output)

Output: One line with the following format:

System Pool ID from which the job’s main storage is allocated
CPU Time Used (in milliseconds)
Auxiliary IO Requests
Number of Interactive Transactions
Total Response Time for initial thread (in milliseconds)
Active Thread Count (including threads suspended or waiting for a resource)
Date Entered System in format CYYMMDDHHMMSS, where:
C – century, 0 indicates years 19xx and 1 indicates years 20xx.
YY – Year
MM- Month
DD – Day
HH – Hour
MM – Minutes
SS – Seconds
Date Job Active in format CYYMMDDHHMMSS
Job Description Name
Job Description Library
Submitter Job Name
Submitter User Name

Sample Output:

```
2|70793|23690|0|0|1|1040511011518|1040511011518|EVJOBD      |EVIEW   |EVJOBD  |EVUSER
```

03 Job Activation Groups and Attributes

Returns a list of all the activation groups that are associated with a given job and their attributes.

Parameters: Internal Job ID in printable hexadecimal (available from the fourth field of the “01” active jobs output)

Output: One line for each activation group, in the format:

Activation Group Name
Activation group Number
Number of Activations
Number of Heaps
Static Size (total amount of static storage allocated to the activation group, in bytes)
Heap Size (total amount of heap storage allocated to the activation group, in bytes)
Root Program Name (blanks if this is the default activation group)
Root Program Library (blanks if this is the default activation group)
Root Program Type (0=*PGM, 1=*SRVPGM, 2=Java)
Activation Group State (0=User, 1=System)
Shared Activation Group Indication (0=not shared, 1=shared)
In-use Indicator (0=not in use, 1=in use)

04 List Subsystems

Returns a list of all active subsystems.

Parameters: None

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

Subsystem Name
Library in which subsystem's description resides
Maximum number of active jobs allowed in subsystem, or -1 if no maximum
Number of jobs currently active in the subsystem
Subsystem description text

Sample Output:

```
EVSBS     |EVIEW     |-1|10|EVIEW/400I VERSION 7.0
QBASE     |QSYS      |-1|11|BASIC CONTROLLING SYSTEM
QHTTPSVR  |QHTTPSVR  |-1|8|HTTP SERVER SUBSYSTEM
QSERVER   |QSYS      |-1|13|FILE SERVER SUBSYSTEM
QSPL      |QSYS      |-1|1|SPOOLING SUBSYSTEM
QSYSWRK   |QSYS      |-1|81|SYSTEM SUBSYSTEM
QUSRWRK   |QSYS      |-1|13|USER SUBSYSTEM
EOF
```

07 Job Queue Request

Returns a list of job queues.

Parameters: None

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

Job Queue Name
Job Queue Library
Number of Jobs in Job Queue
Subsystem Name
Job Queue Status

Sample Output:

```
EVJOBQ         |EVIEW   |0|EVSBS          |RELEASED
EVJOBQ         |EVREL2  |0|               |RELEASED
NTMAINQ        |MR1     |0|               |RELEASED
NTMAINQ        |MR1PT1  |0|               |RELEASED
EVJOBQ         |MR2_P41 |0|               |RELEASED
EVJOBQ         |OVREL   |0|               |RELEASED
Q1ABRMNET      |QBRM    |0|               |RELEASED
QDMT           |QDMT    |0|               |RELEASED
QDNNOTIFY       |QDMT   |0|               |RELEASED
QPGMR          |QDMT    |0|               |RELEASED
QBASE          |QGPL    |0|QBASE          |RELEASED
QBATCH         |QGPL    |0|QBASE          |RELEASED
```

08 Output Queue Request

Returns a list of output queues

Parameters: None

Output: One line for each output queue, in the format:

Output Queue Name
Output Queue Library
Number of Files
Name of Writer Job for this Queue (blanks if no writer job started)
Output Queue Status
Printer Device Name (blanks if no writer job started for this queue)

Sample Output:

```
EVCMD     |EVIEW     |2|          |RELEASED  |
EVCMD     |EVREL2    |0|          |RELEASED  |
NETTECH   |MR1       |0|          |RELEASED  |
NTCMD     |MR1       |0|          |RELEASED  |
EVCMD     |OVREL     |0|          |RELEASED  |
QDMT      |QDMT      |0|          |RELEASED  |
QDKT      |QGPL      |0|          |RELEASED  |
QPRINT    |QGPL      |164|          |RELEASED  |
QPRINTS   |QGPL      |0|          |RELEASED  |
QPRINT2   |QGPL      |0|          |RELEASED  |
QSPRCLOUTQ|QRCL      |0|          |RELEASED  |
```

09 Output Queue Listing

Returns a list of jobs in the specified output queue

Parameters: (separated by |)

Output queue name
Output queue library

Output: One line for each spool file in the queue in the format:

Job Name
User Name
Job Number
Spooled File Name
Spooled File Number
Spooled File Status
Spooled File Open Date and Time: CYYMMDDHHMMSS
User Data
Form Type
Total Pages
Number of Copies
Priority

Sample Output:

```
QPADEV0007|BRYAN     |020755|QPDSPLOG  |1|*READY    |1020913090127|          |*STD      |2|1| 5
QPADEV0007|BRYAN     |020755|QPDSPLOG  |2|*READY    |1020913112248|          |*STD      |1|1| 5
EVINIT    |BRYAN     |037258|EVINIT    |1|*READY    |1030806143553|          |*STD      |5|1| 5
QPADEV0005|BRYAN     |037284|EVCMSGM   |1|*READY    |1030807111424|CRTCMOD   |*STD      |40|1| 5
QPADEV0005|BRYAN     |037284|EVCMSGM   |5|*READY    |1030807140745|CRTCMOD   |*STD      |37|1| 5
EVINIT    |BRYAN     |037306|EVINIT    |1|*READY    |1030807141530|          |*STD      |5|1| 5
QPADEV0006|BRYAN     |037411|EVCCMDM   |6|*READY    |1030811140323|CRTCMOD   |*STD      |19|1| 5
QPADEV0004|BRYAN     |040417|QPSUPRTF  |1|*READY    |1031211164959|          |*STD      |1|1| 5
QPADEV0004|BRYAN     |040417|QPSUPRTF  |2|*READY    |1031211164959|          |*STD      |4|1| 5
QPADEV0004|BRYAN     |040973|QPDSPLIB  |1|*READY    |1040224160919|          |*STD      |3|1| 5
QPRTJOB   |QSECOFR   |015605|EVC020    |15|*READY    |1040429075548|          |*STD      |3|1| 5
```

11 Directory Listing

Returns a list of files from an AS/400 IFS directory

Parameter:

IFS directory name

Output: One line for each file found

Sample Output:

```
.
..
preserve
tmp
EOF
```

16 Job Queue Listing

Returns a list of jobs in the specified job queue

Parameters: (separated by |)

Job Queue Name
Job Queue Library

Output: One line for each job in the queue in the format: Job Name

Job Name
User Name
Job Number
Job Type
Job Queue Priority
Submitter Job Name
Submitter User Name
Submitter Job Number
Status on Job Queue
Date and Time Job Entered System: CYYMMDDHHMMSS

Sample Output:

```
STATJOB   |QSECOFR   |018324|B| 5|QPADEV0003|QSECOFR   |018210|HLD       |1090817151919
EOF
```

20 System Statistics

Returns one line of system statistics

Parameters: None

Output: One line in the following format:

Number of Users Currently Signed On (not incl. system request and group jobs)
Batch Jobs Waiting
Batch Jobs Running
Batch Jobs Held
Number of Batch Jobs Held on a Job Queue (jobs submitted but held)
Number of Batch Jobs on a Held Job Queue (assigned to a subsystem but held)
Total number of user jobs and system jobs currently in the system

Percentage of Maximum Addresses for Permanent Objects in Use (in thousandths)
Percent of Maximum Addresses for Temporary Objects in Use (in thousandths)
System ASP (in MB)
System ASP Utilization (in decimal percentage, the sample below is 49.8126%)
Total Auxiliary Storage on the System (in MB)
Current Unprotected Storage Used for Temporary Objects (in MB)
Maximum Unprotected Storage Used for Temporary Objects since Last IPL (in MB)
Reserved (coded "0")
Reserved (coded "0")
System Percent CPU (divide by 10 to get utilization in tenths)
Start of Statistics Interval (seconds since 1/1/1970)
End of Statistics Interval (seconds since 1/1/1970)

Sample Output:

```
1|0|48|0|0|0|142|13|64|8590|498126|8590|534|789|0|0|16|1084276165|1084276225
```

21 System Pools Request

Returns line for each memory pool

Parameters: None

Output: One line for each pool in the following format:

Pool Number
Pool Size (in KB)
Reserved Size (in KB)
Maximum Active Threads that can be Active in the Pool
Rate of Database Faults (in tenths : 123 = 12.3 page faults per second)
Rate of Database Pages brought into the Pool (in tenths : 123 = 12.3 pages per second)
Rate of Non-database Page Faults (in tenths : 123 = 12.3 page faults per second)
Rate of Non-database Pages brought into the Pool (in tenths : 123 = 12.3 pages per second)
Rate of Transitions from Active to Wait (in tenths: 123 = 12.3 transitions per minute)
Rate of Transistions from Wait to Ineligible Condition (in tenths)
Rate of Transitions from Active to Ineligible Condition (in tenths)
Pool Name
Subsystem Name
Subsystem Library Name
Paging Option

Sample Output:

```
1|58804|32868|32767|0|0|0|0|137|0|0|*MACHINE |            |                |*FIXED
2|189268|8|52|0|0|0|0|654|0|0|*BASE      |        |                |*FIXED
3|13104|0|10|0|0|0|0|0|0|0|*INTERACT |   |                |*FIXED
4|968|0|1|0|0|0|0|0|0|0|*SPOOL |         |        |*FIXED
```

22 System CPU Utilization

Returns one line with current system CPU utilization

Parameters: None

Output: 1 line in the following format:

Percent CPU utilization (in tenths, the sample below is 1.9%)
Statistics interval start time (seconds since 1/1/1970)
Statistics interval end time (seconds since 1/1/1970)

Sample Output:

```
19|1084276285|1084276345
```

27 Active Job Log

Returns the last 500 lines from the active job log of the specified job.

Parameters: (separated by vertical bar)

Internal Job ID (available from the fourth field of the “01” active jobs output)
Number of lines of job log output to display (the most recent output is returned).

Output: Job log output in the following format:

Message ID
Message Type
Message Severity
Message Time
Sending Program
Sending Text

Sample Output:

```
CPF2447|15|40|1050107144314|QMHLDISP  |NO ENTRIES EXIST IN CURRENT VERSION
OF LOG.

       |10|00|1050107144314|QCADRV    |  8700 - DLYJOB DLY(5)

       |10|00|1050107144319|QCLCLCPR  |  5300 - CALL PGM(EVIEW/EVHTIME)
/* THE CALL COMMAND CONTAINS PARAMETERS */

       |10|00|1050107144319|QCADRV    |  5600 - DSPLOG PERIOD((144314
010705) (144319 010705)) OUTPUT(*PRTWRAP)

CPF2447|15|40|1050107144319|QMHLDISP  |NO ENTRIES EXIST IN CURRENT VERSION
OF LOG.

       |10|00|1050107144319|QCADRV    |  8700 - DLYJOB DLY(5)

       |10|00|1050107144324|QCLCLCPR  |  5300 - CALL PGM(EVIEW/EVHTIME)
/* THE CALL COMMAND CONTAINS PARAMETERS */

       |10|00|1050107144324|QCADRV    |  5600 - DSPLOG PERIOD((144319
010705) (144324 010705)) OUTPUT(*PRTWRAP)

CPF2447|15|40|1050107144324|QMHLDISP  |NO ENTRIES EXIST IN CURRENT VERSION
OF LOG.

       |10|00|1050107144324|QCADRV    |  8700 - DLYJOB DLY(5)

       |10|00|1050107144329|QCLCLCPR  |  5300 - CALL PGM(EVIEW/EVHTIME)
/* THE CALL COMMAND CONTAINS PARAMETERS */

       |10|00|1050107144329|QCADRV    |  5600 - DSPLOG PERIOD((144324
010705) (144329 010705)) OUTPUT(*PRTWRAP)

CPF2447|15|40|1050107144329|QMHLDISP  |NO ENTRIES EXIST IN CURRENT VERSION
OF LOG.

       |10|00|1050107144329|QCADRV    |  8700 - DLYJOB DLY(5)
       |10|00|1050107144334|QCLCLCPR   |  5300 - CALL PGM(EVIEW/EVHTIME)
/* THE CALL COMMAND CONTAINS PARAMETERS */

       |10|00|1050107144334|QCADRV     |  5600 - DSPLOG PERIOD((144329
010705) (144334 010705)) OUTPUT(*PRTWRAP)
```

28 QSYSOPR Inquiry Messages

Returns outstanding (messages needing reply) inquiry messages from QSYSOPR.

Parameters: None

Output:

Message ID
Message Key
Fully qualified job name (Jobname/User/JobID)
Message Queue Name (QSYSOPR)
Time Stamp of Message (CYYMMDDHHMMSS)
Message Text

There are two lines of output for each outstanding message. The second line contains any message text that exceeds the output length of the first line.

Sample Output:

```
CPA4086|30128|QPADEV0005KENNY     004701|QSYSOPR   |1041223113920|DEVICE TAP01 WAS NOT
READY OR NEXT VOLUME WAS NOT LOADED.  (C R)


OPT1486|30416|QPADEV0003KENNY     004699|QSYSOPR   |1041223114031|LOAD NEXT VOLUME ON
OPTICAL DEVICE OPT01. (C G)


BJW0006|31648|QPADEV0006KENNY     004704|QSYSOPR   |1041223165127|Aren't you a little
short for a stormtrooper? (Y N)


BJW0007|31808|QPADEV0006KENNY     004704|QSYSOPR   |1041223165151|What good are snub
fighters going to be against that? (A B C D E F)
```

30 Data Queue Information

Returns information about the specified data queue.

Parameters:

Data queue Name
Data queue Library

Output:

Data Queue Name
Data Queue Library
Message Length
Key Length
Sequence
Include Sender ID
Force Indicators
Type
Automatic Reclaim
Number of Messages
Maximum Number of Messages
Maximum Entries Allowed
Initial Number of Entries

Sample Output:

```
EVSENDQ   |EVIEW     |287|0|F|N|N|0|0|0|208|55184|16
```

31 ASP Statistics

Returns statistics on Auxiliary Storage Pools.

Parameters: None

Output: One line is returned for each ASP found (up to 20) in the following format:

ASP Number, a unique identifier for each pool
Number of Disk Units in the ASP (mirrored units are counted as one)
ASP Capacity Total (in MB)
ASP Capacity Available Total (in MB)
ASP Capacity Protected by mirroring or device parity (in MB)
ASP Capacity Available Protected by mirroring or device parity (in MB)
ASP Capacity Unprotected by mirroring or device parity (in MB)
ASP Capacity Available Unprotected by mirroring or device parity (in MB)
ASP System Storage currently allocated in the ASP for system use (in MB)
Overflow Storage – Number (in MB) overflowed from user ASP to system ASP
Auxiliary Storage Space Allocated to Error Log (in MB)
Auxiliary Storage Space Allocated to the Machine Log (in MB)
Auxiliary Storage Space Allocated to the Machine Trace (in MB)
Auxiliary Storage Space Allocated for Main Storage Dump Space (in MB)
Auxiliary Storage Space Allocated to the Microcode (in MB)
Storage Threshold Percentage (message sent to QSYSOPR when reached)
ASP type:
00 – System ASP
10 – User ASP that does not contain libraries
11 – User ASP that does contain libraries

Sample Output:

```
1|1|17549|5827|0|0|17549|5827|3|0|1|55|1|139|938|90|00
```

32 Disk Performance Statistics

Returns statistics on individual disk units. Performance data collection option 1 (EV400_PERF1 parameter) must be set to YES to use this option.

Parameters:

None

Output:

Disk serial number
ASP Number
Unit Number
I/Os per second (in tenths)
Reads per second (in tenths)
Writes per second (in tenths)
Disk Busy (in tenths)

Unit Control

0 There is no unit control value.
1 The disk unit is active.
2 The disk unit has failed.
3 Some other disk unit in the disk subsystem has failed.
4 There is a hardware failure within the disk subsystem that affects performance, but does not affect the function of the disk unit.
5 There is a hardware failure within the disk subsystem that does not affect the function or performance of the disk unit.
6 The disk unit's parity protection is being rebuilt.
7 The disk unit is not ready.
8 The disk unit is write protected.
9 The disk unit is busy.
10 The disk unit is not operational.
11 The disk unit has returned a status that is not recognizable by the system.
12 The disk unit cannot be accessed.
13 The disk unit is read/write protected.

Mirror Unit Protection

0 One mirrored unit of a mirrored pair is not active
1 Both units of a mirrored pair are active

Mirror Unit Reported

0 Mirrored unit is missing, information returned may not be current
1 Mirrored unit reported, information is current

Mirror Unit Status

1 Active
2 Mirrored unit being synchronized
3 Mirrored unit suspended

Compression Status

0 No compression
1 Compression active

Disk Protection Type

0 No protection
1 Mirrored
2 Part of parity protection array

Sample Output:

```
68-0DD1BE0|1|1|1|7|16|16|1|0|0|0|0|0
```

33 ASP Status

Returns statistics on Auxiliary Storage Pools.

Parameters: None

Output: One line is returned for each ASP in the following format:

ASP Number, a unique identifier for each pool
Unique system-assigned name of the disk unit (up to 10 characters)
Name of device description that activated the ASP (independent ASPs only, blanks otherwise)
Version of objects in an independent ASP:
0 – Objects are usable by a system at release level V5R1M0
1 – Objects are usable by a system at release level V5R2M0
2 – Objects are usable by a system at release level V5R3M0
Usage that is assigned to the ASP:
0 – N/A or not known (system ASP and basic user ASPs)
1 – User-defined file system ASP
2 – Primary ASP
3 – Secondary ASP
Device configuration status of an ASP:
0 – No status (used for the system ASP and basic user ASPs)
1 – Varyoff
2 – Varyon
3 – Active
4 – Available
Name that is assigned to the database that this ASP defines, or blanks if not a primary or
secondary ASP (up to 18 characters)

Sample Output:

```
1|          |          |0|0|0|
EOF
```

34 Disk Unit Information

Returns information about individual disk units.

Parameters: None

Output: One line for each disk unit, in the format:

ASP Number
Disk Type
Disk Model
Disk Serial Number
Resource Name: a unique system-assigned name for the disk unit
Disk Unit Number: a unique identifier for the disk unit; mirrored disks will have the same number
Capacity: total size of the disk unit, in megabytes
Storage Available in megabytes
Storage Reserved for System in megabytes
Disk Protection Type:

0 No protection
1 Mirrored
2 Part of parity protection array

Sample Output:

```
1|4326|0050|68-0DF5F29|DD001      |1|35166|25094|2|0
1|4326|0050|68-0E1494E|DD002      |2|35166|25085|2|0
EOF
```

41 List Logical Interfaces

Return information about the network interfaces

Parameter:

Interface Type (optional, default is *ALL)

Output: One line for each interface with the following format:

IP Address
Network Address
Interface Subnet Mask
Interface Name
Line Status
Interface Type
MAC Adapter Address

Sample Output:

```
127.0.0.1     |127.0.0.0    |255.0.0.0    |*LOOPBACK |ACTIVE     |NONE     |
192.168.1.113 |192.168.1.0  |255.255.255.0|ETHLINE   |ACTIVE     |ETHERNET |00096B6BBF83
EOF
```

42 Network Connection Status

Return standard Netstat information about the network connections.

Parameters:

Starting Local Port (0 to 65535, default is 0)
Ending Local Port (0 to 65535, default is 65535)
Type (TCP, UDP, IPS, or *ALL, default is *ALL)
IP Version (IPV4 or IPV6, default is IPV4)

All parameters are optional, but vertical bar placeholders must be used if earlier parameters are omitted.

Output: One line for each network connection in the format:

Remote Address
Remote Port
Local Address
Local Port
Connection Type (TCP, UDP, or IPS)
User Who Performed the Bind
Idle Time (seconds)
Bytes In
Bytes Out
Connection Status
Job(s) Using the Connection (multiple jobs separated by commas)

Sample Output:

```
0.0.0.0        |    0|0.0.0.0        | 8478|TCP|QUSER     |3769403.4|      0|       0|LISTEN   |QZDASRVSD
0.0.0.0        |    0|0.0.0.0        | 8479|TCP|QUSER     |3769405.0|      0|       0|LISTEN   |QNPSERVD
127.0.0.1      |11423|127.0.0.1      | 8002|TCP|EVUSER    |    156.1|      0|1155666783|ESTABLISH|EVSTCPPROC
127.0.0.1      |37751|127.0.0.1      | 8473|TCP|QUSER     |   6885.1|   4866|   55872|ESTABLISH|QPWFSERVSO
127.0.0.1      | 8002|127.0.0.1      |11423|TCP|EVUSER    |    156.1|1155666783|       0|ESTABLISH|EVTCTLPROC
127.0.0.1      | 8473|127.0.0.1      |37751|TCP|QSECOFR   |   6885.1|  55872|    4866|ESTABLISH|QSRVMON
192.168.0.127  |54573|192.168.0.113  |   23|TCP|QTCP      |    221.7|  42338| 1186348|ESTABLISH|QPADEV000G,QTVDEVICE
0.0.0.0        |    0|192.168.0.113  |  427|TCP|QSYS      |3769414.6|      0|       0|LISTEN   |QSLPSVR
0.0.0.0        |    0|192.168.0.113  | 4800|TCP|QSYS      |3769414.6|      0|       0|LISTEN   |QSLPSVR
192.168.0.81   |49161|192.168.0.113  | 8000|TCP|EVUSER    |      5.8| 668488|296622990|ESTABLISH|EVTCTLPROC
192.168.0.98   | 4655|192.168.0.113  | 8000|TCP|EVUSER    |      5.8| 692072| 8416528|ESTABLISH|EVTCTLPROC
192.168.0.170  |54482|192.168.0.113  | 8000|TCP|EVUSER    |      5.8| 733336|2942542|ESTABLISH|EVTCTLPROC
0              |    0|0.0.0.0        |  138|UDP|QSYS      |     95.3|60298356| 2255330|UNKNOWN  |
0              |    0|0.0.0.0        |  427|UDP|QSYS      |3769420.2|      0|       0|UNKNOWN  |
0              |    0|0.0.0.0        |  427|UDP|QSYS      |3769420.7|      0|       0|UNKNOWN  |
0              |    0|192.168.0.113  |  427|UDP|QSYS      |3769420.7|      0|       0|UNKNOWN  |
EOF
```

Coordinating Multiple OM Management Servers

The AS/400 agent has the capability to communicate with multiple OM management servers. The agent may be configured to send messages to all connected OM management servers by setting the agent configuration EV400_MSG_DISTRIB option to “YES” in the Node Configurator (see “Add iSeries Nodes and Runtime Parameters” on page 16).

The AS/400 agent also may be configured to send messages only to a designated primary management server. In this case, the EV400_MSG_DISTRIB option must be set to “NO” and the host name or IP address of the primary OM server must be entered in the EV400_PRIMARY_SERVER field.

When operating with multiple EView/400 proxy servers and a designated primary proxy server, the server command vp400ragt (ev400ragt on Windows) may be used to communicate temporary changes in the primary management server to the AS/400 agent. The command may be issued from a command prompt or may be configured as a tool for execution by an OM operator. The syntax of the command is:

vp400ragt -node *as400_name* -primmgr [*server*]

where:

`as400_name` The fully qualified hostname of the AS/400 agent.

`server` The OM server hostname or IP address that will become the designated primary server. If not specified, the server on which the command was entered is the designate primary server.

When using the vp400ragt command to communicate primary manager changes to the agent, the following conditions are checked by the agent:

- If the primary manager designated in the agent configuration file is connect and is in the primary role, the vp400ragt request must come from that primary manager.
- The request can come from a secondary OM management server only if the designated primary is not connected or the secondary OM server is currently in the primary role.
- The server to assume the primary role must already be connected to the agent (i.e., the management processes must have been started via the EView/400i Node Configurator).

Setting the EV400_MSG_DISTRIB parameter to "Y" may be useful in environments where it is desired to maintain a "hot backup" OM server that is always receiving the same messages as the primary OM server.

When sending the same message to multiple OM servers, care must be taken with message policy conditions that execute automatic actions, as it will be possible to duplicate the execution of automatic actions that are configured on both OM servers.

Another possibility when setting the EV400_MSG_DISTRIB parameter to "YES" is to control connectivity to the EView/400i agent via starting and stopping the server connection processes. The EView/400i server processes can be started and stopped using the command line utility vp400sv. The command syntax is:

```
vp400sv [-start|-stop] as400_full_qualified_name
```

In the case where more control is desired over the distribution of messages to a primary or backup OM server without having to start of stop the EView/400 server processes, the EV400_MSG_DISTRIB parameter should be set to "NO". In this case, the EV400_PRIMARY_SERVER parameter must be set to the hostname or IP address of the EView Proxy server that is designated as the primary server. In this scenario, EView/400 server processes may be active on all OM servers, however, the EView/400 agent will only send messages to the server designated as the primary when the primary is active and connected to the agent. To change the designated primary while the EView/400 agent is active, use the `vp400ragt` command line utility to instruct the agent as to which OM server is to act as the primary server.

5

Troubleshooting EView/400i

This chapter describes how to troubleshoot problems with EView/400i.

General Troubleshooting

Before you troubleshoot a particular problem you run into when installing, configuring, or using EView/400i, you should verify that your EView/400i environment is correctly installed and configured.

Correct installation and configuration of EView/400i ensures, among other things, that messages are processed correctly:

- **Message Generation**

  Messages are generated by the EView/400i system.

- **Message Interception**

  Messages are intercepted by the EView/400i policies and monitors.

- **Message Browser**

  Messages appear in the OM Message Browser in the form you expect.

Use EVSTATUS Command to Verify Status of iSeries Agent

On the iSeries agent, use the command EVIEW/EVSTATUS to collect the status of the several components of the EView/400i agent and their interaction with the iSeries system. The command is called from an iSeries terminal. The format is:

```
EVIEW/EVSTATUS [PARM('options')] [OUTPUT(outoption)]
```

where:

options One or more of the following, separated by spaces:

VER	EView/400i version information
CONF	Current distributed configuration files
JOBS	Status of jobs in the EVSBS subsystem
TCP	Defined TCP/IP ports and current status
DQS	Data queues status
AUD	System QAUDLVL vs. EView/400i audit options
USP	Defined user spaces
SYS	AS/400 system information
ALL	All of the above (Default)
? or **HELP**	Display help options

outoption One of:

*****	For output to a terminal
***PRINT**	For output to the user's print queue (Default)

Example call:

```
EVIEW/EVSTATUS PARM('JOBS TCP SYS') OUTPUT(*)
```

Browse the output text of this command and look for "NOTE" or "WARNING" messages that may indicate how to resolve outstanding problems. Retain a copy of the output for possible transmission to EView Technology support personnel.

Use EVTRACING Command to Dynamically Activate Debug Tracing

On the iSeries agent, use the EVTRACING command to start and stop debug tracing without restarting an EView/400i job. Use this tracing facility at the direction of EView Technology support personnel. The trace output will be written to the EVIEW/EVTRACE output queue. The syntax is:

```
CALL EVIEW/EVTRACING [PARM('jobname' '[x]tracelevel')]
```

where:

jobname The name of the job running under the EVSBS subsystem. The valid ***jobname*** values for EVTRACING are:

EVTCTLPROC
EVSTCPPROC
EVACMDPROC
EVPERFPROC
EVCCTLPROC
EVMSGQMON
EVSCMDPROC
EVSHSTPROC
EVSRSCPROC

tracelevel A decimal value 0-65535 (or hexadecimal 0-FFFF if preceded by an "x") indicating the trace level. A ***tracelevel*** of "0" turns off job tracing.

Issuing the **CALL EVIEW/EVTRACING** command with no parameters displays the current trace level for each job. These values can also be set before the EVSBS subsystem starts by editing the appropriate "_TRACE" parameter value in the EVIEW/EVPARMS parameter file.

Specific Troubleshooting

This section explains how to solve specific problems you may encounter when using EView/400i.

Verifying Connectivity and Agent Operation

When trouble shooting problems with the operation of the EView/400i product it is important to verify the correct operation of the server components and the agent processes. The following steps should be performed to verify correct operation.

On the EView proxy server:

- Issue the command:

vp400sv -status <*as400nodename*> (Linux)

ev400sv –status <as400nodename> (Windows)

Verify that all processes are running

- Check the status of the TCP/IP ports used to connect to the agent. For example, if the default ports are used, issue the commands:

  netstat -a|grep 9000

  netstat -a|grep 9001

  For correct operation, both ports should be “Established”.

- Check EView/400i log files in the `/var/opt/OV/log/vp400` directory on Linux or the `log` subdirectory of the EView/400 installation path on Windows for any error messages.

On the iSeries agent node:

- Issue the command:

  WRKACTJOB SBS(EVSBS)

  The following display shows the minimum number of jobs that should be running and the typical job status:

```
EVSBS              QSYS        SBS      .0                    DEQW
     EVACMDPROC    EVUSER      BCH      .0   PGM-EVCCMD       SELW
     EVCCTLPROC    EVUSER      BCH      .0   PGM-EVCCTL       DEQW
     EVSCMDPROC    EVUSER      BCH      .0   PGM-EVC050       DEQW
     EVSMSGPROC    EVUSER      BCH      .0   PGM-EVC010       DEQW
     EVSTCPPROC    EVUSER      BCH      .0   PGM-EVCHCI       SELW
     EVTCTLPROC    EVUSER      BCH      .0   PGM-EVCMSG       SELW
```

  Depending on the agent configuration and options selected there may be up to eleven jobs running as part of the agent subsystem. This list represents the minimum jobs that must be running for basic operation of the agent.

- Check the agent message queue for any error messages that may have been issued with the command:

  DSPMSG EVIEW/EVLOGQ

- Check the agent trace files for any output that may have generated. The trace files are in the EVTRACE output queue in the EVIEW library.

- Check the status of the TCP ports used by the agent using the command:

  NETSTAT *CNN

  Using Function Key 15 (F15) it is possible to limit the NESTAT output to the range of ports configured for the agent. The ports configured in parameters EV400_AS400_MSG_PORT and EV400_AS400_CMD_PORT should show “Established” if the OVO management server processes are connected. It is also normal for these two ports to also be in a “Listen” state. The port configured in parameter EV400_AS400_SERVER_PORT must show “Established” before any messages or command responses can be sent to the OM management server.

- Check the condition of the agent data queues. The agent uses several data queues to store requests and messages. Data queue objects may become damaged due to unexpected interruption or system errors. If this occurs this can cause agent jobs to fail. To check the data queues issue the following commands:

  ```
  ADDLIBLE EVIEW

  DDQ EVIEW/EVSENDQ

  DDQ EVIEW/EVAPIQ

  DDQ EVIEW/EVCMDQ

  DDQ EVIEW/EVMRSPQ
  ```

  If any data queues have been damaged an exception message will be generated when issuing the command. If the data queue properties are displayed check to be sure the maximum entry length is not zero. A zero length in this field is an indication of a damaged data queue.

If No Messages are on the OMi Management Server

Symptom

No iSeries messages are arriving on the OM management server.

Solution

1. Verify that the connection between the OM management server and EView/400i is up and running by checking the status of the processes using the Add/Edit node screen of the EView/400 configuration interface.
2. Verify that the MICRO FOCUS Operations agent has been correctly installed and configured on the EView/400 proxy server.
3. Verify that the OM agent processes (in particular, the control agent) are running.
4. Verify that the iSeries policies have been correctly assigned and distributed to the EView/400 proxy server.
5. Check filter file that has been distributed to the agent to verify the correct message IDs are in the filter.
6. Check message queue setup on the AS/400 agent to verify that the message queue is configured correctly. For example, check the severity setting on the message queue to ensure that it is set to a severity which will allow all wanted messages to be received.

A

iSeries Agent Messages

This appendix describes all messages generated by the EView/400i jobs running on the iSeries agent system.

EView/400i AS/400 Messages

Table A-1: Messages

Message ID	Severity	Type	Description	System Action	User Action
EVM0001	99	Info	An invalid reply to a message	Processing continues	Contact your system administrator to reply to the message on the iSeries
EVM0007	99	Info	&1 limit reached, queue cleared. Maximum number of records were written to the queue. This is most likely due to the OVO Management Server not being connected to the EView/400 agent.	Queue is cleared. Processing continues	If this appears repeatedly contact support
EVM0015	99	Inq	***WARNING*** Processing has ended due to the possible loss of connectivity	Process stops until message is replied to	To recover enter, A=clear data queue and continue processing, B=end EVSBS subsystem and contact system administrator.
EVM0016	99	Info	&1 has been modified	&1 data file has been modified	Contact your system administrator
EVM0017	99	Info	Message queue &1 was not cleared	Selected message queue is not cleared when EVSBS subsystem was ended	Contact your system administrator
EVM4444	99	Info	Cannot allocate file &1	Restoration of default configuration	Release the lock on &1 and try your

Message ID	Severity	Type	Description	System Action	User Action
				file failed	operation again
EVM5555	99	Info	There is a lock on the EVIEW library	Upgrade installation stopped	To recover enter, I=ignore the message, C=cancel operation and contact system administrator

B

EView Proxy Server Messages

This appendix describes the messages generated on the EView Proxy Server for EView/400i.

EView/400i Management Server Messages

The following is a list of error messages from the EView/400i Proxy Server processes:

Table B-1: Socket Communication Errors

Code	Definition
EVOSOK001	%s failed calling %s, reason: %s
EVOSOK010	Unable to open %s %s socket
EVOSOK020	Unable to bind socket
EVOSOK030	Unable to set socket to non-blocking mode
EVOSOK031	Unable to set socket to blocking mode
EVOSOK040	Error on listen for socket connection
EVOSOK050	Socket connect failed, will retry momentarily
EVOSOK051	Socket connect failed, no retry will be attempted
EVOSOK070	Unable to get socket option: %s
EVOSOK071	Unable to get socket option: %s
EVOSOK080	%s failed reading MMS socket, reason: %s
EVOSOK081	Failure reading %s client UDP socket
EVOSOK082	Failure reading %s server UDP socket, number bytes returned is zero
EVOSOK083	Failure reading %s server UDP socket, entire message not sent
EVOSOK090	Failure writing to %s client UDP socket
EVOSOK092	Failure writing to %s client UDP socket, entire message not sent
EVOSOK199	Failure reading EView/Open Mainframe Message Server, reason: %s
EVOSOK200	Lost connection with EView/Open Mainframe Message Server
EVOSOK201	%s has exited due to read failure on MMS connection
EVOSOK202	%s has lost connection with the MMS
EVOSOK203	%s has exited due to loss of connection with the Command Server
EVOSOK220	MMS failed sending command response to the Command Server

Code	Definition
EVOSOK221	No TCP connection with Doman: %s

Table B-2: Management Platform API Errors

Code	Definition
EVOAPI001	%s can not make initial connection with management API
EVOAPI002	%s was not able fill symbol map
EVOAPI003	%s was not able fill status map
EVOAPI004	%s was not able to lock data base
EVOAPI005	%s failed trying to add a node to the managment platform: %d
EVOAPI006	Current OpenView map \" % s\ " is Read-Only, exiting
EVOAPI100	API error message: %s
EVOAPI101	%s lost connection with API: Exiting
EVOAPI200	No selected icon for %s
EVOAPI201	Only one symbol may be selected
EVOAPI205	Attempting to get the id for %s

Table B-3: Process Initialization Errors

Code	Definition
EVOINI000	%s initialized successfully
EVOINI001	%s initialized successfully for domain %s
EVOINI010	%s started with invalid argument count
EVOINI011	Domain name must be passed in to the %s
EVOINI012	Invalided transaction program name executable used to start %s
EVOINI013	Resource name with domain extension must be passed into the %s
EVOINI020	%s encountered invalid for configuration parameter %s
EVOINI030	%s needs the %s environment variable set properly
EVOINI040	Error setting debug value, unable to find module %s in %s
EVOINI050	%s unable to open log file: exiting

Code	Definition
EVOINI060	Required configuration parameter is missing: %s
EVOINI070	Unable to obtain memory for Status Mapping table
EVOINI071	Unable to open OV Status Map File %1$s - Reason %2$s
EVOINI072	Unable to open vpo_severity.conf file
EVOINI073	Too many entries in vpo_severity.conf file - Notify support
EVOINI074	Invalid severity range on record %1$d - Notify system administrator
EVOINI075	Warning - overlapping range in vpo_severity.conf file record - %1$d
EVOINI076	Invalid severity in vpo_severity.conf file on record %1$d

Table B-4: Process Execution Errors

Code	Definition
EVOEXE0	%s has completed without error
EVOEXE000	%s failed calling %s
EVOEXE001	%s failed calling %s with rc: %4d
EVOEXE002	%s failed calling %s, reason: %s
EVOEXE003	Failing system command: %s
EVOEXE005	%s found bad file format %s
EVOEXE006	%s found bad file format %s, line %d
EVOEXE010	%s failed to open file %s, reason : %s
EVOEXE011	%s failed to delete file %s, reason: %s
EVOEXE012	Please check file permissions
EVOEXE015	%s failed to obtain file statistics for file %s, reason: %s
EVOEXE020	Memory allocation failure, check available memory
EVOEXE030	Unable to obtain machine name
EVOEXE031	Unable to obtain host TCP/IP address from host name: %s
EVOEXE050	Invalid selection made, please select again
EVOEXE100	%s process has exited
EVOEXE102	%s process of domain %s has exited

Code	Definition
EVOEXE120	%s can not reach domain %s: exiting
EVOEXE130	%s received an unsuccessful return from%s: returning

Table B-5: EView/400i Client Errors

Code	Definition
EVOCLI000	Invalid -display option use: -display DisplayName
EVOCLI010	Client received 'message out of sequence' error from server
EVOCLI030	Invalid command form specified, valid values are 1, 2 or 3

Table B-6: Discovery Messages

Code	Definition
EVODIS05	%s has already been run. Exiting

Table B-7: Status Manager

Code	Definition
EVOSM010	PS file could not be opened: exiting
EVOSM020	Cannot store the Status Manager’s Process ID: exiting
EVOSM030	The input file %s will not open: returning
EVOSM121	Another Status Manager is already running: exiting

Table B-8: Active Status

Code	Definition
EVOID010	Act_stat could not open input file %s
EVOID015	Switched_pu could not open input file %s
EVOID020	Return from host is not correct in %s: returning
EVOID030	Cannot check status of Session IDs

Table B-9: Check Status

Code	Definition
EVOCS010	The name of the resource cannot be found on the command line: exiting
EVOCS011	A resource must be selected: exiting

Table B-10: Refresh

Code	Definition
EVOREF070	Starting a passive refresh

Table B-11: Master Message Server (MMS)

Code	Definition
EVOMMS001	TCP connection established from MMS to domain: %s
EVOMMS100	TCP connection lost from MMS to domain: %s
EVOMMS110	Agent version %1$d received

Table B-12: Command Server (CS)

Code	Definition
EVOCSR001	TCP connection established from CS to domain: %s
EVOCSR100	TCP connection lost from CS to domain: %s
EVOCSR110	Agent version %1$d received

Table B-13: ELLI

Code	Definition
EVOLLI001	vp400elli terminating'
EVOLLI010	Unable to Initialize with OpenView process manager
EVOLLI015	Invalid value for maximum reply buffer setting to default (100,000)
EVOLLI016	Maximum reply buffer too small setting to 5000
EVOLLI020	Error initializing with OpenView Operations Management Server
EVOLLI025	Unable to open directory %1$s
EVOLLI030	Unable to allocate memory for reply buffer
EVOLLI040	Reply sent to %1$s
EVOLLI045	Command %1$s sent to %2$s
EVOLLI050	Domain %1$s is not configured notify system administrator
EVOLLI060	Invalid request format for TCP request notify system administrator
EVOLLI065	Program error could not find client connection
EVOLLI066	Program error occurred notify system administrator

Code	Definition
EVOLLI100	Not connected to domain: %s
EVOLLI101	Lost connection to domain: %s
EVOLLI102	Action request queue full contact administrator
EVOLLI103	Response message exceeds buffer size
EVOLLI104	Response timeout for this action
EVOLLI105	Unable to allocate additional memory for reply
EVOLLI106	Invalid TCP Request Port specified using default
EVOLLI107	Unable to Bind TCP Request Port

Table B-14: Discovery

Code	Definition
EVODIS000	Starting discovery process for %1$s
EVODIS010	Unable to get configuration parameters for %1$s
EVODIS015	Unable to connect to OpenView database
EVODIS016	Check to make sure ovwdb process is running
EVODIS020	Discovery is unable to openan event session: %1$s
EVODIS030	...New nodes will be added to the object database
EVODIS040	...Discovery output will be saved in %1$s
EVODIS101	Unable to find status manager
EVODIS102	Error sending signal to status manager error %d
EVODIS105	Unable to add root object to database terminating
EVODIS110	Unable to add domain object to database terminating
EVODIS115	Unable to add SWITCH:CTL object database terminating
EVODIS120	Failed executing vp400hostcmd - Reason %1$s
EVODIS125	Error sending signal to status manager - error %1s
EVODIS200	Lost connection to domain: %s
EVODIS201	Action request queue full - contact administrator
EVODIS202	Response message exceeds buffer size
EVODIS202	Response timeout for this action
EVODIS203	Unable to allocate additional memory for reply

Code	Definition
EVODIS300	Discovery completed - %1$d new objects added

Table B-15: VP400 Map Messages

Code	Definition
EVOMAP100	Error on FieldNametoFieldID for field %1$s, error=%2$s
EVOMAP105	No Object found for field %1$s
EVOMAP110	Vp400map abnormal end
EVOMAP115	send status_event:Unable to create PDU (%1$d)
EVOMAP120	send_ status_event:Error sending evet - Erro-%1$d
EVOMAP125	Error receiving event - Reason:%1$s
EVOMAP130	Lost connection to pmd - Restart GUI
EVOMAP135	Status event has invalid variable type for variable # %1$d
EVOMAP140	process_status_event:OVwSetStatusOnObject failed _ %1$s
EVOMAP145	process_status_event:Unable to set status on object %1$d, status=%2$d
EVOMAP150	OVwUnmanageObject failed for object %1$d - Reason:%2$s
EVOMAP155	OVwManageObject failed for object %1$d - Reason:%2$s
EVOMAP160	Unable to get EVOStatus field for object %1$d
EVOMAP165	Error adding ovwConfirmManageObjects callback - Reason: %1$s
EVOMAP170	Error adding ovwConfirmUnmanageObjects callback - Reason:%1$s
EVOMAP175	Unable to get object info for %1$d - Reason:%2$s
EVOMAP180	delete_symbols:Failed to delete symbol %1$d - Reason:%2$s
EVOMAP185	Unable to get EVOName field for object %1$d
EVOMAP190	Unable to get EVOType field for object %1$d
EVOMAP195	Invalid type value (%1$d) in EVOType field
EVOMAP200	Error creating symbol for object %1$d - Reason:%2$s
EVOMAP205	create_symbol: Failed to delete symbol %1$d - Reason:%2$s

Code	Definition
EVOMAP210	Unable to get parent for object %1$d
EVOMAP215	Unable to create submap for parent object %1$d - Reason:%2$s
EVOMAP220	Parent field is missing for object %1$d
EVOMAP225	Unable to open admin config file %1$s
EVOMAP230	Unable to connect to pmd - retry in 1 minute);
EVOMAP240	Error adding ovwConfirmDeleteObjects callback - Reason:%1$s
EVOMAP245	Unable to connect to pmd - exiting !
EVOMAP250	Error adding ovwEndSession callback - Reason:%1$s

Table B-16: vp400ragt Messages

Code	Definition
EVORAG010	Unable to resolve host name %1$s
EVORAG020	Unable to open configuration file for %1$s
EVORAG030	Error retrieving configuration for %1$s
EVORAG040	Invalid port number found in configuration file for %1$s
EVORAG050	No response from %1$s or error occurred waiting for response
EVORAG100	Primary Manager switch successful
EVORAG110	Distribute All parameter is 'YES' on %1$s
EVORAG120	%1$s Management server requested for primary is not connected
EVORAG130	This server is not authorized to make this request
EVORAG900	Unknown return code %d from %1$s

Table B-17: vp400xreply Messages

Code	Definition
EVOXRY010	Unable to open DISPLAY
EVOXRY020	vp400hostcmd failed

Table B-18: vp400delete Messages

Code	Definition
EVODLT100	Error allocating pdu for event

Code	Definition
EVODLT110	Error adding varbind to event pdu
EVODLT115	Error initializing with ovw session - Reason:%1$s
EVODLT120	Error getting map information - Reason:%1$s
EVODLT125	Error opening event session - Reason:$1$s

Table B-19: vp400addagt Messages

Code	Definition
EVOADD000	**** AS/400 Configuration Tool ****
EVOADD010	Enter Internet name of AS/400:
EVOADD020	Enter a new value or press Enter to keep same value
EVOADD030	%1$s added to Node Bank (Holding Area)
EVOADD100	Unable to resolve hostname. Please try again
EVOADD110	Error copying base config file
EVOADD120	Error opening configuration file
EVOADD130	Error reading existing configuration files
EVOADD140	Error reading existing configuration file for %1$s
EVOADD150	Error opening current configuration file
EVOADD160	Error creating configuration file
EVOADD170	Error writing configuration file
EVOADD180	Interrupt signal received - exiting
EVOADD200	Error connecting to OVO management server - %1$s
EVOADD210	Error creating node structure - retun code %1$d
EVOADD220	Error setting initial node parameter - return code %1$d
EVOADD230	Error getting node defaults - return code %1$d
EVOADD240	Error setting node parameter - return code %1$d (%2$s)
EVOADD250	%1$s already exists in Node Bank
EVOADD260	Adding node failed! Return code %1$d (%2$s)
EVOADD270	Error creating node group! Return code %1$d (%2$s)
EVOADD280	Error setting node group! Return code %1$d (%2$s)
EVOADD290	Error creating nodelist! Return code %1$d (%2$s)
EVOADD292	Error adding node to nodelist! Return code %1$d

Code	Definition
	(%2$s)
EVOADD295	Error assigning node to node group! Return code %1$d (%2$s)
EVOADD297	Error disconnecting from OVO Management Server
EVOADD300	AS/400 configuration complete

C

Jobs in the EVSBS Subsystem

This appendix describes the several jobs that run under the EVSBS subsystem on the iSeries agent.

EView/400i Subsystem (EVSBS)

The jobs that execute in the EVSBS Subsystem:

1. EVACMDPROC – Establishes the TCP/IP socket for bi-directional command and response link.
2. EVCCTLPROC – Controls the processing of pre-defined API's used in command processing.
3. EVMSGQMON – Monitors message queues configured for SCAN mode monitoring.
4. EVPERFPROC – Gathers performance data.
5. EVSCMDPROC – Executes the command processor.
6. EVSMSGPROC – Message queue allocation and message processing.
7. EVSHSTPROC – Extracts messages at a configured time sequence from the QHST message queue depending on the message ID's added to the filter file. These messages are forwarded to the OM server and appear in the message browser.
8. EVSRSCPROC – Monitors status changes on discovered resources at a configured time sequence.
9. EVSTCPPROC – Receives and forwards all processed messages, commands, and API instructions from a central data queue.
10. EVTCTLPROC – Controls multiple connectivity between the OM server and the EView/400i agent.
11. EVAUDJRNL – The RCVJRNE exit which collects audit records from the QAUDJRN journal.

D

Performance Collection Metrics Classes

This appendix lists the performance metrics that can be collected by EView/400i.

Selecting Performance Metrics

Use the Node Configurator application to change the EV400_PERF1 and/or the EV400_PERF2 parameter to "YES" (see "Add iSeries Nodes and Runtime Parameters" beginning on page 16) based on the desired metrics classes listed below. Save and redistribute the configuration to the iSeries agent. Also, use the `ev400addperf.pl` script to initialize OM with the desired performance metric sets (see "Phase 8: Configuring Nodes for Performance Data Collection" on page 38).

Set 1

Specs used to create new OM performance measurement class	**Description**

OS400_HOSTNAME = 101
LABEL "OS/400 HOSTNAME"
TYPE TEXT LENGTH 256;
	iSeries agent name, used to define the class name. (Character string)

AVG_USERS_SIGNED_IN = 102
LABEL "Avg Users Signed In"
PRECISION 0;
	Average number of users signed on to the system during the polling interval. (Integer)

MIN_USERS_SIGNED_IN = 103
LABEL "Min Users Signed In"
PRECISION 0;
	Minimum number of users signed on to the system during the polling interval. (Integer)

MAX_USERS_SIGNED_IN = 104
LABEL "Max Users Signed In"
PRECISION 0;
	Maximum number of users signed on to the system during the polling interval. (Integer)

AVG_GBL_CPU_UTIL = 105
LABEL "Avg Global CPU Util"
PRECISION 1;
	Average percent of the polling interval time during which the processing units were in use. (Integer, in tenths)

MIN_GBL_CPU_UTIL = 106
LABEL "Min Global CPU Util"
PRECISION 1;
	Minimum percent of the polling interval time during which the processing units were in use. (Integer, in tenths)

MAX_GBL_CPU_UTIL = 107	Maximum percent of the polling interval

LABEL "Max Global CPU Util" PRECISION 1;	time during which the processing units were in use. (Integer, in tenths)

AVG_JOBS_IN_SYSTEM = 108
LABEL "Avg Jobs in System"
PRECISION 0;
	Average total number of user and system jobs that are currently in the system, including jobs waiting on queues. (Integer)

MIN_JOBS_IN_SYSTEM = 109
LABEL "Min Jobs In System"
PRECISION 0;
	Minimum total number of user and system jobs that are currently in the system, including jobs waiting on queues. (Integer)

MAX_JOBS_IN_SYSTEM = 110
LABEL "Max Jobs in System"
PRECISION 0;
	Maximum total number of user and system jobs that are currently in the system, including jobs waiting on queues. (Integer)

AVG_PERCENT_DB_CAP = 111
LABEL "Avg Pct DB Cap"
PRECISION 1;
	Average percentage of processor database capability that was used during the polling interval. (Integer, in tenths)

MIN_PERCENT_DB_CAP = 112
LABEL "Min Pct DB Cap"
PRECISION 1;
	Minimum percentage of processor database capability that was used during the polling interval. (Integer, in tenths)

MAX_PERCENT_DB_CAP = 113
LABEL "Max Pct DB Cap"
PRECISION 1;
	Maximum percentage of processor database capability that was used during the polling interval. (Integer, in tenths)

AVG_DATABASE_FAULTS = 114
LABEL "Avg Database Faults"
PRECISION 1;
	Average number of faults over all pools during the polling interval for pages containing either database data or access paths. (Integer, in tenths representing faults per second)

MAX_DATABASE_FAULTS = 115
LABEL "Max Database Faults"
PRECISION 1;
	Maximum number of faults over all pools during the polling interval for pages containing either database data or access paths. (Integer, in tenths representing faults per second)

AVG_DATABASE_PAGES = 116
LABEL "Database Pages"
PRECISION 1;
	Average cumulative rate over all pools during the polling interval at which database pages are brought into the storage pool (Integer, in tenths

	representing pages per second)

AVG_NON_DB_FAULTS = 117
LABEL "Avg Non DB Faults"
PRECISION 1;
	Average number of faults over all pools during the polling interval for pages other than those designated as database pages. (Integer, in tenths representing faults per second)

MAX_NON_DB_FAULTS = 118
LABEL "Max Non DB Faults"
PRECISION 1;
	Maximum number of faults over all pools during the polling interval for pages other than those designated as database pages. (Integer, in tenths representing faults per second)

AVG_NON_DB_PAGES = 119
LABEL "Avg Non DB Pages"
PRECISION 1;
	Average cumulative rate over all pools during the polling interval at which pages other than those designated as database pages are brought into the storage pool (Integer, in tenths representing pages per second)

AVG_JOB_CPU_UTIL = 120
LABEL "Avg Job CPU Util"
PRECISION 1;
	Average percentage of processing time used by all batch jobs during the polling interval. (Integer)

MIN_JOB_CPU_UTIL = 121
LABEL "Min Job CPU Util"
PRECISION 1;
	Minimum percentage of processing time used by all batch jobs during the polling interval. (Integer)

MAX_JOB_CPU_UTIL = 122
LABEL "Max Job CPU Util"
PRECISION 1;
	Maximum percentage of processing time used by all batch jobs during the polling interval. (Integer)

AVG_INT_CPU_UTIL = 123
LABEL "Avg Int CPU Util"
PRECISION 1;
	Average percentage of processing time used by all interactive jobs during the polling interval. (Integer)

MIN_INT_CPU_UTIL = 124
LABEL "Min Int CPU Util"
PRECISION 1;
	Minimum percentage of processing time used by all interactive jobs during the polling interval. (Integer)

MAX_INT_CPU_UTIL = 125 LABEL "Max Int CPU Util"	Maximum percentage of processing time used by all batch jobs during the polling

PRECISION 1;	interval. (Integer)

NUM_INTER_TRANS = 126
LABEL "Number Int Trans"
PRECISION 0;
	Average number of user interactions, such as pressing the Enter key or a function key, for all interactive jobs during the polling interval. (Integer)

AVERAGE_RESP_TIME = 127
LABEL "Avg Response Time"
PRECISION 3;
	Average interactive response time for the initial thread of all interactive jobs during the polling interval. (Integer, in hundredths of seconds)

MAX_AVG_RESP_TIME = 128
LABEL "Max Avg Resp Time"
PRECISION 3;
	Maximum interactive response time for the initial thread of all interactive jobs during the polling interval. (Integer, in hundredths of seconds)

AVG_IO_PER_SEC = 129
LABEL "Avg I/O Per Second"
PRECISION 1;
	Average number of blocks transferred to and from the disk units during the polling interval. (Integer)

MAX_IO_PER_SEC = 130
LABEL "Max I/O Per Second"
PRECISION 1;
	Maximum number of blocks transferred to and from the disk units per second during the polling interval. (Integer)

AVG_READ_PER_SEC = 131
LABEL "Avg Read Per Second"
PRECISION 1;
	Average number of blocks transferred from the disk units per second during the polling interval. (Integer)

MAX_READ_PER_SEC = 132
LABEL "Max Read Per Second"
PRECISION 1;
	Maximum number of blocks transferred from the disk units per second during the polling interval. (Integer)

AVG_WRITE_PER_SEC = 133
LABEL "Avg Write Per Second"
PRECISION 1;
	Average number of blocks transferred to the disk units per second during the polling interval. (Integer)

MAX_WRITE_PER_SEC = 134
LABEL "Max Write Per Second"
PRECISION 1;
	Maximum number of blocks transferred to the disk units per second during the polling interval. (Integer)

AVG_DISK_BUSY = 135
LABEL "Avg Disk Busy"
PRECISION 1;
	Average percentage of time that the disk queues of all disks contained data to read or write during the polling interval. (Integer, expressing percentage to thousandths)

MAX_DISK_BUSY = 136
LABEL "Max Disk Busy"
PRECISION 1;
	Maximum percentage of time that the disk queues of all disks contained data to read or write during the polling interval. (Integer, expressing percentage in thousandths)

Set 2

Specs used to create new OM performance measurement class	**Description**

OS400_HOSTNAME = 101
LABEL "OS/400 Hostname"
TYPE TEXT LENGTH 256;
	iSeries agent name, used to define the class name. (Character string)

PCT_PERM_ADDR = 102
Label "Percent Perm Addr"
PRECISION 1;
	Percentage of the maximum possible addresses for permanent objects that have been used. (Integer, expressing percentage in thousandths)

PCT_TEMP_ADDR = 103
Label "Percent Temp Addr"
PRECISION 1;
	Percentage of the maximum possible addresses for temporary objects that have been used. (Integer, expressing percentage in thousandths)

SYSTEM_ASP = 104
Label "System ASP"
PRECISION 0;
	Storage capacity of the system auxiliary storage pool (ASP1). (Integer, expressed in Megabytes).

PCT_SYSTEM_ASP_USED = 105
Label "Pct System ASP Used"
PRECISION 1;
	Percentage of the system storage pool currently in use. (Decimal, expressed in ten thousandths)

TOTAL_AUX_STORAGE = 106
Label "Total Aux Storage"
PRECISION 0;
	Total auxiliary storage on the system. (Integer, in Megabytes)

CUR_UNPROT_STOR_USED = 107
Label "Cur Unprot Stor Used"
PRECISION 1;
	Current amount of storage in use for temporary objects. (Integer, in Megabytes)

MAX_UNPROT_STOR_USD = 108
Label "Max Unprot Stor Used"
PRECISION 1;
	Largest amount of storage for temporary objects used at any one time since the last IPL. (Integer, in Megabytes)

MAIN_STOR_SIZE = 109
Label "Main Storage Size"
PRECISION 0;
	Amount of main storage in the system. On a partitioned system, the main storage size can change while the system is active. (Integer, in Kilobytes)

NUM_MEMORY_POOLS = 110
Label "Num of Memory Pools"
PRECISION 0;
	The number of pools allocated. (Integer)

E

Message Text of Audit Journal Entries

This appendix describes how iSeries audit records received from the QAUDJRN will be presented on the OM browser. All journal messages begin with an "AUD0000" message ID header.

Audit Journal Type AD (Auditing changes)

(AD) {*cmdname*|**Undefined**} **command, Object:** *objname*/*libname* **Type:** *objtype* **Value:** *audval* **Level:** {*actlvl*[,*actlvl*...]|**NONE**} [**DLO Object:** *dloobj*]

where:

cmdname – The command which triggered this audit entry, one of:

```
CHGDLOAUD
CHGAUD
CHGATTR
CHGUSRAUD
```

objname – The name of the object for which auditing was changed.
libname – The name of the library of the object.
objtype – The type of object.
audval – The audit value specified in the command. If the scan attribute was changed using the CHGATR command, *audval* contains the scan attribute value.
actlvl – The level of activity that is audited for *objname*.
dloobj – The DLO object, if one exists.

Sample Message:

```
AUD0000 (AD) CHGUSRAUD command, Object: USER1/QSYS Type: *USRPRF
Value: *ALL Level: *CMD,*CREATE,*DELETE
```

Audit Journal Type AF (Authority failure)

(AF) *failuretext* [**Validation Error Action:** *actiontext*] [**(***violationcode***)** *violationtext*] **Object:** *objname*[/*libname*] [**Type:** *objtype*] **Job Name:** *jobname* **User Profile:** *usrprf*

where:

failuretext – Description of the authority failure, one of:

```
Not authorized to object
Restricted instruction
Validation failure:
Use of unsupported interface
Storage protection error
ICAPI authorization error
ICAPI authentication error
Scan exit program action:
System Java inheritence not allowed
Submit job profile error
Profile token not regenerable
Optical object authority failure
Profile swap error
Hardware protection error
```

```
Default sign-on attempt
Not authorized to TCP/IP port
User permission request not valid
Profile token not valid for generating new token
Profile token not valid for swap
System violation:
Not authorized for a clear JUID operation
Not authorized for a set JUID operation
Undefined violation
```

actiontext – If *failuretext* is either "Validation failure: " or "Scan exit program action: " then this action is taken, one of:

```
Object translation not attempted or failed
Object translation was successful
System install time error detected
Restore failed, signature not in OS/400 format
Unsigned system or inherit state object found
Unsigned user state object found
Mismatch between object and its signature
IBM certificate not found
Invalid signature format found
Scan exit program modified the object
Scan exit program wanted object marked as failure
Unrecognized action
```

violationcode, *violationtext* – If *failuretext* is "System violation: " then this describes the type of violation that occurred, one of:

```
(HCA) Service tool user not authorized for hardware
config
(LIC) PTF not applied due to signature violation
(SFA) Not authorized for system file access
(CMD) Command disabled by sysadmin
```

objname – The name of the object. If *failuretext* is "Not authorized to TCP/IP port", then this field will contain the port number.

libname – The name of the library of the object. This is not displayed if *failuretext* is "Not authorized to TCP/IP port".

objtype – The type of object. This is not displayed if *failuretext* is "Not authorized to TCP/IP port".

jobname – The name of the job.

usrprf – The name of the user that caused the authority failure.

Sample Message:

```
AUD0000 (AF) Not authorized to object Object: MYOBJ/MYLIB Type:
*FILE Job Name: QPADEV0001 User Profile: USER1
```

Audit Journal Type AU (Attribute changes)

(AU) [**New CSSID:** *newcssid* **Old CSSID:** *oldcssid*][,][**New Country ID:** *newcountry* **Old Country ID:** *oldcountry*][,][**New Language ID:** *newlang* **Old Language ID:** *oldlang*][,][**Attribute:** *attrname* **New Value:** *newattr* **Old Value:** *oldattr*]

where:

newcssid, *oldcssid* – The new and old CSSID values, if there was a change.

newcountry, *oldcountry* – The new and old Country ID values, if there was a change.

newlang, *oldlang* – The new and old Language ID values, if there was a change.

attrname – The name of the attribute, if there was a change.

newattr, *oldattr* – The new and old attribute values, if there was a change.

Sample Message:

```
AUD0000 (AU) New Country ID: DE  Old Country ID: US
```

Audit Journal Type CA (Authority changes)

(CA) Object: *objname*/*libname* **User:** *usrprf* **Command type:** *cmdtype* **Authorities altered:** {*auth*[,*auth*...]|**NONE**}

where:

objname – The name of the object.

libname – The library of the object.

usrprf – The user profile whose authority is being modified.

cmdtype – The type of command used, one of:

```
Grant
Grant/Replace
Revoke
GRTUSRAUT
```

auth – The authorities granted or removed, one or more of:

```
*OBJEXIST
*OBJMGT
*OBJOPR
*AUTLMGT
*AUTL
*READ
*ADD
*UPD
*DLT
*EXCLUDE
```

```
*EXECUTE
*OBJALTER
*OBJREF
```

Sample Message:

```
AUD0000 (CA) Object: OBJ1/MYLIB User: USER1 Command type: Grant
Authorities altered: *ADD,*UPD,*DLT
```

Audit Journal Type CD (Command string)

(CD) Command: *cmdstring* **issued from job:** *job***/***user***/***jnum* **CL Program Call:** {**Yes**|**No**}

where:

cmdstring – The name of the command executed.

job – The name of the job that caused this entry to be created.

user – The user profile associated with *job*.

jnum – The job number.

NOTE: To generate a message to OM, the *cmdstring* must be in the list of commands defined in “Phase 5: Identify Command Audit Filters” (see page 34).

Sample Message:

```
AUD0000 (CD) Command: DLTUSRPRF issued from job: USER1/USER1/123456
CL Program Call: No
```

Audit Journal Type CO (Create Object)

(CO) Object: *objname***/***objlib* {**created**|**replaced**}, **Type:** *objtype* **from job:** *job***/***user***/***jnum*

where:

objname – The name of the object.

objlib – The library of the object.

objtype – The type of the object.

job – The name of the job that caused this entry to be created.

user – The user profile associated with *job*.

jnum – The job number.

Sample Message:

```
AUD0000 (CO) Object: MYOBJ/MYLIB created, Type: *MODULE from job:
QPADEV0003/USER1/123456
```

Audit Journal Type CP (User profile changed, created, or restored)

(CP) User profile: *usrprf* **changed via** *method* [**(password changed)**]
[**Profile status:** *status*] [**User class:** *class*] **from job:** *job***/***user***/***jnum*

where:

usrprf – The user profile that was changed.

method – The type of command used, one of:

```
            CRTUSRPRF command
CHGUSRPRF command
RSTUSRPRF command
QSECOFR password reset using DST
QSYSRESPA API
Undefined method
```

status – The user profile status, if changed.

class – The user class of the user, if one exists.

job – The name of the job that caused this entry to be created.

user – The user profile associated with *job*.

jnum – The job number.

Sample Message:

```
AUD0000 (CP) User profile: USER1 changed via CHGUSRPRF command
Profile status: *ENABLED from job: QPADEV0003/USER1/123456
```

Audit Journal Type DO (Delete Operation)

(DO) Object: *objname***/***objlib action*, **Type:** *objtype* **from job:**
*job***/***user***/***jnum*

where:

objname – The name of the object.

objlib – The library of the object.

action – The type of action taken, one of:

```
deleted
pending delete committed
pending create rolled back
```

```
            delete pending
            pending delete rolled back
```

`objtype` – The type of the object.

`job` – The name of the job that caused this entry to be created.

`user` – The user profile associated with `job`.

`jnum` – The job number.

Sample Message:

```
AUD0000 (DO) Object: MYOBJ/MYLIB created, Type: *FILE from job:
QPADEV0003/USER1/123456
```

Audit Journal Type DS (DST security password reset)

(DS) Service Tools User: *userid action* **as requested by** *requestor*

where:

`userid` – The service tools user ID.

`action` – The type of action taken, one of:

```
            ID was changed
            password reset
            password changed
```

`requestor` – The service tools user ID that requested the change.

Sample Message:

```
AUD0000 (DS) Service Tools User USER1 password changed as requested
by QSECOFR
```

Audit Journal Type NA (Network Attribute Change)

(NA) {**Network**|**TCP/IP**} **attribute:** *val* **changed from:** *oldval* **to:** *newval* **from job:** *job*/*user*/*jnum*

where:

`val` – The name of the attribute that was modified.

`oldval` – The value before it was changed.

`newval` – The new value.

`job` – The name of the job that caused this entry to be created.

`user` – The user profile associated with `job`.

`jnum` – The job number.

Sample Message:

```
AUD0000 (NA) TCP/IP attribute: TCPKEEPALV changed from: 120 to: 140
from job: QPADEV0003/USER1/123456
```

Audit Journal Type OW (Object ownership changed)

(OW) Object: *objname***/***libname* **ownership changed from:** *old* **to:** *new* **from job:** *job***/***user***/***jnum*

where:

objname – The name of the object.

libname – The name of the library of the object.

old – The old owner of the object.

new – The new owner of the object.

job – The name of the job that caused this entry to be created.

user – The user profile associated with *job*.

jnum – The job number.

Sample Message:

```
AUD0000 (OW) Object: MYOBJ/MYLIB ownership changed from: USER1 to:
USER2 from job: QPADEV0003/USER1/123456
```

Audit Journal Type PA (Program changed to adopt authority)

(PA) {**Program** *pgmname***/***libname* **adopted authority of owner:** *ownername* | **Object:** {*objname*|**NONE**} [**SETUID mode:** {**Y**|**N**}] [**SETGID mode:** {**Y**|**N**}]}

where:

pgmname – The name of the program that was modified.

libname – The name of the library of the *pgmname*.

ownername – The name of the owner.

objname – The name of the object, if it exists and if the SETUID or SETGID has been modified.

Sample Message:

```
AUD0000 (PA) Program MYPROG/MYLIB adopted authority of owner: USER1
```

Audit Journal Type PG (Change of an object's primary group)

(PG) Object: *objname*/*objlib* **changed group from:** *oldgrp* **to:** *newgrp*

where:

objname – The name of the object for which the group was changed.

libname – The name of the library of the *objname*.

oldgrp – The previous primary group, or "*N" if the old group was not available.

newgrp – The new primary group for the object.

Sample Message:

```
AUD0000 (PG) Object MYOBJ/MYLIB changed group from GRP1 to GRP2
```

Audit Journal Type PW (Invalid password)

(PW) User: *username* **failed:** *violation* **on:** *device* [**remote name:** *remote*] [**local name:** *local*]

where:

username – The job user name or service tools user ID.

violation – The type of violation, one of:

```
APPC bind failure
Service Tools ID name not valid
Service Tools ID password not valid
Password invalid
SQL Decryption password not valid
User name not valid
Service Tools user ID disabled
Service Tools ID not valid
Service Tools ID password not valid
Undefined violation
```

device – The name of the device where the user ID or password was entered. If *violation* is one of: "Service Tools user ID disabled", "Service Tools ID not valid", or "Service Tools ID password not valid", then the *device* field will contain the name of the service tool being accessed.

remote – The name of the remote location for the APPC bind, if one exists.

local – The name of the local location for the APPC bind, if one exists.

Sample Message:

```
AUD0000 (PW) User: USER1 failed: Password invalid on: QPADEV0007
```

Audit Journal Type ST (Use of service tools)

(ST) Service tool *type* **accessed**[**object** *objname*/*libname*][**for job** *jobname*/*username*/*jobnum*]

where:

type – The type of service tool, one of:

```
ANZJVM
STRCPYSCN
QTACTLDV
QWTCTLTR
DMPCLUTRC
DLTCMNTRC
DMPDLO
DMPJVM
DMPOBJ
DMPSYSOBJ,QTADMPTS
ENDCMNTRC
ENDRMTSPT
QYHCHCOP(DASD)
QYHCHCOP(LPAR)
QPYRTJWA
PRTCMNTRC
PRTERRLOG
PRTINTDTA
QP0FPTOS
QWTSETTR
STRCMNTRC
STRSRVJOB
STRRMTSPT
STRSST
TRCTCPAPP
TRCCNN(*FORMAT)
ENDTRC,ENDPEX
TRCINT,TRCCNN(*ON/*OFF/*END)
STRTRC,STRPEX
UNKNOWN
```

objname – The object accessed, if given.

libname – The name of the library of the *objname*.

jobname – Part 1 of the qualified job name, if given.

username – Part 2 of the qualified job name.

jobnum – Part 3 of the qualified job name.

Sample Message:

```
AUD0000 (ST) Service Tool QPOFPTOS accessed object MYOBJ/MYLIB for
job TEST/USER1/123456
```

Audit Journal Type SV (System value changed)

(SV) System value change: *sysval* **changed from:** *oldval* **to:** *newval*

where:

sysval – The system value that was modified.
oldval – The value before it was changed.
newval – The new value.

Sample Message:

```
AUD0000 (SV) System value change: QAUDLVL changed from: *AUTFAIL
*SYSMGT to: *AUTFAIL *SYSMGT *SECURITY
```

Audit Journal Type VA (Changing an access control list)

(VA) Access control list {**addition**|**modification**|**deletion**} {**successful**|**failed**} **from user** *username* **at** *location* **for resource** *rscname*

where:

username – The name of the user issuing the request to change the access control list.
location – The name of the computer issuing the request.
rscname – The name of the resource to be changed.

Sample Message:

```
AUD0000 (VA) Access control list modification successful from user
USER1 at QPADEV0005 for resource n
```

Audit Journal Type VP (Network password error)

(VP) User: *username* **network password error on:** *device*

where:

username – The name of the user attempting to log on.
device – The computer initiating the logon request.

Sample Message:

```
AUD0000 (VP) User: USER1 network password error on: DEV1
```

Audit Journal Type VU (Changing a network profile)

(VU) User: *username* **on device:** *device* **requested network profile action:** *action* **for record:** *rectype* **resource:** *rscname*

where:

username – The name of the user requesting the profile change.

device – The name of the computer requesting the profile change.

action – The requested action, one of:

```
addition
change
deletion
incorrect password
undefined
```

rectype – The type of record changed, one of:

```
group
user
user profile global information
undefined
```

rscname – The name of the resource.

Sample Message:

```
AUD0000 (VU) User: USER1 on device: DEV1 requested network profile
action: change for record: user resource: n
```

Audit Journal Type ZC (Object accessed (changed))

(ZC) Object: *objname*/*libname* **type:** *objtype* {**changed**|**upgraded**} **by job:** *job*/*user*/*jnum* **access type:** *acctype*

where:

objname – The object accessed.

libname – The name of the library of the *objname*.

objtype – The object type of *objname*.

job – The name of the job that caused this entry to be created.

user – The user profile associated with *job*.

jnum – The job number.

acctype – The type of access, one of:

Add
Activate program
Analyze
Apply
Call or TFRCTL
Configure
Change
Check
Close
Clear
Compare
Cancel
Copy
Create
Convert
Debug
Delete
Dump
Display
Edit
End
File
Grant
Hold
Initialize
List
Move
Merge
Open
Print
Query
Reclaim
Receive
Read
Reorganize
Release
Release
Remove
Rename
Replace
Resume
Restore
Retrieve
Run
Revoke
Save
Save with storage free
Save and delete
Submit
Set
Send
Start
Transfer
Trace
Verify
Vary
Work
Read/change DLO attribute
Read/change DLO security
Read/change DLO content
Read/change DLO all parts
Add constraint
Change constraint
Remove constraint
Start procedure
Get access on *OOPOOL
Sign object
Remove all signatures
Clear a signed object
Mount
Unload
End rollback
Undefined: *n*

Sample Message:

```
AUD0000 (ZC) Object: MYOBJ/MYLIB type: *FILE changed by job:
QPADEV0003/USER1/123456 access type: Change
```

Audit Journal Type ZR (Object accessed (read))

(ZR) Object: *objname*/*libname* **type:** *objtype* **read by job:**
job/*user*/*jnum* **access type:** *acctype*

where:

objname – The object accessed.

libname – The name of the library of the *objname*.

objtype – The object type of *objname*.

`job` – The name of the job that caused this entry to be created.

`user` – The user profile associated with `job`.

`jnum` – The job number.

`acctype` – The type of access, one of:

```
Add
Activate program
Analyze
Apply
Call or TFRCTL
Configure
Change
Check
Close
Clear
Compare
Cancel
Copy
Create
Convert
Debug
Delete
Dump
Display
Edit
End
File
Grant
Hold
Initialize
List
Move
Merge
Open
Print
Query
Reclaim
Receive
Read
Reorganize
Release
Release
Remove
Rename
Replace
Resume
Restore
Retrieve
Run
Revoke
Save
Save with storage free
Save and delete
Submit
Set
Send
Start
Transfer
Trace
Verify
Vary
Work
Read/change DLO attribute
Read/change DLO security
Read/change DLO content
Read/change DLO all parts
Add constraint
Change constraint
Remove constraint
Start procedure
Get access on *OOPOOL
Sign object
Remove all signatures
Clear a signed object
Mount
Unload
End rollback
Undefined: n
```

Sample Message:

```
AUD0000 (ZR) Object: MYOBJ/MYLIB type: *FILE read by job:
QPADEV0003/USER1/123456 access type: Read
```

Glossary

Central processing unit See CPU

See CPU.

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

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 Queue

An iSeries (AS/400) system object that holds data in which a program writes to read from in FIFO order.

disk pack

See DASD.

domain

An iSeries (AS/400) system, along with all of its lines, controllers and devices.

Export

Internet address the internet protocol routes data to.

Initial Program Loader

See IPL.

IPL

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

Legacy Link Interface

See LLI.

LLI

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

Mapping

A list usually in a profile that establishes a correspondence between items in two groups.

Message Queue

A data queue that holds messages from a specific area of the iSeries system. For example QSYSOPR is the message queue for the operating system.

Motif

A set of guidelines that specifies how a user interface for graphical computers should appear on the screen and how the user interacts with it.

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.

Node

See Domain.

OpenView Windows

See OVW.

OVW OpenView Windows. Customizable OpenView network management GUI.

Port

An access point for data entry and exit.

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 server, 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.

system restart

See IPL.

system startup

See IPL.

TCP

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

See also TCP/IP.

TCP/IP

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

See also TCP.

Transmission Control Protocol

See TCP.

Transmission Control Protocol/Internet Protocol

See TCP/IP.