

# **About OpenVMS Managed Nodes**

**HP Openview Operations (OVO) Agent for OpenVMS**

February 2006

**Hewlett-Packard Company**

**Palo Alto, California**

---

## **Legal Notices**

### **Warranty**

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to changed without notice.

### **Restricted Rights Legend**

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### **Copyright Notices**

Copyright 2003-2005, 2006 Hewlett-Packard Development Company, L.P.

---

## In this Document

This document explains how to install and configure HP OpenView Operations (OVO) on OpenVMS managed nodes.

---

## Installation Requirements

This section explains OpenView hardware and software requirements for OpenVMS managed nodes.

### Hardware Requirements

Before installing OpenView, make sure the OpenVMS systems you select as managed nodes meet the following disk space requirements:

- ❑ **Patch Kit Size on the OVOU Server:** 7 MB
- ❑ **Kit Sizes on OpenVMS Managed Nodes:**
  - Agents: 15 MB
  - SPI: 2 MB

---

**NOTE** For OpenVMS cluster systems, this disk space is required on each node in the cluster.

---

### Software Requirements

Before installing OpenView, make sure the following software is installed on OpenVMS managed nodes:

#### ❑ Operating System

The OpenVMS operating system versions supported by OpenView are the following:

- Version 7.3-1 with patches
- Version 7.3-2

#### ❑ Required Patches for the OpenVMS Version 7.3-1 Managed Nodes

Choose ONE of the following:

- The update kit containing the following patch kits:
  - VMS731\_UPDATE V2.0 or later
  - VMS731\_PCSI V1.0 or later (required for the UPDATE kit)
- If you do **not** want to use the update kit, install the following files:
  - VMS731\_ACRTL V1.0 or later

- VMS731\_SYS V2.0
- VMS 732\_SYSLOA-V00100

HP recommends that you also install the DCE Denial of Service Security Patch (DCE\_030\_SSRT3608 V1.0).

All of these files are available at the following services web site:

**<http://www2.itrc.hp.com/service/patch/mainPage.do>**

#### ❑ **Basic Networking Services**

The required networking services are the following:

- TCP/IP needs to be running.
- SNMP must be enabled if you want the Management Server to discover your system automatically.

---

#### **NOTE**

The OpenVMS agent has been verified with the HP TCP/IP protocol stack; it has not been verified with any other protocol stacks. No reason exists, however, for the OpenVMS agent not to work.

---

#### ❑ **DCE Runtime Kit**

The DCE Runtime Kit is supplied as part of the base operating system. No installation is required.

---

## **Installing and Deinstalling Agents**

This section describes how to install and deinstall OVO on OpenVMS managed nodes.

To manage an OpenVMS system from an OpenView Management Server, you must complete actions on both the Management Server system and on the OpenVMS system. For this reason, this section contains several sets of instructions. They are the following:

- Configuring the Management Server on a UNIX system
- Installing agent software on an OpenVMS system
- Updating the Management Server
- Installing the Smart Plug-In (SPI) for OpenVMS

The following sections explain how to complete each of these tasks.

### **Configuring the Management Server on a UNIX System**

For a UNIX Management Server, configuration updates are supplied within a gunzip-compressed depot file. (Later in the procedure, you copy the software required for the OpenVMS node or nodes from the OpenView Management Server to the OpenVMS node or nodes.)

To distribute the configuration updates on an OpenView Management Server, perform the following numbered steps:

1. Log in to your OpenView Management Server.

2. Copy the file OVOPC-VMS-CLT.depot.gz to the OpenView Management Server into a temporary directory such as /tmp.
3. To make the required configuration updates to the OpenView Management Server, perform the following lettered steps:

---

**NOTE** Perform the lettered steps in this section ONLY ONCE, even if you add nodes or perform other update tasks later.

---

- a. To distribute the files on the Management Server, set your directory to the root directory, and run gunzip on the file to create the depot file. Then install the depot file by entering the following commands:

```
# cd /  
# gunzip<your-temp-directory>OVOPC-VMS-CLT.depot.gz  
# swinstall<your-temp-directory> \  
> OVOPC-VMS-CLT.depot OVO-CLT.OVOPC-VMS-CLT
```

- b. Load the Agent Platform file into the OpenView database:  
# **opcagtdbcfg -p hp/alpha/vms -d -f**
- c. Upload the templates so that they can be distributed later to the agent nodes:  
# **opccfgupld -add VMSPkg**

---

**NOTE** If you have completed step “c” previously, use the **-replace** command instead of **-add**.

---

4. Copy the agent and SPI kits to your OpenVMS nodes. To do this, perform the following lettered steps:
  - a. Go to the directory that contains the tarred .PCSI files:  
# **cd /var/opt/OV/share/databases/OpC/mgd\_node \**  
**> /vendor/hp/alpha/vms/ A.07.10/RPC\_DCE\_TCP**
  - b. Extract the OpenVMS PCSI files from the tar files:  
# **tar -xvf opc\_pkg.Z**  
# **tar -xvf spi\_pkg.Z**
  - c. Copy the PCSI kits to the OpenVMS nodes that you want to manage. From the OpenView Management Server, connect to the OpenVMS node using the SYSTEM account. Change your directory to the SYSSUPDATE directory, set the transfer mode to binary, and transfer the PCSI kits from the UNIX machine to the OpenVMS node, using the following commands:  
# **ftp <your-OpenVMS-node>**  
**SYSTEM**  
**<password>**  
**ftp> cd SYSSUPDATE:** (The colon is required.)

**ftp> bin**

**ftp> put hp-axpvms-ovoagents-<kit>.pcsi**

**ftp> put hp-axpvms-vmsspi-<kit>.pcsi**

Note: Replace <kit> with the version number that is included in the file name of the extracted PCSI file; for example, V0100-51-1.

## Installing Agent Software on an OpenVMS System

Follow the steps in this section AFTER completing “Configuring the Management Server on a UNIX System” on page 4. Also, you must install the Agent software BEFORE installing the SPI for OpenVMS software

---

**NOTE** OpenVMS supports only manual installations.

---

To install the kit, follow these numbered steps:

1. Log in to the SYSTEM account.
2. Set your default directory to SYSSUPDATE:

**\$ SET DEFAULT SYSSUPDATE**

3. Use the PRODUCT INSTALL command to install the kit:

**\$ PRODUCT INSTALL OVOAGENTS**

The installation procedure moves most files to subdirectories under SYSSSPECIFIC:[OVO]. Several files are moved to standard system directories such as SYSSSTARTUP and SYSSMANAGER. One file, OVO\$LIBOPC\_R.EXE, is moved to SYSSSYSROOT:[SYSLIB].

---

**NOTE** You must use the PRODUCT INSTALL command on each node in an OpenVMS cluster.

---

4. To start the agents, enter the following commands, and, when requested, enter the name of your Management Server:

**\$ @SYSSMANAGER:OVO\$CONFIG**

**\$ @SYSSSTARTUP:OVO\$STARTUP**

## Performing Additional Steps to Update the Management Server

Follow the steps in this section AFTER completing the steps in “Installing the Agent Software on an OpenVMS System.” Refer to the appropriate OpenView documentation if you need additional instructions for completing these tasks:

1. Add the OpenVMS nodes and clusters you want to monitor.
2. Distribute the templates you want to use to each OpenVMS node you want to monitor.
3. Add the OpenVMS message group to the responsibilities matrix for the users who will monitor messages from the SPIs.

## Installing the Smart Plug-In (SPI) for OpenVMS

Follow the steps in this section AFTER you complete the configuration steps AND the installation steps.

To install the files in the kit, follow these numbered steps:

1. Log in to the SYSTEM account.
2. Set your default directory to SYSSUPDATE:

```
$ SET DEFAULT SYSSUPDATE
```

3. Use the PRODUCT INSTALL command to install the kit:

```
$ PRODUCT INSTALL VMSSPI
```

The distribution procedure moves most files to subdirectories under SYSSSPECIFIC:[OVO]. Several files are moved to standard system directories such as SYSSSTARTUP and SYSSMANAGER.

4. Start the SPI:

```
$ @SYSSSTARTUP:VMSSPI$STARTUP
```

To customize the SPI configuration files, refer to the *HP OpenView Smart Plug-In (SPI) for OpenVMS User's Guide*. You can find .PS and .PDF formats of this manual, whose file name is VMSSPI\_USER\_GUIDE, in SYSSSPECIFIC:[OVO].

## Deinstalling Agents

To deinstall agent software that has been installed on managed nodes, follow these steps:

1. Stop the SPI:

```
$ @SYSSSTARTUP:VMSSPI$SHUTDOWN
```

2. Stop the agents running on managed nodes:

```
$ @SYSSSTARTUP:OVO$SHUTDOWN
```

3. Deinstall the SPI:

```
$ PRODUCT REMOVE SPI
```

4. Deinstall the agent software:

```
$ PRODUCT REMOVE OVOAGENTS
```

5. Clean up the directory tree under SYSSSPECIFIC:[OVO...].

---

## Troubleshooting

The following sections contain troubleshooting tips for OpenView managed nodes on OpenVMS systems.

### Producing a Clean DCE Startup

If the OpenView agents do not start up after you enter the OVO\$STARTUP command, you need to clean up your DCE/RPC database. To do this, you must produce a clean DCE startup by following these numbered steps:

1. Check your existing entry point mapping by entering the following command:

```
$ mcr sys$system:dce$rpccp show mapping
```

The system then displays mapping information similar to the following:

```
<object> 2666fadc-6a7a-115b-9af6-0000f81a99d8
<interface id> e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0
<string binding> ncacn_dnet_nsp:19.794 [#69]
<annotation> Endpoint Resolution

<object> 2666fadc-6a7a-115b-9af6-0000f81a99d8
<interface id> e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0
<string binding> ncadg_ip_udp:16.32.192.223 [135]
<annotation> Endpoint Resolution

<object> nil
<interface id> 9e0c0224-3654-0000-9a8d-08000949ab4c,2.0
<string binding> ncacn_ip_tcp:16.32.192.223 [50813]
<annotation> Control Agent (COA)

<object> nil
<interface id> 9e0c0224-3654-0000-9a8d-08000949ab4c,2.0
<string binding> ncadg_ip_udp:16.32.192.223 [49336]
<annotation> Control Agent (COA)

<object> nil
<interface id> 0d8fe322-d6ee-11d2-b858-0800096df3a6,1.0
<string binding> ncacn_ip_tcp:16.32.192.223 [50813]
<annotation> Control Agent (cfgpush)

<object> nil
<interface id> 0d8fe322-d6ee-11d2-b858-0800096df3a6,1.0
<string binding> ncadg_ip_udp:16.32.192.223 [49336]
<annotation> Control Agent (cfgpush)
```

2. To clean up all entry point mapping, enter the following command:

```
$ @sys$manager:dce$rpc_shutdown CLEAN
```

3. To verify that the database is clean, restart DCE and re-enter the SHOW MAPPING command:

```
$ @sys$manager:dce$rpc_startup
```

```
$ mcr sys$system:dce$rpccp show mapping
```

If you still see COA and CFGPUSH entries in the display, delete the DCE database. To do this, enter the following commands:

```
$ sys$manager:dce$rpc_shutdown
```

```
$ delete SYSSYSDEVICE:[SYS0.DCELOCAL.VAR.DCED]-
```

```
_ $ -EP.DB;*
```

4. Now restart DCE and look at mappings by entering the following commands:

```
$ @sys$manager:dce$rpc_startup
```

```
$ mcr sys$system:dce$rpccp show mapping
```

The system display should be similar to the following:



```
<object> 2666fadc-6a7a-115b-9af6-0000f81a99d8
<interface id> e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0
<string binding> ncacn_dnet_nsp:19.794[#69]
<annotation> Endpoint Resolution

<object> 2666fadc-6a7a-115b-9af6-0000f81a99d8
<interface id> e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0
<string binding> ncadg_ip_udp:16.32.192.223[135]
<annotation> Endpoint Resolution

<object> 2666fadc-6a7a-115b-9af6-0000f81a99d8
<interface id> e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0
<string binding> ncacn_ip_tcp:16.32.192.223[135]
<annotation> Endpoint Resolution
```

Note that entries for COA and CFGPUSH are not displayed.

## SYSTEM-F-NOLOGNAM Error When Executing OVO Commands

If the environment that the OVO Agent requires has not been properly set up, you might see a SYSTEM-F-NOLOGNAM error when attempting to use an OVO command; for example:

```
$ opcagt -start
```

```
%SYSTEM-F-NOLOGNAM, no logical name match
```

This error occurs when the logical names required by the agents have not been defined. You can solve this problem in either of the following ways:

- Use the command procedure SYSS\$STARTUP:OVOS\$STARTUP.COM to start the agents the first time after a system reboot. This command procedure ensures that the logicals are defined and that DCE/RPC has been started. For more information, refer to “Installing Agent Software on an OpenVMS System” on page 6.
- Define the logicals by invoking SYSS\$STARTUP:OVO\$DEFINE\_LOGICALS.COM. If you choose this method, you must also make sure that DCE/RPC is started. For more information, please see Step 4 in “Configuring the Management Server on a UNIX System” on page 4.

## OVO Agent Processes Start But Then Stop Soon Afterward

If the OVO Agent processes start but then stop soon afterward, one possible cause is that the control agent process was unable to communicate with the Management Server using DCE/RPC. If this is the case, information similar to the following is displayed in OVOS\$LOGS:OVOS\$OPCCTLA.LOG:

```
Error opcctla.exe Control Agent) (2380): Initialize of the VPO Control Agent failed.
(OpC30-1036).
```

```
Can't lookup servers: Connection request rejected (dce / rpc). (OpC20-108)
```

```
Stopping all VPO Agent processes... (OpC30-1192)
```

If you started the OVO Agents using the command procedure SYSS\$STARTUP:OVOS\$STARTUP.COM, check the output from that procedure to see if there were any problems starting DCE/RPC.

If you started the OVO Agents using the `opcagt -start` command, DCE/RPC might not yet be started on your system. In this case, you can either use SYSS\$STARTUP:OVOS\$STARTUP.COM to start both the DCE/RPC and the OVO Agents, or you can start DCE/RPC separately using the following command:

**\$ @SYSSMANAGER:DCE\$RPC\_STARTUP**

### Cannot Open Message Catalog

If you are running a program that calls VPO API routines and have not called the OpenVMS initialization routine OVOSVMS\_SPI\_INIT, the following error will be displayed:

Error occurred during the initialization of ITO.

Internal error: Cannot open message catalog  
/opt/OV/lib/nls/C.opcagt.cat

---

## Logfile Template Supplied for OpenVMS Agents

Table 5 shows a logfile template that is supplied for OpenVMS agents.

**Table 5** Logfile Template on OpenVMS Managed Nodes

Logfile	Description	Template Name
SYS\$MANAGER:OPERATOR.LOG	Operator log	Operator Log

---

## No SNMP Event Interception

The OVO event interceptor is **NOT** supported on OpenVMS managed nodes.

---

## Types of Default Monitored Objects

Table 6 describes the default monitored objects supplied by the OpenVMS SPI. Refer to the *HP OpenView Smart Plug-In for OpenVMS User's Guide* for additional monitored objects.

**Table 6** Object Thresholds on the Managed Nodes

Object	Description	Threshold	Polling Interval (Minutes)
cpu	Monitors CPU utilization.	80%	1 min. (calc. over 10-min.period)
memory	Monitors memory utilization.	80%	Calc. every minute

**Table 6 Object Thresholds on the Managed Nodes (Continued)**

Object	Description	Threshold	Polling Interval (Minutes)
page-file	Monitors page-file free-space utilization.	70%	Calc. every minute
swap-file	Monitors swap-file free-space utilization.	70%	Calc. every minute
LAN device	Monitors LAN device utilization.	40%	Calc. every minute
disk	Monitors disk utilization: <ul style="list-style-type: none"> <li>• /DIO - number of direct I/Os</li> <li>• /QUEUE_LENGTH of direct I/Os</li> </ul>	<ul style="list-style-type: none"> <li>• def.:100</li> <li>• def: 2</li> </ul>	<ul style="list-style-type: none"> <li>• per second</li> <li>• per second</li> </ul>

---

## The OpenVMS Directory Structure

The directory structure on OpenVMS is very similar to the structure that OpenView agents use on UNIX nodes. On UNIX systems, the directory tree begins at the UNIX root. On OpenVMS, the logical name OVO\$ROOT specifies the root where the directory tree begins. OVO\$ROOT is a rooted logical name that points to \$\$SYSSPECIFIC:[OVO]. For example, on UNIX systems, the directory containing the agent configuration files is referred to as /var/opt/OV/conf/OpC. On OpenVMS systems, this directory is OVO\$ROOT:[VAR.OPT.OV.CONF.OPC].

You will probably not need to find files in this directory tree. However, if you need to navigate through the OVO tree, one way is to set your default to the top of the OVO tree. Because OVO\$ROOT points to a rooted directory, you can use the logical only by including part of the directory tree with it. You can use the usual OpenVMS notation for the top of the directory tree: (OVO\$ROOT:[000000]). Alternatively, the logical OVO\$HOME points to the top of the tree, which allows you to use a slightly shorter command such as:

**\$ SET DEFAULT OVO\$HOME**

To further simplify navigation in this directory tree, you can use other logicals that have been defined; for example:

**OVO\$SYSTEM**

**OVO\$CONF**

**OVO\$LOGS**

Finally, if GNV is installed, you can use either the UNIX path names in GNV commands such as **cd** and **ls**, or OpenVMS path names in DCL commands such as **SET DEFAULT** and **DIRECTORY**. For example, to see when log files have been created, you can use either of the following commands:

**\$ ls -1 /var/opt/OV/log/OpC**

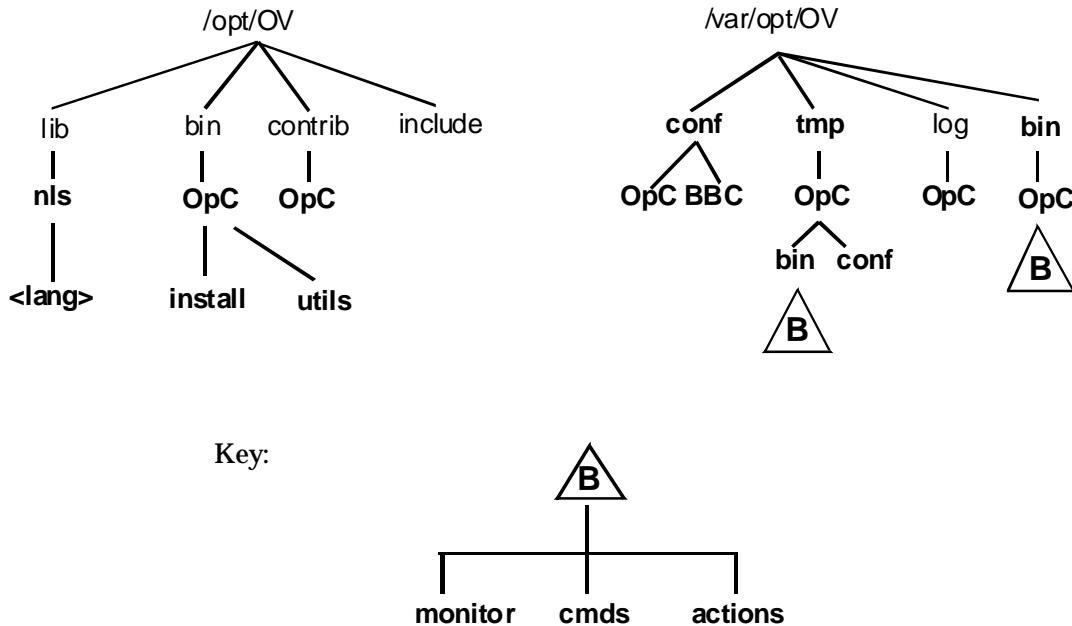
**\$ DIRECTORY/DATE OVO\$ROOT:[VAR.OPT.OV.LOG.OPC]**

---

## Organization of Managed Nodes

Figure 1 shows how OpenView software is organized on OpenVMS managed nodes. The trees are shown from both OpenVMS and UNIX perspectives.

**Figure 1** OVO Software on OpenVMS Managed Nodes



### Location of Files

This section explains where various files are located on an OpenVMS managed node. Each path is specified as both an OpenVMS and a UNIX path name.

On OpenVMS managed nodes, files are located as follows:

- ❑ **Process-related Files**  
OVOS\$ROOT:[VAR.OPT.OV.TMP.OPC]
- ❑ **Agent Configuration Files**  
OVOS\$ROOT:[VAR.OPT.OV.CONF.OPC]
- ❑ **Log Files and OPCerror**  
OVOS\$ROOT:[VAR.OPT.OV.LOG.OPC]
- ❑ **OpenVMS Command Procedures**

This section describes command procedures that are supplied with the OpenVMS agent for an OpenVMS system manager.

---

**NOTE**

Those familiar with OVO Agents on a UNIX platform might be more comfortable using the OVO Agent UNIX commands supported on OpenVMS. To use these commands, you must first invoke some OpenVMS command procedures to set up the environment properly. The actions required are listed within the following command procedure descriptions.

---

— **SYSSSTARTUP:OVO\$STARTUP.COM**

Use this command procedure to start the OVO Agents. You can invoke it interactively or include it in the system startup command procedure SYSSSTARTUP:SYSTARTUP\_VMS.COM to automatically start the agents at each system startup.

Alternatively, you might prefer to use UNIX-style commands. If so, ensure that DCE/RPC is running, and then enter the following command:

**\$ opcagt -start**

— **SYSSSTARTUP:OVO\$SHUTDOWN.COM**

Use this command procedure to stop the OVO Agents. If you prefer to use UNIX-style commands, enter the following command:

**\$ opcagt -kill**

— **SYSSSTARTUP:OVO\$DEFINE\_LOGICALS.COM**

This command procedure defines the logical names that you must set up before using the product. OVO\$STARTUP takes care of these definitions.

If you prefer to use UNIX-style commands to start the agents, make sure this command procedure is invoked first: SYSSSYSROOT:[SYSMGR]OVO\$DEFINE\_COMMANDS.COM.

The easiest way to do this is to invoke this command procedure in the system startup procedure.

— **SYSSMANAGER:OVO\$DEFINE\_SYMBOLS.COM**

This command procedure defines the OVO Agent commands supplied by the OpenVMS agent. Some of the commands supplied are the following:

**opcagt**

**opcqchk**

**opcdcode**

**opctemplate**

These commands are modelled after the commands supplied by OVO agents on UNIX platforms.

Each user who wants to use these commands must invoke the OVO\$DEFINE\_SYMBOLS command procedure. The easiest way to do this is to include the command procedure in your LOGIN.COM file.

— **SYSSMANAGER:OVO\$CONFIG.COM**

This command procedure configures the OVO Agent. The only item currently configured is the name of the management server. OVO\$STARTUP.COM ensures that this is done before startup. If you need to change the name of your management server later, you can run this command procedure again.

If you want to use UNIX-style commands to start the product, make sure this command procedure is invoked before the first `opcagt -start` command.

## Types of Libraries

OpenView for OpenVMS provides API routines in the shared image `SYSS$SHARE:OVO$LIBOPC_R.EXE`. (This is similar to the library `libopc_r.so` provided by OpenView on most UNIX platforms.)

Table 7-1 describes the managed node shared image for `OVOA.07.xx`.

**Table 7-1 Managed Node Shared Image for OVOA.07.xx**

DCE	Shared Image	OVO\$LIBOPC_R.EXE
	Shared image linked to the OVO shared image	SYSS\$LIBRARY:DCE:LIB_SHR.EXE SYSS\$LIBRARY:DTSS\$SHR.EXE SYSS\$LIBRARY:CMA\$OPEN_LIB_SHR.EXE SYSS\$LIBRARY:CMA\$LIB_SHR.EXE SYSS\$LIBRARY:CMA\$OPEN_RTL.EXE SYSS\$LIBRARY:CMS\$RTL.EXE
	Linker options in options file	SYSS\$SHARE:OVO\$LIBOPC_R/SHARE

### About the Include File

To compile programs that use the API, use the following include file:

**OVO\$ROOT:[opt.OV.include]opcapi.h**

## Tips for Compiling, Linking, and Executing Files

These tips are intended to help you compile, link, and execute programs that call API routines.

- Programs calling APIs must include a call to the routine `OVO$VMS_SPI_INIT` before calling an API routine. If the program does not call this routine, when you run it, the program displays an error indicating that some of the files in the directory structure cannot be found; for example:

```
Error occurred during the initialization of ITO.
```

```
Internal error: Cannot open message catalog /opt/OV/lib/nls/C/opcagt.cat
```

- The compile command requires the qualifier `/NAMES=AS_IS`. Without this qualifier, C converts the `opc` API calls to uppercase, and they cannot be found at link time.

- The program must be linked against the shared image OVO\$LIBOPC\_R.EXE. You can do this by including the following line in an options file:

**SYSSSHARE:OVO\$LIBOPC\_R/SHARE**

- The program must be run from the SYSTEM ACCOUNT.

