

# **HP OpenView Console Manager's Guide**

**HP 3000 MPE/iX Computer Systems**

**Edition 2**



**Manufacturing Part Number: B3118-90012  
E0895**

U.S.A. August 1995

---

## **Notice**

The information contained in this document is subject to change without notice.

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

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

---

## **Restricted Rights Legend**

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

---

## **Acknowledgments**

UNIX is a registered trademark of The Open Group.

Hewlett-Packard Company  
3000 Hanover Street  
Palo Alto, CA 94304 U.S.A.

© Copyright 1992 and 1995, by Hewlett-Packard Company

---

# Contents

## 1. Introducing OVC

Introducing HP OpenView Console.....	14
Overview .....	14
System Requirements .....	15
System Requirements for the OVC PC .....	15
System Requirements for the MPE/iX.....	16

## 2. Installing OVC

Overview .....	18
Configuring the OVC PC Software .....	19
Configuring the HP 3000 Software .....	21
Where to Go from Here .....	22

## 3. OVC Building Blocks

Overview .....	24
A Close Look at OVC Events .....	25
Event Processing on the HP 3000 .....	26
EML Report Generation .....	28
HP 3000 to OVC PC Communication Link.....	29
Event Processing on the OVC PC .....	30
Clearing Map Symbol Colors .....	31
The Management Session Interface (MSI) .....	32
SysMgr Files for OVC.....	33
Group EMS on the Managed Node.....	33
Group SECURE on the Managed Node — For Automated Response .....	35
Group EMSKIT on the Management Node — Stores All Managed Node Software .....	35
Group COMMON on the Management Node — Used by JCOMMON Job .....	37
Group PUB on the Management Node — Common Services .....	37
Group SYSMGR on the Management Node .....	38

## 4. Customizing Your Map and System Events

Overview .....	40
Generating an Event.....	40
Syntax .....	41
Matching Events with Map Symbols.....	43
How to Use the MPE00DEF File .....	43
Syntax .....	44
How to Tailor Events Using the USER0DEF File.....	44
How Customized Changes Take Effect .....	45
Customizing Map Symbols .....	50
Adding Your Private Map Symbols .....	52

---

# Contents

## 5. OVC Utilities

Overview .....	54
NTC00EXE.EMS.SYSMGR.....	55
NETALLOW.EMS.SYSMGR.....	56
VERCHECK.EMS.SYSMGR.....	57
REMOVEXE.COMMON.SYSMGR.....	58
DBEMLEXE.COMMON.SYSMGR.....	60
USEREXE.COMMON.SYSMGR .....	61
CONVEXE.COMMON.SYSMGR .....	62
SMUTIL.PUB.SYSMGR .....	64
Syntax.....	64
Required Files.....	65
Main Menu .....	65
VERCHECK.PUB.SYSMGR.....	68

## 6. Troubleshooting Problems

Overview .....	72
Version Checking.....	73
Troubleshooting Techniques .....	74
Troubleshooting the HP 3000 Module.....	74
Troubleshooting the OVC PC Module.....	74
Frequently Asked Questions.....	76
Question: Why do some symbols on the map remain blue?.....	76
Question: How do I remove all events logged against OVC symbols without using the Event Browser to delete every object individually?.....	76
Question: Why are bitmaps in OVC windows misaligned? .....	76

### A. Using the NMEVENT API

The NMEVENT API .....	80
Using the NMEVENT API .....	81
Syntax.....	81
Parameters.....	82
Return Values.....	84
Programming Examples .....	85
Programming Example with Pascal (Modcal).....	85
Programming Example with SPL.....	87

### B. Console Setup Script

Overview .....	90
CONSETUP.RCL Script Logic.....	90

### C. Customizing the OVC PC Environment

Overview .....	94
C:\OV\OVWIN.INI .....	94
C:\OV\XOVSVR.INI .....	95
C:\OV\SYSMGR\SYSMGR.INI .....	96

## **D. Upgrading from OVC to HP OvenView SysMgr**

Overview .....	102
Upgrading Your OVC PC Software .....	103
Upgrading Your OVC HP 3000 Software .....	104
Completing the Upgrade .....	104



---

## Figures

Figure 2-1. HP OpenView Window . . . . .	19
Figure 2-2. HP OpenView Window Displaying Default CONSOLE Map . . . . .	20
Figure 2-3. Pop-Up Menu with Describe... Menu Item . . . . .	20
Figure 3-1. EML Database Archive and Related File Relationships . . . . .	27
Figure 3-2. OpenView Alarm Setup Dialog Box . . . . .	31
Figure 4-1. DEF0LIST Listing Example (Part 1) . . . . .	46
Figure 4-2. DEF0LIST Listing Example (Part 2) . . . . .	47
Figure 5-1. Example of DELLIST Contents. . . . .	58
Figure 5-2. Example of Deleting Trace Files Using REMOVEXE . . . . .	59
Figure 5-3. SMUTIL Main Menu (Part 1) . . . . .	66
Figure 5-4. SMUTIL Modify Configuration Database Menu (Part 2) . . . . .	67
Figure 6-1. Dialog Box Showing OVC PC Program Version Report . . . . .	73



---

## Tables

Table 1-1. OVC PC Hardware Requirements. . . . .	15
Table 1-2. OVC PC Software Requirements. . . . .	15
Table 1-3. OVC HP 3000 Requirements. . . . .	16
Table 4-1. Message Severity for Map Symbols. . . . .	41
Table 4-2. Application to Icon Mapping . . . . .	48
Table 5-1. OVC Utility Programs . . . . .	54
Table 5-2. CONVEXE PARM Values . . . . .	62
Table 5-3. INFO Syntax Options . . . . .	64
Table 6-1. OVC PC Error Codes . . . . .	75
Table A-1. NMEVENT Parameters . . . . .	82
Table A-2. NMEVENT Values. . . . .	84



---

## Preface

Read this manual if you are responsible for the day-to-day operation of an HP 3000 system managed by an OpenView Console. You should have knowledge of HP 3000 systems and utilities of the MPE/iX operating system, including the commands required for operation.

Since HP OpenView Console (OVC) delivers its functionality through Windows, you should be familiar with the Windows user interface. The operating system for your PC is MS-DOS so you should also be familiar with DOS concepts and utilities. For more information, refer to the manuals provided by Microsoft.

The *HP OpenView Console User's Guide* introduces you to some general HP OpenView concepts. For more information, refer to the *HP OpenView for Windows User's Guide*.

It is also important that you are familiar with PC-related hardware issues. For more information about PC concepts and troubleshooting, refer to your PC installation guide.



This chapter introduces HP OpenView Console (OVC) and provides information about hardware and software requirements for your reference.

## Introducing HP OpenView Console

HP OpenView System Manager (referred to as SysMgr throughout this guide) and HP OpenView Console (referred to as OVC) are HP applications that provide a single integrated system management solution. SysMgr provides the integrated solution used to manage *multiple* HP 3000 computers in a network. OVC is essentially a subset of SysMgr and provides the local console and GUI to manage a *single* MPE/iX system. Since OVC is a subset of the full SysMgr solution, much of the information in this guide is extracted from the *HP OpenView System Manager Manager's Guide*.

### Overview

The OVC PC is the console for your HP 3000. You access the HP 3000's console port (ldev-20) through the WRQ Reflection terminal emulation window. By using a terminal emulator, you have added features, such as scripting and logging. In addition to ldev-20 port access, OVC provides immediate status updates for the system components of your HP 3000 through the HP OpenView map.

By using the principle of *management by exception*, OVC increases operational efficiency, freeing operators from monitoring every system console message. Instead, operators are notified only about problems that are truly important by color changes on the HP OpenView map. You can also automate responses to events that require a constant reply. Using OVC, you can monitor applications that are HP-supplied, user-written, or third-party.

OVC uses a client/server design that includes software that resides on both the OVC PC and HP 3000:

#### **OVC PC software**

This software provides the graphical user interface (GUI.) Additional OVC PCs can be added. However, each PC requires a valid OpenView software license.

#### **HP 3000 software**

This software resides on an MPE/iX system and provides the ability to collect console events and perform automated response. It also provides a common event repository and supports the messaging link with the OVC PC.

## System Requirements

Most users will receive the OVC PC bundle shipped with the MPE/iX system. The OVC software is pre-installed, and your local HP support engineer will set up the OVC software for you. The system requirements information listed in the next two sections is for your reference only.

### System Requirements for the OVC PC

Table 1-1 lists the minimum and recommended hardware requirements for the OVC PC.

**Table 1-1 OVC PC Hardware Requirements**

Minimum	Recommended
386/25 PC	486/66 PC
8 MB memory	16 MB memory
540 KB conventional memory	N/A
20 MB free disk space	40 MB free disk space
14-inch VGA color monitor	17-inch VGA color monitor
LAN adapter card	N/A
High-speed serial port (COM port should use the 16550AF UART chip.)	N/A

Table 1-2 lists the software requirements for the OVC PC.

**Table 1-2 OVC PC Software Requirements**

Software package	Version
MS-DOS	5.0 or higher
Microsoft Windows or Microsoft Windows for Work Groups	3.1 3.11
HP OpenView for Windows <sup>a</sup> SNMP platform or Workgroup Node Manager	7.2 or higher 1.0 or higher
HP OpenView OVC PC software	B.00.01 or higher
WRQ Reflection 1 for Windows	4.10 or higher
WRQ NS Open	4.0 or higher

a. PC/TCP (a TCP/IP stack from FTP, Inc.) is bundled with HP OpenView for Windows.

## System Requirements for the MPE/iX

Table 1-3 lists the requirements for the HP 3000 that will run the OVC software.

**Table 1-3** OVC HP 3000 Requirements

Requirement	Version
HP 3000 series 9xx system	N/A
HP MPE/iX	4.0 or higher
HP NS services	N/A
HP TurboIMAGE database	N/A
OVC HP 3000 software <sup>a</sup>	A.01.03 or higher

- a. Use the VERCHECK.PUB.SYSMGR program to extract current version information.

Resources for two batch jobs are required to run the OVC HP 3000 software. Running this software requires a minimum of 100,000 sectors of free file space. Disk usage is proportional to the size of the Event Message Log (EML) database (the Image database) and the number of copies of the EML database you keep. For more information refer to Chapter 3 , “OVC Building Blocks.”

This chapter gives you step-by-step instructions on how to configure the OVC software on your PC and the HP 3000.

## Overview

Before using OVC, you need to configure the OVC software. You can do this in a two-part procedure:

- OVC PC software configuration
- HP 3000 software configuration

The next sections provide the necessary steps to complete the configuration.

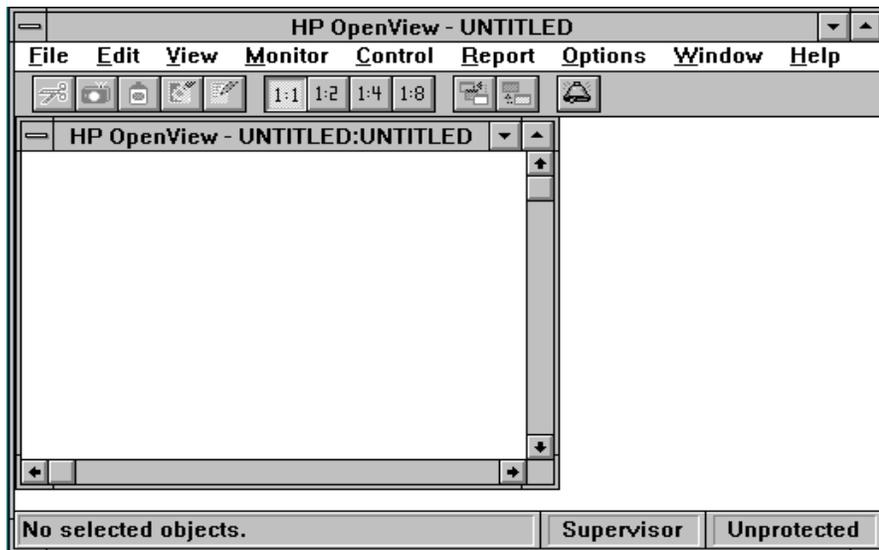
---

## Configuring the OVC PC Software

Follow these steps to configure the pre-installed OVC PC software:

1. Double click on the HP OpenView icon in the HP OpenView program group to start the OVC software. The HP OpenView window is displayed (see Figure 2-1).

**Figure 2-1** HP OpenView Window



2. Click on the Open menu item from the File menu, and load the default OVC map file, `console.ovm`, from the `c:\OV\SYSTEMGR\MAP` directory. The default SYSTEM map is displayed (see Figure 2-2).

**Figure 2-2 HP OpenView Window Displaying Default CONSOLE Map**



3. Click on the **Customize HP OpenView** menu item from the **Options** menu.

The **Customize HP OpenView** dialog box is displayed.

4. Type `c:\OV\SYSMGR\MAP\CONSOLE.OVM` in the **Default Map** text box. Click **OK**.

This ensures that `console.ovm` is loaded whenever you restart HP OpenView.

5. Double-click on the **SYSTEM** symbol in your top-level map to display the second-level task map. Then right-click on the center computer system symbol to display the pop-up menu (see Figure 2-3).

**Figure 2-3 Pop-Up Menu with Describe... Menu Item**



6. Click on the **Describe...** menu item. The **Describe Large Computer** dialog box is displayed.
7. Type the fully qualified NS name of your HP 3000 in the **Name** text box, and then type the node name only in the **Label** text box. Click on the **OK**.
8. Click on the **Save** menu item from the **File** menu in the HP OpenView window to save your map.

---

## Configuring the HP 3000 Software

Follow these steps to configure the pre-installed HP 3000 software:

1. Double-click on the Console Setup program icon in the HP OpenView program group to start the Reflection configuration script. The `Reflection 1` dialog box will be displayed.
2. Type the full NS node name of your HP 3000 system, and click on the OK button. The `MPE logon` dialog box is displayed.
3. Type the MPE logon string for `MANAGER.SYS` in the text box, and click on the **OK** button.

After you complete the logon, the script takes a few moments to configure your OVC HP 3000 files for use.

4. When the script is completed, a dialog box is displayed. Click on **Yes** if you want the OVC software to be launched on the HP 3000. The OVC HP 3000 is launched by streaming the `JCOMMON.COMMON.SYSMGR` job.

After the job is streamed, the OVC software on the PC should start posting events to the icons when they occur. You can test this by issuing this command on the HP 3000:

```
TELLOP NMEV#200@223 Test event for Disks.
```

After you issue this command, you should receive a yellow (warning) event for the disk symbol.

---

### NOTE

After starting the OVC HP 3000 software, a `SHOWJOB` listing should show the `JCOMMON` job and the `EMG` job.

---

## Where to Go from Here

At this point your OVC PC and HP 3000 should be configured for you to use.

The remaining chapters and appendices contain information about OVC that enables you to more fully understand OVC, customize your OVC setup, as well as support and troubleshoot OVC problems.

Chapter 3 , “OVC Building Blocks,” gives information about the OVC components.

Chapter 4 , “Customizing Your Map and System Events,” gives information about customizing your setup.

Chapter 5 , “OVC Utilities,” gives information about several OVC utilities.

Chapter 6 , “Troubleshooting Problems,” gives information about troubleshooting common OVC problems.

The appendices outline other software tools and provide information you may need while administering OVC either on the PC or the HP 3000.

Refer to the *HP OpenView Console User's Guide* to learn how to use OVC.

This chapter provides a technical description of the OVC building blocks.

## Overview

OVC provides an integrated system management solution for a single system environment, using a client/server design. This chapter explains in detail the characteristics applied to this design including: event operation (event collection, event delivery, event storage) and the OVC PC map update. The management session interface (MSI) is also introduced. Also the entire list of SysMgr files for OVC is provided as a reference.

## A Close Look at OVC Events

OVC is an event-reporting system that alerts you when important events occur on your managed HP 3000 machine. Through a special interface with the MPE operating system, OVC receives console messages as displayed on the local console. For console messages to be viewed as events by the OVC event collection agent, they are *instrumented*, meaning that a special event marker is added to each console message to specify the severity and the originator of the message.

MPE uses catalog files to store standard console messages. Important messages in most of these catalog files are already instrumented. Other than modifying the system message catalog, an application can invoke the MPE TELLOP command (from a job file), or the MPE PRINTOP intrinsic (from a program file) to post OVC events. By invoking the OVC NMEVENT intrinsic, user applications can post events directly to the OVC event collection agent. For more information about the event marker format, system catalog file, and the NMEVENT intrinsic, read Chapter 4 , “Customizing Your Map and System Events.”

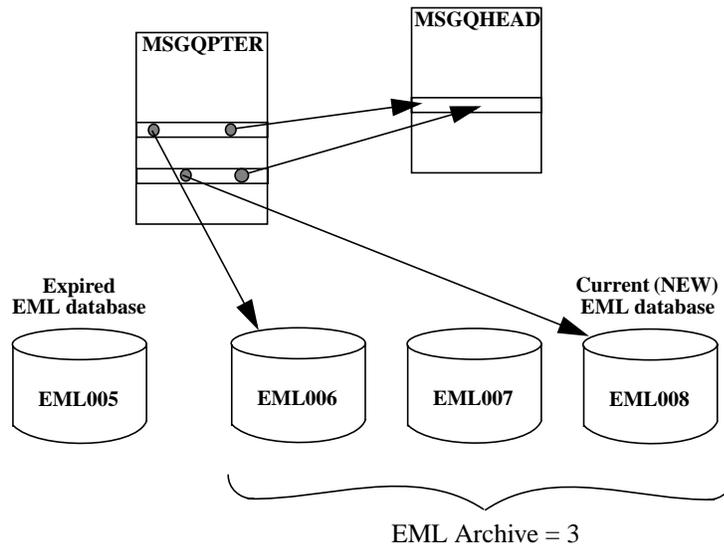
## Event Processing on the HP 3000

OVC is designed as an open system product that provides services for third party applications. The COMMON group stands for the common services. Programs in this group provide the user logon, PC network link creation, event depository, and user notification. The root program of the JCOMMON job is SYSEXE.COMMON.SYSMGR. This program waits for input from the EML0000X.COMMON.SYSMGR message file, and any application can be programmed to send EML event messages to this message file.

Once a new EML event message arrives, the SYSEXE program writes it into the EML database, sets up the review queue entry, and optionally alerts the current PC users. The EML database is an IMAGE database of fixed size. Once a database is full, a *switch log* operation is required to generate a new database. Since multiple EML databases are open at the same time, a consistent record search index is required across all open EML databases to facilitate event browsing. The SYSEXE process uses two queue files to store this search index: MSGQPTER and MSGQHEAD. Both files reside in the COMMON.SYSMGR group. The MSGQHEAD file keeps one record for each *known object*, and the MSGQPTER file keeps one record for each event message. Once the SYSEXE process receives an event message that carries an unknown (new) object name and object type, it adds a new entry to the MSGQHEAD file so this object is known (see Figure 3-1). With this design, any application can send events with their designated object name and object type. When the JCOMMON job starts, these two queue files are created automatically, if the files do not yet exist. If the two queue files already exist, SYSEXE rebuilds them with the existing content to ensure they are consistent. As long as you have enough disk space, the two queue files extend themselves automatically to accommodate new events. During the SYSEXE initialization stage, the SYSEXE program shrinks the queue files if too much memory has been allocated.

Acknowledging or removing an event means that you remove the event pointer, which points to the message in the EML database from the MSGQPTER file. Once an event is acknowledged, you cannot see it in the event browser, but the message still remains in the EML database. You can use the EML Report feature to generate a report that includes the message.

**Figure 3-1 EML Database Archive and Related File Relationships**



If you have enough disk space, you can keep a maximum of 99 copies of the EML database on your disk, ranging from EML001 to EML099. The EML database capacity is a user-defined value. You can modify the value using the OVC Configure EML Database menu item from the Customize HP SysMgr submenu of the Options menu interface on the OVC PC. The *EML archive* is another value that you can modify using the same interface. These two values control the EML switch log operation. Once the current EML database reaches the high water mark (the default value is 95% full), the SYSEXE program triggers the EML switch log operation. You can customize this high water mark by adding a JCW in the JCOMMON.COMMON.SYSMGR job file. For example, use this entry:

```
:setjcw EMLLIMIT=80
```

This entry sets SYSEXE to trigger the EML switch log operation when the database is 80% full. During the switch log operation, a backup copy of the queue files is created in case the switch log is not successful. The backup queue files are BAKQPTER and BAKQHEAD. The most frequent failure is due to nonavailability of disk space.

The EML archive value indicates how many copies of the EML database should be kept. Once the new EML database is created, SYSEXE labels the oldest EML database outside the EML archive value as the *expired one*. The SYSEXE program removes all the event entries in the MSGQPTER file associated with this expired EML database. Be aware of this *auto purge* feature. Events in the “Folder” object are also subject to this rule and have a limited life span. You should tune your EML setup to assure that the switch log is not frequent and that the EML database provides the required amount of

event storage. You can force the EML switch log at any moment using the SMUTIL.PUB.SYS utility. This operation is necessary when you want to back up the EML databases. After a switch log operation, the SYSEXE program closes all the open EML databases and open only the newest one (the blank one) for new events. You can force the EML switch log by inserting this command line into your system backup job file:

```
!run smutil.pub.sys;info="SWITCH"
```

From a terminal session at the HP 3000, you can view the EML database entry using the SMUTIL.PUB.SYSMGR utility or just run the DBEMLEXE.COMMON.SYSMGR program directly. If you want to access the EML database directly with MPE QUERY, the database schema is in the EML00XSC file.

You are responsible for disposing of the expired EML database. At the end of the EML switch log operation, the SYSEXE program searches for a EMLSLTPL.COMMON.SYSMGR file. This file is a template for building the job file JEMLSWLG.COMMON.SYSMGR. You should edit this template to meet your local needs. The SYSEXE program builds the JEMLSWLG job file and streams it automatically for you. The default operation in this template is to purge the expired database.

## EML Report Generation

Since you know the schema of the EML database and IMAGE query tools are available, it may not be obvious why you need EML report generation tools. These query tools let you generate a report for a *single* EML database. This EML report generation feature lets you generate reports across *multiple* EML databases. Using the interface, you can specify the criteria used to select events and store the generated reports. You can send the report to the PC, let it reside on the HP 3000, or send it to the HP 3000's print queue.

Once the events are selected from the EML databases, the OVC report generation program, RPTEXE.COMMON.SYSMGR, uses the report template to generate the report. Using a report template is the most flexible way to produce a report. The OVC report template is in a text file format and contains the information for the layout of the cover page, bottom page, regular page, and each record. You can use standard MPE editing tools, such as EDIT or TDP, to produce this template. Be aware that this template file should be saved as an unnumbered file. A general rule to follow is to save all OVC template files used for various purposes in an unnumbered format. Two standard templates are shipped with the OVC HP 3000 software, FORM00PC and FORM00MN. You can find them in the COMMON.SYSMGR group. If you want to customize your EML report, refer to Chapter 4, "Generating Reports," in the *HP OpenView Console User's Guide*.

---

## HP 3000 to OVC PC Communication Link

The UCSEXE.COMMON.SYSMGR program controls the user logon from the OVC PC. This program is a son process of the SYSEXE program which waits on a network *well known port* registered for OVC PC use. This is the program that receives the connection request from the SysMgr Communication Manager (COMMMGR) on the OVC PC. The UCSEXE program also checks the OVC PC software interface version with the supported version defined in the VERSION.COMMON.SYSMGR file.

---

**NOTE** Do not remove or modify this file version unless instructed by an HP support engineer. The OVC interface version is a product internal version, a different value from the information shown by the VERCHECK utility, or the version listed in OpenView's About Window.

---

The CONFIG.COMMON.SYSMGR database, an IMAGE database with four data sets, stores all PC user logon information. You can modify those data sets using the *Customize HP SysMgr* menu item from the *Options* menu of the OVC PC. All the CONFIG data sets will remain closed unless a read or update is required. This design simplifies the effect of backing up the CONFIG database.

If the UCSEXE program satisfies the user logon check, it notifies SYSEXE to create the communication server at the host end. The SYSEXE program creates the OVSEXE process, which communicates with the SysMgr Communication Manager (COMMMGR) at the OVC PC. To ensure that the OVC PC is on line, the OVSEXE process sends out a "PING" message to the OVC PC at user-defined time intervals. The default value is 20 seconds. You can set up a JCW in the JCOMMON job stream to override this value by using this line:

```
! setjcw PINTIMER=30
```

This example changes the time interval to 30 seconds (using a zero value disables this feature).

If the OVC PC has the HP 3000's node name in the C:\OV\SYSMGR\SYSMGR.INI file, it always generates a NOCAP user logon with the HP 3000. The OVC PC software remembers the HP 3000 name as soon as a link is established successfully. The OVSEXE link is always operating so that your map symbols can receive their most current color. Once you select an OVC menu, the logon dialog box is displayed for you to enter your OVC user name and password. After this logon process the existing NOCAP link is closed, and a new OVSEXE link is created to provide the required message exchange.

## Event Processing on the OVC PC

The COMMMGR.EXE program, found in the C:\OV\SYSMGR directory, is the event receiver which posts events to the HP OpenView map. Once a new event is received by SYSEXE, it passes that information to COMMMGR on the OVC PC. This event contains an object type defined in the MPE00DEF file, but it is unknown to HP OpenView so a translation is required. The C:\OV\SYSMGR\COMMMGR.TBL file serves as the translation table. If the responding symbol is in the current map, its color changes to reflect this event posting.

When an event is posted, you can configure it to generate one of three effects:

### **An audible beep**

This provides an alert to the user.

### **An alert message**

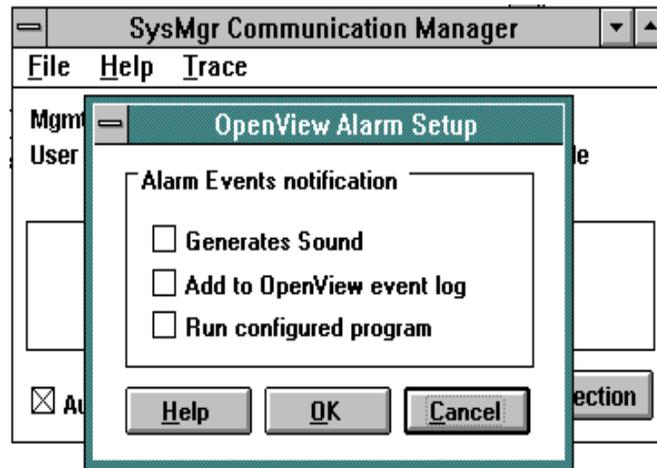
The message is inserted into OpenView's alarm log; the OpenView alarm symbol changes color to indicate an alarm exists. You must remove the alarm entry from the OpenView alarm log at a later time.

### **Trigger a preconfigured program**

This event posting will activate a preconfigured program. For example, it can wake up a program to perform beeper notification.

You use the COMMMGR.EXE program, which always appears as the SysMgr Communication Manager icon on your OVC PC screen, to configure the preferred effect. Double-click on the COMMMGR icon, then select the Alarm Setup menu item from the File menu. The OpenView Alarm Setup dialog box is displayed (see Figure 3-2).

**Figure 3-2** OpenView Alarm Setup Dialog Box



---

**NOTE** All settings are saved each time you change them.

### Clearing Map Symbol Colors

To *clear* a map symbol's color, you need to remove the pending review event from the SysMgr Event Browser. You can acknowledge the event immediately or move it to a folder symbol. The folder object works exactly like the ordinary object and has an entry in the MSGQHEAD file.

You can create as many folders as you want, but keep in mind that the events associated with them also have a limited life on the system. If you want to save an event, save it to a disk file. OVC is not a trouble-ticketing system, but with OVC PC software you can export OVC events to your trouble-tracking application.

## The Management Session Interface (MSI)

The Management Session Interface (MSI) in OVC provides a simple VT (Virtual Terminal) session to the HP 3000 on the OVC PC using Reflection 1 as the terminal emulator.

You can use the VT session to perform simple day-to-day HP 3000 maintenance tasks. You start the MSI session from HP OpenView. You click on the `SystemMgr MSI...` menu item from the **Control** menu, and a window named `Reflection 1 - MSI0.R1W` is displayed. (For more information on using MSI, refer to Chapter 2, "OVC Features," in the *HP OpenView Console User's Guide*.) You can launch only one MSI session at a time. If you need more VT sessions, then you can use Reflection 1 for Windows to open up more VT - MGR connections to the HP 3000 system.

---

## SysMgr Files for OVC

---

**NOTE** In these groups of SysMgr files used in OVC, whenever the word *managed node* is cited, it refers to the management node (the HP 3000).

---

OVC includes many SysMgr files. During its operation, a number of new files are created. Most of these files are temporary message files and can be purged after the product shuts down. No clean up facility is provided at the managed node, so all the message files generated at the managed node are listed here. The message files at the management node are not listed here since they are automatically removed by the REMOVEXE utility, which is run from the JCOMMON job.

### Group EMS on the Managed Node

<b>ARS00000.EMS.SYSMGR</b>	A message file for ARS00EXE to receive events from the EMG00EXE process.
<b>ARS00CMD.EMS.SYSMGR</b>	A message file for ARS00EXE to receive commands from the EMG00EXE process.
<b>ARS00EXE.EMS.SYSMGR</b>	A program file used to check the need for automated response.
<b>DEF0LIST.EMS.SYSMGR</b>	A text file used to show the logical object dispatching table.
<b>EMG00000.EMS.SYSMGR</b>	A message file used by EMG00EXE to receive messages from NTC00EXE.
<b>EMG0000X.EMS.SYSMGR</b>	A message file used by EMG00EXE to receive events from the automated response modules.
<b>EMG00EXE.EMS.SYSMGR</b>	A program file, the Event Message Gatherer.
<b>EMG00JOB.EMS.SYSMGR</b>	A job file used to run the EMG00EXE program.
<b>EMG00TPL.EMS.SYSMGR</b>	A text file used as a template file to build the EMG00JOB job file. Since the EMG00TPL file can be replaced during the managed node software update, you should not modify the EMG00TPL.
<b>EMGINIT.EMS.SYSMGR</b>	A CI script file used during the EMG00EXE start-up stage. It is executed by CI.PUB.SYS to set up the

required job environment. For example, you can set up file equations and CI variables that are required for your automated response operation.

<b>EML000X.EMS.SYSMGR</b>	A message file used by the EMS00??? program to receive events from the EMG00EXE process. It is not used by OVC.
<b>EMS00XL.EMS.SYSMGR</b>	A program file (a NM mode program), the Event Message Server transfers all events to EMMEXE at the management node. If this is a MPE/V system, the file name will be EMS00MPE (a CM mode program). It is not used by OVC.
<b>EMS00JOB.EMS.SYSMGR</b>	A job file used for running the EMS00??? program. It is not used by OVC.
<b>EMS00TPL.EMS.SYSMGR</b>	A text file used to build the EMS00JOB file.
<b>LAUNCHER.EMS.SYSMGR</b>	A program file used to build and launch the EMS00JOB file.
<b>MPE00DEF.EMS.SYSMGR</b>	A text file used to store required information to match a NMEV# marker with a map logical symbol. Do not modify this file.
<b>NETALLOW.EMS.SYSMGR</b>	A program file which grants access to privileged MPE commands.
<b>NTC00EXE.EMS.SYSMGR</b>	A program file that interfaces with EMG00EXE to display the console messages.
<b>RSP000XL.EMS.SYSMGR</b>	A program file, the NM version of the Automated Response engine.
<b>RSP00000.EMS.SYSMGR</b>	A message file used as the command input file for the ARS engine.
<b>RSP0000X.EMS.SYSMGR</b>	A message file used as the event input file for ARS engine.
<b>RSP000IN.EMS.SYSMGR</b>	A message file which acts as the STDIN file for the responding CI process.
<b>RSP000OUT.EMS.SYSMGR</b>	A message file which acts as the STDOUT file responding to the CI process.

<b>SMCAT000.EMS.SYSMGR</b>	A catalog file which acts as the message catalog used by the OVC managed node software modules.
<b>USER0DEF.EMS.SYSMGR</b>	A text file used to store required information to match a NMEV# marker with a map logical symbol. It is owned by the user, and the content will override definitions in the MPE00DEF file.
<b>VERCHECK.EMS.SYSMGR</b>	A program file which acts as the managed node version checking program.
<b>VRCL0G00.EMS.SYSMGR</b>	A circular file containing the operation log generated by the managed node program modules.
<b>VRCL0G01.EMS.SYSMGR</b>	A circular file containing the previous operation log. It is renamed from the VRCL0G00 when a new VRCL0G00 is created.
<b>VTC0000.EMS.SYSMGR</b>	A file used as the console message input file for the VTC00??? program.
<b>VTC000XL.EMS.SYSMGR</b>	A NM mode program file used to send the console messages to the management node. If this is a MPE/V system, the file name is VTC00MPE, a CM mode program.

### **Group SECURE on the Managed Node — For Automated Response**

<b>ARSLOG??.SECURE.SYSMGR</b>	The log files for ARS operation.
<b>EMG0AUTO.SECURE.SYSMGR</b>	A text file used as the rule file of the ARS operation.
<b>EXECUTE.SECURE.SYSMGR</b>	A CI script file, an example of an ARS script.
<b>NOTICE.SECURE.SYSMGR</b>	A CI script file, an example of an ARS script.

### **Group EMSKIT on the Management Node — Stores All Managed Node Software**

<b>ARS00EXE.EMSKIT.SYSMGR</b>	A program file used to check the need for automated response.
-------------------------------	---

- EMG0AUTO.EMSKIT.SYSMGR** A text file used as the rule file of the ARS operation.
- EMG00EXE.EMSKIT.SYSMGR** A program file, the root program for EMG00JOB job.
- EMG00TPL.EMSKIT.SYSMGR** A text file used as a template file to build the EMG00JOB file.
- EMGINIT.EMSKIT.SYSMGR** A script file used as the EMG00EXE program initialization file.
- EMS000XL.EMSKIT.SYSMGR** A program file, the NM mode version of the EMS program.
- EMS00MPE.EMSKIT.SYSMGR** A program file, the CM mode version of the EMS program.
- EMS00TPL.EMSKIT.SYSMGR** A text file used as a template file to build the EMS00JOB file.
- EXECUTE.EMSKIT.SYSMGR** A CI script file used as an example of an ARS script.
- GETRNEXE.EMSKIT.SYSMGR** A program file used by the managed node installation script to get the name of the managed node.
- JRNINST.EMSKIT.SYSMGR** A job file used in the previous release to pull files from the management node.
- LAUNCHER.EMSKIT.SYSMGR** A program file that builds and launches the EMS00JOB file.
- MPE00TPL.EMSKIT.SYSMGR** A text file used to store the required information to match a NMEV# marker with a map logical symbol. DO NOT modify this file.
- MPETEST.EMSKIT.SYSMGR** A program file used by the managed node installation script to check if the managed node is a MPE/V or a MPE/iX system.
- NETALLOWV.EMSKIT.SYSMGR** A program file, the CM mode version of the NETALLOW program.
- NETALLOWX.EMSKIT.SYSMGR** A program file, the NM mode version of the NETALLOW program.
- NOTICE.EMSKIT.SYSMGR** A CI script file used as an example of an ARS script.
- NTC00EXE.EMSKIT.SYSMGR** A program file used to interface with the EMG00EXE program to display the console messages.

**RSP000XL.EMSKIT.SYSMGR** A program file, the NM version of the Automated Response engine.

**RSP00MPE.EMSKIT.SYSMGR** A program file, the CM version of the Automated Response engine.

**SMCAT000.EMSKIT.SYSMGR** A catalog file, a message catalog used by OVC remote software modules.

**SMNMP1.EMSKIT.SYSMGR** A text file used as a dummy file place holder.

**VERCHECK.EMSKIT.SYSMGR** A program file, the managed node version checking program.

**VTC00MPE.EMSKIT.SYSMGR** A program file, the CM mode version of the VTC program.

**VTC000XL.EMSKIT.SYSMGR** A program file, the NM mode version of the VTC program.

### **Group COMMON on the Management Node — Used by JCOMMON Job**

**CONFIGSC.COMMON.SYSMGR** A text file used as the schema for the CONFIG database.

**CONVEXE.COMMON.SYSMGR** A program file used to build or convert the CONFIG database.

**DBEMLEXE.COMMON.SYSMGR** A program file activated alone or from SMUTIL.PUB.SYSMGR to view the entries in the EML databases.

**DELLIST.COMMON.SYSMGR** A text file read by the REMOVEXE program to find unwanted message files to remove from the system.

**EML00TPL.COMMON.SYSMGR** A text file used as the EML schema template by the SYSEXE program to perform the EML switch log operation.

**EMLSLTPL.COMMON.SYSMGR** A text file used as the template file for building the JEMLSWLG file.

**EML00XSC.COMMON.SYSMGR** A text file used as the schema for the EML database.

### **Group PUB on the Management Node — Common Services**

**GETMNEXE.PUB.SYSMGR** A program file used by the managed node installation script to get the name

	of the management node.
<b>I00I936N.PUB.SYSMGR</b>	A job file used to setup the OVC software at the management node.
<b>OVUDC.PUB.SYSMGR</b>	A UDC file required for OpenView console operation.
<b>SMUDC.PUB.SYSMGR</b>	A UDC file required for OVC product operation.
<b>SMUTIL.PUB.SYSMGR</b>	A program file used by the OVC management node software to allow menu driven tasks to be performed.
<b>TESTEVNT.PUB.SYSMGR</b>	A job file used to test the event path for OVC's default task oriented map.
<b>VERCHECK.PUB.SYSMGR</b>	A program file, the management node version checking program.

### **Group SYSMGR on the Management Node**

The files in this group are not used for the OVC product.

---

# 4

## **Customizing Your Map and System Events**

This chapter describes how you customize the HP OpenView map and your system events.

## Overview

Your OVC PC software comes with a default map called `console.ovm`. Typically you would use this default map as it is, except for changing the object name of the system symbol or your map from `nodename.domain.organization` to the actual node name for your HP 3000.

In the case where you have multiple OVC PCs monitoring the same HP 3000, you may want to customize the maps on each PC to suit the particular operator who will be monitoring it. To customize the maps, you must do the following:

- Delete task objects from the map that do not pertain to the operator's job
- Add tasks to be managed by OVC to the map (for example, user applications)

You can also add new applications that you want OVC to manage. These may be newly acquired applications or existing applications not currently managed by OVC. To add an application to the management domain, you must modify these elements:

- OpenView map on the PC
- USER0DEF file on the HP 3000 from which you receive events

To successfully make modifications, you need to understand the relationships between system events, map objects, and the object definition files that define and control these relationships. At the end of the chapter you will find instructions for adding your private map symbols. However, this chapter does not explain how to draw the map and add symbols. For information on building and modifying an OpenView map, refer to the *HP OpenView for Windows User's Guide*.

## Generating an Event

You can recognize (generate) an OVC event in one of two ways:

- By inserting a special marker into the console message
- By programmatically calling the NMEVENT intrinsic

---

### NOTE

A console message is a message issued using a TELLOP or PRINTOP, or is defined as one because it is in one of the MPE message catalogs.

The OVC event collector at the HP 3000 monitors all local console output. The event collector automatically receives console messages with the special event marker from the operating system through a special interface. The event marker provide this information:

- Message severity — dictates the map symbol color
- Message class — logically groups messages
- Application ID — identifies MPE subsystems and applications

The format of the event marker is:

**Syntax**

NMEV#pcc@aaa

where:

p	Message severity	(0–3)
cc	Message class	(0–99)
@	Separation character between message class and application ID	
aaa	Application ID	(0–320)

The @aaa portion is optional and, if omitted, has the default value of @310. Events with an application ID of 310 belong to the NO-ID symbol (defined in the MPE00DEF.EMS.SYSMGR file). The OVC software also generates events. These events have an application ID of 320 and appear against the computer symbol on your map.

The message class field, cc, further breaks down event information into subgroups (not used by MPE). This field is optional and, if omitted, defaults to a value of 00. The message severity (also called message priority) indicates the severity of the event, and affects the color of the map symbol to which the event is logged. The NMEV# pattern without severity is not a valid marker. The OVC event collector does not pick up console messages without a valid NMEV# marker.

Table 4-1 shows the relationship between the message severity and the map symbol color.

**Table 4-1 Message Severity for Map Symbols**

Message Severity Code	Status	Symbol
0	Archival	None
1	Informational	Magenta
2	Warning	Yellow
3	Critical	Red

You can issue local console messages in several ways. The MPE operating system uses message catalog files extensively to generate system messages. Most system catalogs come with a predefined message instrument. In OVC the term *instrumented* means adding a

special marker to the message text. You use the MPE GENCAT.PUB.SYS catalog utility to convert or merge catalog files. You can modify the catalog files (source format) with your favorite editor. For more information on the GENCAT utility or message catalogs, refer to the *MPE/iX Message Catalogs Programmer's Guide*. On MPE/iX systems, you use SYSGEN to make the system message catalog active.

You can use the MPE CI command TELLOP to issue console messages. The TELLOP method is useful for instrumenting production job files so they can notify you of completion or failure. To issue OVC events programmatically, you can use the MPE PRINTOP intrinsic. This intrinsic is good for short messages. For messages longer than 64 bytes, you can use the OVC NMEVENT intrinsic call. If you are interested in using this Application Programming Interface (API) in your application, refer to Appendix A , “Using the NMEVENT API.”

The OVC event gatherer also supports the INCS# marker, which has the same syntax as the NMEV# marker. The only difference between these two markers is that the INCS# remains in the local console output.

---

## Matching Events with Map Symbols

The definition (DEF) files on the HP 3000 system maps an applid/class/priority combination to an object id/object name pair (an object) on the map. OVC supports two DEF files:

- MPE00DEF.EMS.SYSMGR
- USER0DEF.EMS.SYSMGR

---

### NOTE

MPE00DEF should not be modified by users for any reason.

The USER0DEF file is provided for user customization. Changes to this file are applied on top of MPE00DEF definitions.

### How to Use the MPE00DEF File

The OVC event collector uses a logical object table to dispatch event messages. The term *message dispatching* means associating an event with a map symbol. The MPE00DEF file in the EMS group at the HP 3000 provides this linkage. The file contains two kinds of entries:

- Symbol declaration record
- Object definition record

This is a part of the symbol declaration record in the MPE00DEF file:

```
#52 MAIL                (HEX 23c9)
#53 DISK                (HEX 23C8)
#54 NETWORK            (HEX 23C7)
#55 DATABASE          (HEX 23C6)
#56 NO_ID_EVENT       (HEX 23C5)
```

This type of record defines a symbol type name for each map symbol. This is analogous to declaring constants in a program. These symbolic names are used in the object definition record, making the definitions easier to read. These two types of records can be mixed together as long as you follow a simple rule: you must define a symbol type name before you reference it. This is the syntax for a symbol declaration record:

```
#<SysMgr symbol ID><space><symbol type name>[<space><comment>]
```

Each entry must start with a “#” character. The SysMgr symbol ID must be a unique number. It is an error if two records define the same SysMgr symbol ID. One space character (and only one) is used to separate the SysMgr symbol ID and the symbol type name. It is an error if a symbol type name is longer than 16 characters. A comment can be added to the line after the symbol type name with at least one

intervening space. For example, the “(HEX ????)” is not part of the definition and indicates the matching “OpenView symbol ID.” Using two types of ID for a symbol requires ID translation at the PC. However, this design allows OVC to be ported to other management platforms in the future.

The format for a definition section entry is:

## Syntax

```
AAA-CC-PPPP<space>
```

where:

<b>AAA</b>	Application ID
<b>CC</b>	Message class
<b>PPPP</b>	A priority mask in which the left-most digit is archival, and the right-most digit is critical. For example, priority mask <i>0011</i> applies to events with a priority of warning or critical.

The symbol type name should be declared by some previous entry. The map object name should match the object name you entered for your map symbol using the `Describe` dialog box. Any part of `aaa-cc-pppp` can be omitted; but if it is missing, it is interpreted as “any value.”

These are two examples:

```
example 1:      217-- TAPE2:TAPE_OBJECT
example 2:      228--0001 PRINTER2:CRITICAL_PRINTER_OBJECT
                228--1110 PRINTER2:OTHER_PRINTER_OBJECT
```

The first example sends all the tape-related messages generated for Store/Restore symbol (regardless of class or severity) to the map symbol `TAPE_OBJECT`. The second example causes critical events from the High End Printer (ID 228) to appear against the `CRITICAL_PRINTER_OBJECT` map symbol, and all the non-critical events of the High End Printer to appear against the `OTHER_PRINTER_OBJECT` map symbol. From the explanation of these two examples, you should be able to read the `MPE00DEF` file.

## How to Tailor Events Using the `USER0DEF` File

If you do not like the definitions made by the `MPE00DEF` file, you can modify it *without editing it*.

---

### NOTE

Since you may lose your modified `MPE00DEF` file during the OVC HP 3000 software update, you should never make changes directly to the `MPE00DEF` file.

---

You create a USER0DEF file with required changes. The EMG00EXE process first loads the MPE00DEF file, then it searches for the USER0DEF file. If neither file exists, the EMG process does not build the logic object table. In this situation, all the OVC events are posted against the computer symbol representing the HP 3000 on the map. If you have task (component) symbols on your map, they will remain blue.

Based on the concept that a later entry can override an earlier entry, you can tailor this object dispatching table to meet your needs. For example, consider this entry in the MPE00DEF file:

```
228-- PRINTER1:PRINTER_OBJECT
```

This entry causes all printer-related events to be sent to the PRINTER\_OBJECT. If you want to change this so that only critical events are sent to this object while all non-critical events are sent to a new object called NON\_CRITICAL\_PRINTING, then you must add the following entry to the USER0DEF file.

```
228--1110 PRINTER1: NON_CRITICAL_PRINTING
```

If you want to eliminate the entry in the MPE00DEF file, you use this entry in your USER0DEF file to nullify it. The syntax is:

```
aaa-cc-ppp<space>VOID
```

The VOID keyword tells the EMG program to ignore a previous object definition record. For example, it nullifies the effect created in the previous example by appending the following entry to the USER0DEF file:

```
228--1110 VOID
```

Using the VOID\_MESSAGE keyword as the object name has a special meaning: discard the event message. Therefore, if you have this entry in your USER0DEF file:

```
228-- VOID:VOID_MESSAGE
```

the EMG process discards all the messages generated from the “High End Printer.” This is the most efficient way to discard events without using the Automated Response Service (ARS). For more information on ARS, refer to Chapter 5, “Automated Response Service (ARS),” in the *HP OpenView Console User’s Guide*.

## How Customized Changes Take Effect

As the previous examples show, the definition file provides a great amount of flexibility in defining how events appear on the map. So you have considerable control over customizing the management environment. But once you modify the USER0DEF file to meet your specifications, you must re-initialize the EMG process so that the changes can take effect. You do this by issuing this command using this syntax:

```
: tellop NMEV#INIT
```

When the changes take effect, you will see TELL messages from the EMG process. These messages indicate that the old logical object table has been removed and re-created. You can display the newly created table by printing out the DEF0LIST file in the EMS group. The file is very useful for displaying the combined effect of both definition files. Figure 4-1 and Figure 4-2 show an example listing of the DEF0LIST file created from a standard MPE00DEF file. In this example, the OVC PC is using a system-based map so every object name has the node name appended.

**Figure 4-1** DEF0LIST Listing Example (Part 1)

```
95/01/13 15:45:47:01 EMG00EXE [A0103017] #J0 [MPE00DEF.EMS.SYSMGR]

object type = 56 [NO_ID_EVENT]
object type = 55 [DATABASE]
object type = 54 [NETWORK]
object type = 53 [DISK]
object type = 52 [MAIL]
object type = 51 [TAPE1]
object type = 50 [PRINTER1]
object type = 84 [TAPE2]
object type = 83 [PRINTER2]
object type = 82 [SPOOLER]
object type = 81 [JOBS]
object type = 80 [USER_DEFINE3]
object type = 79 [USER_DEFINE2]
object type = 78 [USER_DEFINE1]
object type = 77 [CONSOLE_TERM]
object type = 76 [HARDWARE]
object type = 75 [PERFORMANCE]
object type = 74 [SECURITY]
object type = 73 [ENVIRONMENT]
object type = 72 [SYSMGR]
object type = 71 [MPE-IX]
object type = 70 [FOLDER]
```

**Figure 4-2 DEF0LIST Listing Example (Part 2)**

Appl_ID	Class	Pri_mask	Type	Object_name
298	ANY	1111	76	UPS_MONITORING_OBJECT_HPCSYN54
238	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
237	ANY	1111	74	SECURITY_OBJECT_HPCSYN54
236	ANY	1111	51	TAPE_OBJECT_HPCSYN54
235	ANY	1111	54	HOST-DTC-MGR_OBJECT_HPCSYN54
234	ANY	1111	54	HOST-DTC-MGR_OBJECT_HPCSYN54
233	ANY	1111	51	TAPE_OBJECT_HPCSYN54
232	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
231	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
230	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
229	ANY	1111	76	HARDWARE_OBJECT_HPCSYN54
228	ANY	1111	50	PRINTER_OBJECT_HPCSYN54
227	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
226	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
225	ANY	1111	76	HARDWARE_OBJECT_HPCSYN54
224	ANY	1111	77	LOGGING_OBJECT_HPCSYN54
223	ANY	1111	53	DISK_OBJECT_HPCSYN54
222	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
221	ANY	1111	77	LOGGING_OBJECT_HPCSYN54
220	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
218	ANY	1111	81	JOBS_OBJECT_HPCSYN54
217	ANY	1111	51	TAPE_OBJECT_HPCSYN54
216	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
215	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
214	ANY	1111	51	TAPE_OBJECT_HPCSYN54
213	ANY	1111	50	PRINTER_OBJECT_HPCSYN54
212	ANY	1111	50	PRINTER_OBJECT_HPCSYN54
211	ANY	1111	82	SPOOLER_OBJECT_HPCSYN54
206	ANY	1111	71	MPE/IX_OBJECT_HPCSYN54
200	ANY	1111	50	PRINTER_OBJECT_HPCSYN54
199	ANY	1111	54	HP-NMGR_OBJECT_HPCSYN54
198	ANY	1111	54	HP-NETWORK_OBJECT_HPCSYN54
197	ANY	1111	54	HP-NETWORK_OBJECT_HPCSYN54
196	ANY	1111	54	NS-NETWORK_OBJECT_HPCSYN54
195	ANY	1111	54	PUBLIC-NETWORK_OBJECT_HPCSYN54
194	ANY	1111	54	PUBLIC-NETWORK_OBJECT_HPCSYN54
55	ANY	1111	73	PREDICTIVE_SUPPORT_HPCSYN54
53	ANY	1111	55	DATABASE_OBJECT_HPCSYN54
52	ANY	1111	75	PERFORMANCE_OBJECT_HPCSYN54
51	ANY	1111	52	MAIL_OBJECT_HPCSYN54
50	ANY	1111	75	PERFORMANCE_OBJECT_HPCSYN54
310	ANY	1111	56	CONSOLE_MSG_OBJECT_HPCSYN54
:				

If an error occurs in the definition file, an OVC warning (yellow) event is generated against the HP 3000 computer symbol. You can find detailed error information in the event's annotation text. If everything goes well, the newly defined objects on the map turn green (from blue). You should see events generated against the new map object.

**NOTE**

Although changes to a particular user's map (such as deleting objects on the map) affect only the user at the OVC PC, changes to the USER0DEF file affect all users managing the system or tasks on the system where the definition files reside.

Table 4-2 lists the standard symbol mapping for MPE subsystems.

**Table 4-2 Application to Icon Mapping**

<b>Symbol Label</b>	<b>Subsystem</b>	<b>Appl ID</b>	<b>Object Name</b>	<b>Obj ID</b>
DTC	DCC Surrogate	235	HOST-DTC-MGR_OBJECT	54
DTC	Terminal ID	234	HOST-DTC-MGR_OBJECT	54
Database	Allbase/SQL	053	DATABASE_OBJECT	55
Hardware	SPU Switchover	225	HARDWARE_OBJECT	76
Jobs	Job/Sessions	218	JOBS_OBJECT	81
Logging	System Logging	221	LOGGING_OBJECT	77
Logging	User Logging	224	LOGGING_OBJECT	77
MPE/iX	Command Interpreter	230	MPE/IX_OBJECT	71
MPE/iX	Diagnostics-System	229	HARDWARE_OBJECT	76
MPE/iX	File system resiliency	215	MPE/IX_OBJECT	71
MPE/iX	I/O Services	206	MPE/IX_OBJECT	71
MPE/iX	Label Management	232	MPE/IX_OBJECT	71
MPE/iX	MPE/iX System and Error Management	231	MPE/IX_OBJECT	71
MPE/iX	Native Language Support	238	MPE/IX_OBJECT	71
MPE/iX	Process Manager	220	MPE/IX_OBJECT	71
MPE/iX	Reply Information Table	225	MPE/IX_OBJECT	71
MPE/iX	SYSMAN	227	MPE/IX_OBJECT	71
MPE/iX	Threshold Manager	216	MPE/IX_OBJECT	71
MPE/iX	Transaction Management	222	MPE/IX_OBJECT	71
Network	Network-NS	196	NS-NETWORK_OBJECT	54
Network	Network-OSI	185	HP-NETWORK_OBJECT	54
Network	Network-SNA	198	HP-NETWORK_OBJECT	54
Network	Public Networking	194	PUBLIC-NETWORK_OBJECT	54
Performance	Laser/Rx	052	PERFORMANCE_OBJECT	75
Printer	Ciper Device	200	PRINTER_OBJECT	50
Printer	High End Printe	228	PRINTER_OBJECT	50
Printer	Page Printer Device Manager	213	PRINTER_OBJECT	50
Printer	Page Printer	212	PRINTER_OBJECT	50

**Table 4-2            Application to Icon Mapping**

<b>Symbol Label</b>	<b>Subsystem</b>	<b>Appl ID</b>	<b>Object Name</b>	<b>Obj ID</b>
Security	MPE/iX Security	237	SECURITY_OBJECT	74
Spooler	Native Mode Spooler	211	SPOOLER_OBJECT	82
Spooler	Magneto-Optic Library	233	TAPE_OBJECT	51
Spooler	Labeled Tape	236	TAPE_OBJECT	51
Spooler	Print, Tape, Spooler, Console	214	TAPE_OBJECT	51
Spooler	Store/Restore	217	TAPE_OBJECT	51

---

## Customizing Map Symbols

This section explains how to add OVC symbols

to the OpenView toolbox. For information on how to draw or modify your OpenView map to meet your local needs (or if you have questions), refer to the *HP OpenView Windows User's Guide*.

---

### NOTE

You must enter all the map object names using uppercase. The OpenView object name is case sensitive, and OVC requires that all object names be specified in uppercase.

OVC provides many task symbols for managing MPE subsystems. OpenView uses icon files for map symbols. All the OVC icon files reside in the C:\OV\SYSMGR\SYMBOLS directory, and their file names start with *S0*.

All the OVC map symbols are defined in [OVSymbols] of the SYSMGR.INI file in the C:\OV\SYSMGR directory. This list shows all the entries in the [OVSymbols] section:

```
[OVSymbols]
4307=3 0 S0LCOMPR "sysmgr (L) system+"
4308=3 0 S0MCOMPR "sysmgr (M) system+"
4309=3 0 S0SCOMPR "sysmgr (S) system+"

23CF=2 -1 S0FOLDER "Folder (SysMgr)"
23CE=2 -1 S0MPE "MPE/iX (SysMgr)"
23CD=2 -1 S0SYSMGR "SysMgr (SysMgr)"
23CC=2 -1 S0ENV "Environment (SysMgr)"
4300=2 -1 S0SAFE "Security (SysMgr)"
4301=2 -1 S0METER "Performance (SysMgr)"
4302=2 -1 S0CHIPS "Hardware (SysMgr)"
4303=2 -1 S0CONS "Console (SysMgr)"

4350=2 -1 S0JOBS "Jobs (SysMgr)"
4351=2 -1 S0SPOOL "Spooler (SysMgr)"
4352=2 -1 S0PNTR2 "Printer_2 (SysMgr)"
4353=2 -1 S0TAPE2 "Tape_2 (SysMgr)"

23CB=2 -1 S0PNTR1 "Printer_1 (SysMgr)"
23CA=2 -1 S0TAPE1 "Tape_1 (SysMgr)"
23C9=2 -1 S0MAIL "Mail (SysMgr)"
23C8=2 -1 S0DISK "Disk (SysMgr)"
23C7=2 -1 S0NETWRK "Network (SysMgr)"
23C6=2 -1 S0DB "Database (SysMgr)"
23C5=2 -1 S0EVENT "Event (SysMgr)"
```

These fields have the following meaning:

- The first field is the SysMgr symbol ID.
- The numbers between the equal sign and the icon file name indicate the OpenView symbol such as compound symbols, computer symbols, and component symbol types.

- The text in quotes is the symbol description which will label the symbol icon in the toolbox.

In the table the first three symbols listed are compound symbols, and the remainder are component symbols.

The OVC PC program, COMMMGR.EXE, *posts* all the OVC events, meaning that the program needs to convert the SysMgr symbol ID to its matching OpenView symbol ID. The conversion table is stored in the C:\OV\SYSMGR\COMMMGR.TBL file, which is a text file that you can modify with a DOS text editor.

## Adding Your Private Map Symbols

Once you understand how OVC adds symbols to the OpenView toolbox, you can replace the icon files with your files. If you plan to keep your new icons in a different file, change the SYSMGR.INI file entry. If you are adding more symbols, add new OpenView symbol IDs to the file. Each OpenView symbol has a unique OpenView ID. If you require fewer than three new symbols, you can use a SysMgr pre-allocated symbol ID. You can locate these three reserved symbol IDs in the MPE00DEF file.

To add a new map symbol, draw the icon first. You should use “window color” as the background color for this icon so the background color can be changed for this symbol. Then modify the USER0DEF file. Add your private symbol declaration record in the USER0DEF file, keeping the same format that the file currently uses. On the OVC PC, add a new entry to the SYSMGR.INI and SYSMGR.TBL files for this new map symbol.

---

### NOTE

It is not recommended that every OVC customer create a large number of their own symbols. If you require a large number of additional symbols, send a request to HP. By doing so, HP can ensure that all new symbols have a common style, and that each symbol is included in future releases.

---

This chapter describes the various support programs that are included with this release of OVC.

---

## Overview

OVC has several utility programs that reside on the HP 3000. Table 5-1 lists the programs with a brief description.

**Table 5-1 OVC Utility Programs**

OVC Program	How Used
NTC00EXE.EMS.SYSMGR	To view the local console output
NETALLOW.EMS.SYSMGR	To obtain special MPE capabilities
VERCHECK.EMS.SYSMGR	To check the managed node's software version
REMOVEXE.COMMON.SYSMGR	To delete unwanted files
DBEMLEXE.COMMON.SYSMGR	To view the selected EML database
USEREXE.COMMON.SYSMGR	To manage the USER dataset
CONVEXE.COMMON.SYSMGR	To load the CONFIG database
SMUTIL.PUB.SYSMGR	To manage OVC software
VERCHECK.PUB.SYSMGR	To check the management node's software version

---

## NTC00EXE.EMS.SYSMGR

The NTC00EXE utility is a stand-alone MPE program used outside the OVC environment to view the console output. If you have the OVC software (EMG00JOB job) running at the HP 3000, every user on the HP 3000 can run the NTC00EXE program. To run the program, enter this command:

```
: run ntc00exe.ems.sysmgr
```

When the program starts, it checks first for the existence of EMG00JOB. If the EMG job does not exist, you receive the following error messages:

```
NTC-C- missing HP SysMgr EMG00EXE process (NTC003)  
NTC-C- program aborted (NTC004)
```

If the EMG00JOB is running, NTC00EXE builds a CON????? message file in your current working group with your current job ID. For example, CON00050 is created for job 50. Then NTC00EXE notifies the EMG process to transmit a copy of the console output when it happens. The EMG process passes a maximum of 24 lines of the latest console messages to NTC00EXE via this CON????? message file, so the NTC program can display them at the terminal. After this initial display, all the new console messages are appended to the existing display. Your terminal looks like a local console, but you cannot issue any commands.

If you want to use your terminal to issue a command, you can issue a **Ctrl-Y** to stop the NTC process. Be aware that the CON????? message file remains open throughout the entire logon session, and the EMG process continues to write each new console message that arrives. To resume viewing console messages, you can reactivate NTC00EXE by using the “run” command. If you stop this program for a long time, you may see three dotted lines indicating the console events that were lost due to CON????? file overflow. Since the EMG program delivers all the console messages, it also detects when a logon session terminates and purges the CON????? file for you.

This program also displays Automated Response Service (ARS) execution for you. Once an automated response starts, the EMG process sends all the output to your CON????? file. The VTC00??? program works exactly as the NTC00EXE program, so you can also monitor the automated response from a “Monitor Console” window at your OVC PC.

## **NETALLOW.EMS.SYSMGR**

The NETALLOW utility is a stand-alone MPE program used outside the OVC environment to allow users to reply to console messages during a Management Session (MSI), after MSI is started. The NETALLOW utility provides access to the additional capabilities required to issue **Ctrl-A** type commands. There are two versions of the program:

- NETALLOWX.EMSKIT.SYSMGR — the NM version
- NETALLOWV.EMSKIT.SYSMGR — the CM version

The OVC HP 3000 installation script installs the correct version and renames the file NETALLOW.

---

## VERCHECK.EMS.SYSMGR

The VERCHECK utility is a stand-alone MPE program used outside the OVC environment to verify the version number for your currently installed OVC HP 3000 software. You run this program from your logon in the SYSMGR account at the HP 3000. This example lists a sample output after you enter the command:

```
: run vercheck.ems

HP SysMgr NMEVENT module versions:

SL procedure: NMEVENT_VERS Version: A0101007

HP SysMgr NMEVENT overall version = A.01.01

HP SysMgr Managed Node HP36936 module versions:

NM program file: EMS000XL.EMS.SYSMGR Version: A0103013
CM program file: EMG00EXE.EMS.SYSMGR Version: A0103017
NM program file: VTC000XL.EMS.SYSMGR Version: A0103002
CM program file: NTC00EXE.EMS.SYSMGR Version: A0103000
NM program file: NETALLOW.EMS.SYSMGR Version: A0103R22
CM program file: ARS00EXE.EMS.SYSMGR Version: A0103003
NM program file: RSP000XL.EMS.SYSMGR Version: A0103007
CM program file: LAUNCHER.EMS.SYSMGR Version: A0103002
Catalog file: SMCAT000.EMS.SYSMGR Version: A0103000

HP SysMgr Managed Node HP36936 overall version = A.01.03

END OF PROGRAM

:
```

The sample output shows the version number, A.01.03, for your installed OVC HP 3000 software.

## REMOVEXE.COMMON.SYSMGR

The REMOVEXE utility is an MPE program used to delete a group of files using a file name with a wildcard. The INFO string is used to pass the file name specification. If the INFO string is not specified, REMOVEXE reads the DELLIST (in the COMMON group) to do the file deletion. The REMOVEXE program does not delete an open file or a PRIV file. If PARM=1 is specified, REMOVEXE lists every file that is removed. Figure 5-1 lists the current contents of the DELLIST.COMMON.SYSMGR.

**Figure 5-1**      **Example of DELLIST Contents**

```
!  
!  
! This file lists all the files which may be left from a product abort.  
! SysMgr's REMOVEXE utility reads this file and performs the disk cleanup.  
!  
! release version A.01.03  
!  
! Files left in the SYSMGR group  
!  
EMM00000.SYSMGR  
EMMTIMER.SYSMGR  
EMC00000.SYSMGR  
EMC00001.SYSMGR  
RCI#####.SYSMGR  
TIM#####.SYSMGR  
CAI#####.SYSMGR  
MGR00000.SYSMGR  
CTL#####.SYSMGR  
VCS#####.SYSMGR  
!  
! Files left in the COMMON group  
!  
SYSCMD00.COMMON  
UTL#####.COMMON  
OVS#####.COMMON  
UCS00000.COMMON  
RPT#####.COMMON  
RTP#####.COMMON  
RTS#####.COMMON  
MSGQTEMP.COMMON
```

REMOVEXE is a general purpose utility that anyone can use to perform a group delete with wildcard file names. Figure 5-2 shows an example that deletes all the trace files in the SYSMGR account.

**Figure 5-2 Example of Deleting Trace Files Using REMOVEXE**

```
:run removexe.common.sysmgr;info="tr@.@.sysmgr";parm=1

HP-SysMgr REMOVE utility A0103000 (c) HEWLETT-PACKARD 1992

REM-I- PURGE TRC0SYS0.COMMON.SYSMGR
REM-I- PURGE TRC0UCS0.COMMON.SYSMGR
REM-I- PURGE TRS00074.COMMON.SYSMGR
REM-I- PURGE TRS00079.COMMON.SYSMGR
REM-I- PURGE TRS00083.COMMON.SYSMGR
REM-I- PURGE TRS00100.COMMON.SYSMGR
REM-I- PURGE TRS00109.COMMON.SYSMGR
REM-I- PURGE TRS00112.COMMON.SYSMGR

REM-I- 8 FILES HAVE BEEN PURGED

END OF PROGRAM
:
```

## **DBEMLEXE.COMMON.SYSMGR**

The DBEMLEXE utility is an MPE program used to view the EML database. You run this program from a logon terminal at the HP 3000 or as a son process under the SMUTIL.PUB.SYSMGR utility. This program is designed for the OVC administrator who needs to examine the EML database on the HP 3000.

## USEREXE.COMMON.SYSMGR

The USEREXE utility is an MPE program used to update OVC user logon information. You run the program from a logon terminal at the HP 3000 or as a son process under the SMUTIL.PUB.SYSMGR utility. As the OVC administrator, you log on to the SYSMGR account in the COMMON group to run this program. This example lists the output after you run the program:

```
:run userexe

1 -- Show/Load User Record...
2 -- Add a new User Record...
3 -- Update a User Record...
4 -- Delete a User Record...
5 -- List User Records
6 -- Display User Dataset Capacity

0 -- Return to Previous Menu

Selection: 0

END OF PROGRAM
:
```

---

## CONVEXE.COMMON.SYSMGR

The CONVEXE utility is used by the management node's setup job file to add default entries to the CONFIG database. You can also use this program to merge two CONFIG databases. Table 5-1 lists the PARM values that control the operation of this program.

**Table 5-2** CONVEXE PARM Values

PARM value	Meaning
0	Load the CONFIG database contents into four binary files. The four binary files are MNCNFTMP, RNCNFTMP, USCNFTMP and MSCNFTMP.
1	Load the four files into the CONFIG database. Program fails if the identical entry already exists in the CONFIG database.
2	Load the four files into the CONFIG database. Duplicated entry is ignored if the identical entry already exists.
3	Load the four files into the CONFIG database. Duplicate entries are updated with the entry in the ??????TMP file.

The next example shows how to copy the CONFIG database from one HP 3000 to another HP 3000. First you run the CONVEXE program to save the CONFIG database at the old HP 3000, using this command syntax:

```
:run convexe  
  
CNV-I- HP SysMgr CONFIG convert program (A0103003)  
  
CNV-I- Save CONFIG USER dataset...  
CNV-I- Save CONFIG MSI dataset...  
CNV-I- Save CONFIG RN dataset...  
CNV-I- Save CONFIG MN dataset...  
  
CNV-I- Save done successfully  
  
END OF PROGRAM
```

**Then build the new CONFIG database at the new HP 3000, using this command syntax:**

```
: chgroup common
: file dbstext=configsc
: run dbschema.pub.sys;parm=3
```

**After the root file is created in COMMON.SYSMGR, run the DBUTIL program to create the CONFIG database, using this command syntax:**

```
:run dbutil.pub.sys

HP30391C.05.01 TurboIMAGE/XL: DBUTIL (C) COPYRIGHT
HEWLETT-PACKARD ...
>>create config
Database CONFIG has been CREATED.
>>exit

END OF PROGRAM

:
```

**Copy the MNCNFTMP, RNCNFTMP, USCNFTMP, and MSCNFTMP files to the new HP 3000. Then run the CONVE.XE program, using this command syntax:**

```
:run convexe;parm=1

CNV-I- HP SysMgr CONFIG convert program (A0103003)

CNV-I- Load CONFIG USER dataset...
CNV-I- USCNF record [nocap] added.
CNV-I- USCNF record [mgr] added.
CNV-I- file USCNFTMP deleted.

CNV-I- Load CONFIG MSI dataset...
CNV-I- file MSCNFTMP deleted.

CNV-I- Load CONFIG RN dataset...
CNV-I- RNCNF record [TITAN.ITG.HP] added.
CNV-I- RNCNF record [SYSMGR.ITG.HP] added.
CNV-I- RNCNF record [SAMPLE.DOMAIN.ORGANIZATION] added.
CNV-I- RNCNF record [HOZONE.ITG.HP] added.
CNV-I- file RNCNFTMP deleted.

CNV-I- Load CONFIG MN dataset...
CNV-I- MNCNF record added.
CNV-I- file MNCNFTMP deleted.

CNV-I- Load done successfully

END OF PROGRAM

:
```

## SMUTIL.PUB.SYSMGR

The SMUTIL utility program is used to perform various control and troubleshooting functions on the HP 3000. You can run the program from any terminal connected to the HP 3000. If run without the `INFO=` option, a series of menus leads you through an operation. You can perform a number of functions from the MPE command line by entering the appropriate INFO string. Valid INFO string values include:

### Syntax

```
: RUN SMUTIL.PUB.SYSMGR [ ;INFO= "TRACE ON" ]
                        [ ;INFO= "TRACE OFF" ]
                        [ ;INFO= "CONFIG" ]
                        [ ;INFO= "STATUS" ]
                        [ ;INFO= "SWITCH" ]
                        [ ;INFO= "STOP" ]
                        [ ;INFO= "FIX" ]
```

describes the valid INFO syntax options.

**Table 5-3**

**INFO Syntax Options**

Options	Description
TRACE ON	Turns on tracing for all processes and users.  CAUTION: This command enables tracing for every Event Message Server (EMS) on every managed node. If TRACE OFF is later used to turn off tracing for all processes and users at once, the trace file for every managed node is copied across the network to the HP 3000 simultaneously. This may result in a surge of network traffic.
TRACE OFF	Turns off tracing for all processes and users.
CONFIG	Displays the entire contents of the HP 3000 configuration database (CONFIG.COMMON.SYSMGR).
STATUS	Displays all OVC processes and users currently active on the HP 3000.
SWITCH	Initiates a SWITCH EML operation. This operation closes the current EML database, opens a new/empty EML database, and begins writing all subsequent event messages to the new database.
STOP	Stops both the JCOMMON and the JSYSMGR jobs on the HP 3000.
FIX	Fixes all the broken VRC and EMS links.

## Required Files

For SMUTIL to work properly, the following files are run as son processes of SMUTIL and must exist on the HP 3000:

- DBEXE.SYSMGR.SYSMGR
- DBEMLEXE.COMMON.SYSMGR
- USEREXE.COMMON.SYSMGR

## Main Menu

Figure 5-3 shows the main menu that is displayed when you run SMUTIL. The ellipses (. . .) following a command indicate that no direct action is taken when you select the command; a submenu is displayed instead.

Figure 5-3

**SMUTIL Main Menu (Part 1)**

```
:run smutil.pub.sysmgr;parm=100

*****
*****
*****      HP OpenView System Manager Utility      *****
*****
***** SMUTIL (A0103008) (c) Hewlett-Packard Co. 1992 *****
*****

-----WARNING-----
The "Modify Configuration Database" portion of this utility
is for use by trained HP support personnel only. Hewlett-
Packard will not be responsible for problems arising from
customer modifications made to the CONFIG##.SYSMGR database
with this utility.
-----WARNING-----

1 -- Tracing...
2 -- Display EML Records...
3 -- Display Configuration Database...
4 -- Display Process Status...
5 -- Switch to New EML Database
6 -- Fix Console/Network Paths...
7 -- Stop SysMgr...
8 -- Manage Users...
9 -- Modify Configuration Database...

0 -- Exit Utility

Selection: 9

-----
Do not update configuration databases while the SYSMGR
programs are running. Stop the programs, update the databases,
then restart the programs.
-----

Modify Configuration of:

1 -- Management Node
2 -- Managed Nodes...
3 -- Users...
4 -- EML...

0 -- Return to Previous Menu

Selection: 0
```

Entering the "PARM=100" is a special key for the SMUTIL program. With this parameter entered, the program displays the ninth menu, "Modify Configuration Database..." (see Figure 5-4). You can now modify the CONFIG database from the HP 3000. Be aware that some of the changes made with this method are *static*, meaning the MGREXE program is not aware of the changes until the next SysMgr program reload.

**Figure 5-4 SMUTIL Modify Configuration Database Menu (Part 2)**

```
Selection: 0

1 -- Tracing...
2 -- Display EML Records...
3 -- Display Configuration Database...
4 -- Display Process Status...
5 -- Switch to New EML Database
6 -- Fix Console/Network Paths...
7 -- Stop SysMgr...
8 -- Manage Users...
9 -- Modify Configuration Database...

0 -- Exit Utility

Selection: 0

END OF PROGRAM
:
```

---

## VERCHECK.PUB.SYSMGR

The VERCHECK utility is a stand-alone MPE program used to verify the HP 3000 software version during installation and other critical times during ordinary operation. To run this program, log on to the SYSMGR account at the HP 3000, using this command syntax:

```
: run vercheck.pub
```

HP SysMgr Management Node HP36936 module versions:

```
NM program file: SYSEXE.COMMON.SYSMGR      Version: A0103011
NM program file: UCSEXE.COMMON.SYSMGR      Version: A0103003
NM program file: OVSEXE.COMMON.SYSMGR      Version: A0103004
NM program file: RPTEXE.COMMON.SYSMGR      Version: A0103006
NM program file: REMOVEXE.COMMON.SYSMGR    Version: A0103000
NM program file: DBEMLEXE.COMMON.SYSMGR    Version: A0103001
NM program file: USEREXE.COMMON.SYSMGR    Version: A0103006
NM program file: CONVEXE.COMMON.SYSMGR     Version: A0103003
NM program file: MSI00EXE.COMMON.SYSMGR    Version: A0103005
NM program file: PSWDUTIL.COMMON.SYSMGR    Version: A0103000
NM program file: SMUTIL.PUB.SYSMGR         Version: A0103008
NM program file: GETMNEXE.PUB.SYSMGR       Version: A0103000
NM program file: VERCHECK.PUB.SYSMGR       Version: A0103003
NM program file: MGREXE.SYSMGR.SYSMGR      Version: A0103003
NM program file: RCI00X25.SYSMGR.SYSMGR    Version: A0103009
NM program file: RCI00LAN.SYSMGR.SYSMGR    Version: A0103009
NM program file: CAIEXE.SYSMGR.SYSMGR      Version: A0103000
NM program file: EMMEXE.SYSMGR.SYSMGR      Version: A0103003
NM program file: EMCEXE.SYSMGR.SYSMGR      Version: A0103006
NM program file: DBEXE.SYSMGR.SYSMGR       Version: A0103006
NM program file: VRCUTIL.SYSMGR.SYSMGR     Version: A0103006
NM program file: LSDUTIL.SYSMGR.SYSMGR     Version: A0103004
NM program file: SCXUTIL.SYSMGR.SYSMGR     Version: A0103002
CM program file: EMG00EXE.EMSKIT.SYSMGR    Version: A0103017
CM program file: ARS00EXE.EMSKIT.SYSMGR    Version: A0103003
CM program file: NTC00EXE.EMSKIT.SYSMGR    Version: A0103000
CM program file: VERCHECK.EMSKIT.SYSMGR    Version: A0103002
NM program file: EMS00XL.EMSKIT.SYSMGR     Version: A0103013
CM program file: EMS00MPE.EMSKIT.SYSMGR    Version: A0103013
NM program file: VTC00XL.EMSKIT.SYSMGR     Version: A0103002
CM program file: VTC00MPE.EMSKIT.SYSMGR    Version: A0103002
NM program file: NETALOWX.EMSKIT.SYSMGR    Version: A0103R22
CM program file: NETALOVV.EMSKIT.SYSMGR    Version: A0103000
NM program file: RSP00XL.EMSKIT.SYSMGR     Version: A0103007
CM program file: RSP00MPE.EMSKIT.SYSMGR    Version: A0103005
Download:      VERSION.COMMON.SYSMGR       Version: A0103001
Download:      DELLIST.COMMON.SYSMGR       Version: A0103000
Download:      RPT00ERR.COMMON.SYSMGR      Version: A0103000
Download:      FORM00MN.COMMON.SYSMGR      Version: A0103000
Download:      FORM00PC.COMMON.SYSMGR      Version: A0103000
Download:      EML00TPL.COMMON.SYSMGR      Version: A0103000
Download:      EMLSLTPL.COMMON.SYSMGR      Version: A0103000
Download:      MNCNFTPL.COMMON.SYSMGR      Version: A0103002
```

```
Download: RNCNFTPL.COMMON.SYSMGR Version: A0103001
Download: USCNFTPL.COMMON.SYSMGR Version: A0103000
Download: EML00XSC.COMMON.SYSMGR Version: A0103001
Download: CONFIGSC.COMMON.SYSMGR Version: A0103000
Download: OVCAT000.COMMON.SYSMGR Version: A0103000
Download: MSIOHELP.COMMON.SYSMGR Version: A0103000
Download: MSIIHELP.COMMON.SYSMGR Version: A0103000
Download: SMUDC.PUB.SYSMGR Version: A0103003
Download: OVUDC.PUB.SYSMGR Version: A0103001
Download: TESTEVNT.PUB.SYSMGR Version: A0103003
Download: HALTMSGF.SYSMGR.SYSMGR Version: A0103001
Download: EMG00TPL.EMSKIT.SYSMGR Version: A0103007
Download: SMCAT000.EMSKIT.SYSMGR Version: A0103000
Download: JRNINST.EMSKIT.SYSMGR Version: A0103005
Download: EMG0AUTO.EMSKIT.SYSMGR Version: A0103000
Download: MPE00TPL.EMSKIT.SYSMGR Version: A0103006
Download: EMGINIT.EMSKIT.SYSMGR Version: A0103000
Download: NOTICE.EMSKIT.SYSMGR Version: A0103001
Download: EXECUTE.EMSKIT.SYSMGR Version: A0103000
```

HP SysMgr Management Node HP36936 overall version = A.01.03

END OF PROGRAM

:

---

**NOTE**

The previous example gives you an idea of the output from the VERCHECK program. The version numbers in the example may not reflect the latest OVC HP 3000 software version.

---

OVC Utilities  
VERCHECK.PUB.SYSMGR

---

# 6

# Troubleshooting Problems

This chapter describes how you troubleshoot some of the most common problems when using OVC.

## Overview

You may need help from many different experts to troubleshoot an OVC problem. For example, if a WINSOCK error is displayed on your OVC PC screen, you may need the assistance of your local network administrator. If you have problems streaming jobs at the HP 3000, you may need the assistance of your local system manager.

This chapter does not help you troubleshoot OpenView platform problems, NS network problems, or PC configuration issues. Specific manuals and HP support experts are available to help you with those subjects.

What this chapter does provide are the necessary procedures to troubleshoot OVC problems. First you are given instructions on how to check software versions. Then you are given information on how to identify problems, and finally answers to the most frequently asked OVC questions. (It is not surprising that many questions relate to installing and setting up OVC.)

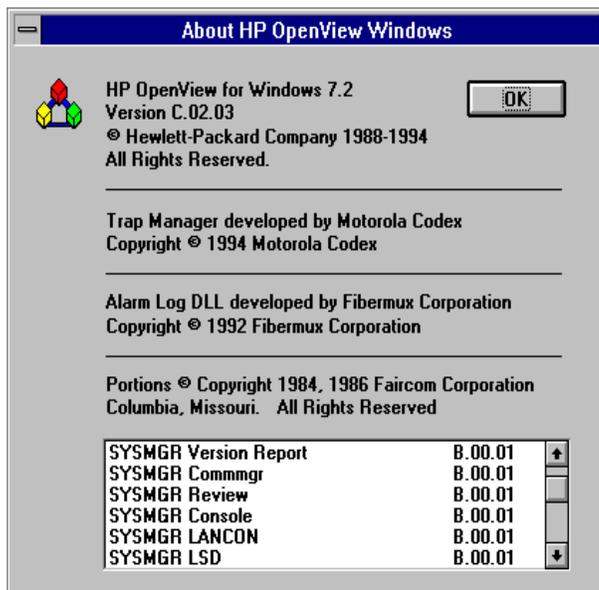
After trying many paths to solve your problem, if you decide that a program defect is still occurring, you should contact an HP support engineer.

## Version Checking

OVC does not operate correctly if software versions are mixed on the HP 3000 or the OVC PC. Utilities are provided which report version information for installed OVC software. You can use these utilities to do a version check:

1. For HP 3000 software version information, type the following command line at the HP 3000:  
  
:RUN VERCHECK.PUB.SYSMGR
2. For OVC PC software version information, click on the About . . . option in the Help menu of the OpenView window. The About HP OpenView Windows dialog box is displayed (see Figure 6-1).

**Figure 6-1** Dialog Box Showing OVC PC Program Version Report



In the list box are all the OVC PC programs with the version information. (The version numbers of your OVC PC programs may differ from those listed in this figure.)

## Troubleshooting Techniques

Since the OVC product includes software that resides on both the HP 3000 and the OVC PC, this section is divided into two parts, one for each group of software.

### Troubleshooting the HP 3000 Module

As described in Chapter 3, “OVC Building Blocks,” the JCOMMON job runs on the HP 3000 and provides all the required server functions for the OVC PC. If the job fails to execute, find the job spool file and read it. The cause of failure should be very visible at the end of the spool file.

### Troubleshooting the OVC PC Module

The C:\OV\OVWIN.EXE program gets launched first when you double-click on the HP OpenView program icon. The C:\OV\OVWIN.LOG file is the log file used by the C:\OV\OVWIN.EXE program. Check this log file if the OpenView platform starts, but some of the OpenView applications fail. If you have a problem running the C:\OV\OVWIN.EXE program, you must isolate the problem first by commenting out these entries in the C:\OV\OVWIN.INI file to disable all the OVC PC modules:

```
[OpenViewApps]
;XOVSRVR=C:\OV\XOVSRVR.EXE
;SYSMGR=C:\OV\SYSMGR\SYSMGR.EXE
;SMMSI=C:\OV\SYSMGR\MSI.EXE
;SMUSRCFG=C:\OV\SYSMGR\USRCFG.EXE
```

If the problem still exists after you do this, and if you still have problems starting the OpenView program (C:\OV\OVWIN.EXE), notify an HP representative indicating that the problem definitely belongs to the OpenView platform. The OVC product and the OpenView platform are two totally different HP software products. OVC is an OpenView application that runs on top of the OpenView PC platform just as OpenView is an application that runs on top of Microsoft Windows.

The OVC PC software includes many programs. Most of these programs rely upon the C:\OV\SYSMGR\COMMMGR.EXE program to provide network communication with the HP 3000 server. This design isolates network operation and improves product reliability. The COMMMGR program has extended trace logic for troubleshooting network problems. Once the trace is enabled, it displays the trace messages using the DBWIN.EXE window. The DBWIN.EXE program is a utility program that is included with the Microsoft Software Development Kit (SDK). If you cannot obtain copies of the DBWIN.EXE and DBWIN.DLL programs, connect a terminal with the COM1 (or AUX)

port to display the same trace messages. Use the COMMMGR program's on-line help to get information on how to use this tracing feature.

If an error message dialog box is displayed, first determine if it is from an OVC PC module. All OVC error messages on the PC have a designated error code attached at the end of the message. This error code has the P?-?? format (for example, P1-2). This error code serves as the index for you to find the error explanation in Appendix B, "OVC PC Error Messages" of the *HP OpenView Console User's Guide*. The P? part of the error code indicates which program generated the error. Table 6-1 lists the error code prefixes and the corresponding program that generated the error.

**Table 6-1 OVC PC Error Codes**

<b>Error Code</b>	<b>Related Programs</b>
P1	C:\OV\SYSMGR\COMMMGR.EXE
P2	C:\OV\SYSMGR\SYSMGR.EXE
P3	C:\OV\SYSMGR\CONSOLE.EXE
P4	C:\OV\SYSMGR\REVIEW.EXE
P6	C:\OV\SYSMGR\USRCFG.EXE
P7	C:\OV\SYSMGR\MNCFG.EXE\
P8	C:\OV\SYSMGR\EMLCFG.EXE
PA	C:\OV\SYSMGR\MSI.EXE
PB	C:\OV\SYSMGR\REPORT.EXE
PC	C:\OV\SYSMGR\LANCON.EXE
P0	Error message common to all programs.

## Frequently Asked Questions

This section contains the most frequently asked questions that you may have when setting up or using OVC.

### **Question: Why do some symbols on the map remain blue?**

Check to see if the symbol's object name is valid, that is, there is a corresponding entry in the MPE00DEF file on the HP 3000. If you are also using a USER0DEF file, then you should look at the DEF0LIST file.

---

**NOTE**

---

It is very important that the object names for the map symbols be in uppercase.

### **Question: How do I remove all events logged against OVC symbols without using the Event Browser to delete every object individually?**

1. Stop the HP 3000 job using the OVSTOP UDC.
2. Delete the two event queue files, MSGQHEAD.COMMON.SYSMGR and MSGQPTER.COMMON.SYSMGR.
3. Restart the HP 3000 job using the OVSTART UDC.
4. Issue this command on the HP 3000:

```
: TELLOP NMEV#INIT
```

Doing this re-initializes the OVC event collecting agent.

If this `TELLOP` command is not issued, the symbols on the map remain blue until events get posted to them.

These actions clear all map symbols on the OVC PC that is connected to the HP 3000. All folder object events are also destroyed as a result of this action. The drop-down folder list box in the `Event Browser` is also cleared.

### **Question: Why are bitmaps in OVC windows misaligned?**

This is sometimes a common problem with the `Event Browser` where one line event messages do not align with the rectangular buttons on the left.

This occurs when you select a video mode which uses large fonts.

Reconfigure your OVC PC to use small fonts, and the problem should not reoccur.



---

# A

## Using the NMEVENT API

This appendix describes the NMEVENT API and how to use it.

## The NMEVENT API

---

**NOTE**

In this chapter, whenever the word *management node* is cited, it refers to the HP 3000.

The NMEVENT Application Programming Interface (API) allows application programs running on managed nodes to send events to SysMgr running on the management node. Events generated by applications calling NMEVENT are treated by SysMgr as events generated by the console. Both types of events are processed in the same way on the managed and management nodes.

Because NMEVENT depends on the managed node software and the NS transport, both must be running on the managed node in order to have events sent to the management node. NS must also be running on the management node.

---

## Using the NMEVENT API

You can use the NMEVENT API in the following way:

### Syntax

```
NMEVENT (application_name, application_name_size,
         I32 CA I32
         application_code, msg_text, msg_size,
         I32 I32 I32
         event_state, status, supersede);
```

---

## Parameters

Any program running on a managed node can call NMEVENT. Each event must be associated with the following parameters, and all parameters are required. Table A-1 lists and describes the parameters.

**Table A-1** NMEVENT Parameters

Parameter	Description
application_name	A user-defined character array, 0 to 16 characters long, used to identify the application calling the intrinsic. It is descriptive only. Identifiers longer than 16 characters are truncated on the right. Any non-printing ASCII characters are replaced by a space. This parameter is passed by reference.
application_name_size	This 32-bit integer should be set to the length of the application name. Valid range is 0 - 16. This parameter is passed by value.
application_code	<p>A user-defined 32-bit integer that uniquely identifies the application calling the intrinsic and the type of event. The value of the user-defined application code for NMEVENT messages must range from 0 to 31999.</p> <p>It is the combination of the value of the Application ID and the class number that makes the application_code. For example, an Application ID with a value of 233 plus a class number of zero together make an application_code value of 23300.</p> <p>Refer to Chapter 4 , “Customizing Your Map and System Events,” for more information.</p>
msg_text	A user-defined character array that holds the message text. The maximum size of this array is 278 characters. SysMgr uses the character (byte) defined as (msg_size + 1) for SysMgr internal program use. This leaves a maximum of 277 bytes for message text. If the length of the message text exceeds 277, the message text is truncated on the right. Any nonprinting ASCII characters are replaced by a space. This parameter is passed by reference.
msg_size	This 32-bit integer should be set to the length of the message text. Valid range is 0 to 277. This parameter is passed by value.

**Table A-1** NMEVENT Parameters

Parameter	Description
event_state	<p>This 32-bit integer is used to control the display of the message in Show Events.</p> <p>The event state should be set from 0 to 3:            If set to 0, the message is not displayed.            If set to 1, the message is displayed as an informational event (magenta).            If set to 2, the message is displayed as a warning event (yellow).</p> <p>If set to 3, the message is displayed as a critical event (red).</p> <p>All messages are logged to the EML database regardless of what the event state is set to. This parameter is passed by value.</p>
status	<p>This 32-bit integer value is returned to the application. It reports the outcome of the call to NMEVENT. These values are listed later in this appendix. This parameter is passed by reference</p>
supersede_flag	<p>This 32-bit integer determines how the event state of this event affects the status on the map. The difference between node status and application status is that <i>node status</i> is the sum of all application statuses and status reported by SysMgr. The <i>application status</i> is the severity level of each application on a given node.</p> <p>The node status of a given node (the color on the map) is set to the highest severity level for that node:</p> <p>If the supersede flag is set to -1, the event is stored in the EML but does not affect either the application status or the node status.</p> <p>If the supersede flag is set to 0, the application status is set to whichever is the higher severity, the current application status or the event state of the event.</p> <p>If the supersede flag is set to 1, the application status is set to the event state of the event. This allows an application to reset a higher level severity to “normal,” as the case that all the pending events have been acknowledged (removed).</p> <p>Whenever any application status changes, the node status is updated, if appropriate. This parameter is passed by value.</p>

**NOTE** Use this supersede\_flag carefully to avoid confusing the map status.

## Return Values

An integer value, reporting the outcome of the call to NMEVENT, is returned to the application. Table A-2 lists and describes the return values.

**Table A-2 NMEVENT Values**

<b>Value</b>	<b>Description</b>
0	OK — message was successfully given to the managed node software.
1	Warning — msg size parameter is negative. NMEVENT replaced the bad parameter with a default value of 0. The message was successfully given to the managed node software.
2	Warning — msg size parameter is too large. NMEVENT replaced the bad parameter with a default value of 278. The message was successfully given to the managed node software.
3	Warning — event state parameter is negative. NMEVENT replaced the bad parameter with a default value of 0. The message was successfully given to the managed node software.
4	Warning — event state parameter is greater than 3. NMEVENT replaced the bad parameter with a default value of 0. The message was successfully given to the managed node software.
5	Warning — application name size parameter is negative. NMEVENT replaced the bad parameter with a default value of 0. The message was successfully given to the managed node software.
6	Warning — application name size parameter is too large. NMEVENT replaced the bad parameter with a default value of 16. The message was successfully given to the managed node software.
7	Warning — supersede flag has invalid parameters. NMEVENT replaced the bad parameter with a default value of -1. The message was successfully given to the managed node software.
8	Warning — application code parameter is greater than 31999. This is the HP reserved range of application codes. The message was successfully given to the managed node software.
-1	Error — application code parameter is negative. The message was not given to the managed node software.
-4	SysMgr has not been installed or activated. The message was not given to the managed node software.

## Programming Examples

To allow the same version of your application to link and execute properly in both MPE/iX and MPE/V systems, the next sections provide two examples of how you can create your own procedure, NMEVSPL. NMEVSPL checks to see if NMEVENT exists before trying to call it. If you are programming in Modcal (Native Mode Pascal) on an MPE/iX system, use the Modcal procedure example. If you are programming on an MPE/V system or in Compatibility Mode on an MPE/iX system, use the SPL procedure example.

The Modcal (Native Mode Pascal) procedure `NM_EVENT` uses the intrinsic `hpgetprocplabel` to determine whether NMEVENT is in the system. To use this procedure, copy the PROCEDURE, CONST, TYPE, and VAR declarations into your code. You may call it from the main program or another procedure. If NMEVENT is installed, `NM_EVENT` calls it. If not, `NM_EVENT` does not call it.

The SPL procedure `NMEVSPL` uses assembly language and the intrinsics `loadproc` and `unloadproc` to check whether NMEVENT is in the system SL. To use this programming example, copy this procedure into your code and compile it as an SPL procedure. You may call it from the main program or another procedure. If NMEVENT is installed, `NMEVSPL` calls it. If not, `NMEVSPL` does not call it and returns a status of -4.

### Programming Example with Pascal (Modcal)

This is a standard Pascal (Modcal) example:

```
$STANDARD_LEVEL `HP_MODCAL`$
PROGRAM EXAMPLE (input, output);

CONST
    APLNAMLEN = 16;
    MAXMSGLEN = 278;

TYPE
    status_type = RECORD
        CASE BOOLEAN OF
            TRUE: (int      : integer);
            FALSE (status  : shortint;
                  subsys   : shortint);
        END;
    appl_name_type = PACKED ARRAY [1..APLNAMLEN] OF char;
    msg_text_type  = PACKED ARRAY [1..MAXMSGLEN] OF char;
    procedure_type = PROCEDURE (
        appl_name      : appl_name_type;
        appl_name_size : INTEGER;
        appl_code      : INTEGER;
        msg_text       : msg_text_type;
        msg_size       : INTEGER;
```

## Using the NMEVENT API Programming Examples

```

                                event_state      : INTEGER;
VAR status                      : INTEGER;
                                supersede_flag   : INTEGER);
VAR
  appl_name      : appl_name_type;
  appl_name_size : INTEGER;
  appl_code      : INTEGER;
  msg_text       : msg_text_type;
  msg_size       : INTEGER;
  event_state    : INTEGER;
  status         : INTEGER;
  supersede_flag : INTEGER;
  pstatus        : status_type;
  procname       : PACKED ARRAY[1..20] OF CHAR;
  pbool          : BOOLEAN;
  plabel         : procedure_type;

PROCEDURE hpgetprocplabel; INTRINSIC;

{define procedure nm_event call}

PROCEDURE nm_event(
                                appl_name      : appl_name_type;
                                appl_name_size : INTEGER;
                                appl_code      : INTEGER;
                                msg_text       : msg_text_type;
                                msg_size       : INTEGER;
                                event_state    : INTEGER;
                                VAR status     : INTEGER;
                                supersede_flag : INTEGER);

BEGIN

  procname      := 'nmevent';
  pbool         := FALSE;
  hpgetprocplabel(procname, plabel, pstatus.int,, pbool);
  IF pstatus.int = 0 THEN

    {if nmevent intrinsic exists, call instrinsic}

    call (plabel, appl_name, appl_name_size,
          appl_code, msg_text, msg_size, event_state,
          status, supersede_flag);

  ELSE
    BEGIN

      {nmevent intrinsic does not exist, do nothing}

      writeln(pstatus.status);
    END;

  END; {end of nm_event procedure}

BEGIN {main program}
```

```
{somewhere in your program you can call nm_event procedure}

appl_name           := 'BIG COMPANY';
appl_name_size     := 11;
appl_code          := 1000;
msg_text           := 'Normal Product Shutdown';
msg_size           := 23;
event_state        := 1;
supersede_flag     := -1;

nm_event(
    appl_name, appl_name_size, appl_code, msg_text,
    msg_size, event_state, status, supersede_flag);

{rest of program}

END.
```

## Programming Example with SPL

NMEVSPL does a LOADPROC on the NMEVENT API. If the NMEVENT API is loaded in the system SL, then NMEVSPL loads the proper parameters onto the stack and calls NMEVENT directly with a PCAL 0. After the NMEVENT call completes, NMEVSPL returns the status to the calling program.

If the NMEVENT API is not in the system SL, then the LOADPROC is unsuccessful. At this point, NMEVSPL sets the return status to -4, indicating that NMEVENT is not in the system SL. NMEVSPL returns to the calling program without calling NMEVENT.

This is a standard SPL example:

```
procedure nmevspl (applname,
                  applnamesize,
                  applcode,
                  msgtext,
                  msgsize,
                  eventstate,
                  status,
                  supersede);
value applnamesize, applcode, msgsize, eventstate, supersede;
byte array applname;
double applnamesize, applcode;
byte array msgtext;
double msgsize, eventstate, status, supersede;

begin
    equate lib'system = 4;
    define nmevent1 = "nmevent" "#;
    byte array procname(0:7);
    integer plabel, pnum;
    intrinsic loadproc, unloadproc;

    move procname := (nmevent1);
    pnum := loadproc(procname, lib'system, plabel);
```

Using the NMEVENT API  
Programming Examples

```
    < fail to load intrinsic, do nothing >

    if << then begin
    status := -4D;
    return;
    end;

    tos:=@applname;
    tos:=applnamesize;
    tos:=applcode;
    tos:=msgtext;
    tos:=msgsize;
    tos:=eventstate;
    tos:=@status;
    tos:=supersede;
    tos:=plabel;
    assemble(PCAL 0);
    unloadproc(pnum);

end; < Procedure NMEVSPL>
```

---

**B****Console Setup Script**

This appendix describes the operating logic of the Console Setup script file provided with the OVC PC installation.

## Overview

When you double-click on the Console Setup program icon in the HP OpenView program group, you start the `consetup.rcl` Reflection script file. This file sets up the OVC environment for local use and resides in the `C:\OV\SYSMGR\INST` directory on your PC. This appendix describes the operating logic of this OVC setup script file.

### CONSETUP.RCL Script Logic

This is the procedure for the `consetup.rcl` script:

1. Extract the HP 3000 NS node name from the `[sysmgr]` section of the `c:\ov\sysmgr\sysmgr.ini` file and confirm it with the user.
2. Get the logon information from the user and log on to the HP 3000.
3. Update the `Mgmt Node=` entry in the `[sysmgr]` section of the `c:\ov\sysmgr\sysmgr.ini` file on the OVC PC to contain the user-supplied HP 3000 node name given in step 1.
4. Update the logon and password in the JCOMMON job file with the user-supplied logon information given in step 2. The job will run at an INPRI of 13.
5. Update the EMLSLTPL (EML Switch Log job template) file on the HP 3000 with the user-supplied logon information.
6. Build the CONFIG database on the HP 3000 if it does not exist. This database will contain the OVC-related configuration information.
7. Build the EML database on the HP 3000 if it does not exist. This database will contain the events from the HP 3000.
8. Build the EML message file on the HP 3000 if it does not exist. This file is used by OVC to exchange events between modules.
9. Configure the OVC HP 3000 system as a “large computer” in the CONFIG database.
10. Create the EMS group on the HP 3000 with specific capabilities using this command:  

```
NEWGROUP EMS;CAP=PH,PM,IA,BA;ACCESS=(R,X:ANY;W,A,L,S:AC)
```
11. Copy the required files from the EMSKIT group to the EMS group.
12. Create the EMG00JOB job file on the HP 3000 with the user-supplied logon information given in step 2.
13. Create the logical object definition file, MPE00DEF, on the HP 3000 from the template file, MPE00TPL.

14. Create the SECURE group on the HP 3000 using this command:

```
NEWGROUP SECURE ; CAP=PH , PM , IA , BA ; ACCESS=  
( R , X , W , A , L , S : AC )
```

15. Copy the files related to Automated Response Service (ARS) from the EMSKIT group to the SECURE group.

16. With the user's permission, start OVC on the HP 3000 system by streaming the JCOMMON.COMMON.SYSMGR job file.



---

# C

## **Customizing the OVC PC Environment**

This appendix describes the initialization files (.INI files) that you can modify to change the OVC PC environment for local use.

## Overview

PC software applications typically use initialization files (usually \*.INI files) to obtain specific information about start-up default values. These files contain entry value pairs that are read by Windows, OpenView, or OVC at start-up. Three \*.INI files on your OVC PC affect your OVC PC's working environment:

### **C:\OV\OVWIN.INI**

The “\*.INI” file for OVWIN.EXE.

### **C:\OV\XOVSRRV.INI**

The “\*.INI” file for XOVSRRV.EXE.

### **C:\OV\SYSMGR\SYSMGR.INI**

The “\*.INI” file for SysMgr programs.

This chapter provides you with the necessary information so that as OVC administrator you can configure these files when necessary.

### **C:\OV\OVWIN.INI**

The C:\OV\OVWIN.INI file is used by the OVWIN.EXE to load map and OpenView applications. Only five entries in this file require your attention. The first entry is the [OpenView] section:

```
[OpenView]
map=C:\OV\SYSMGR\MAP\<your default map name>
```

In this section you specify the default map you want for each OpenView session. You can use this syntax to edit the line according to your map's file name.

Or you can also use the pop-up dialog box displayed when you click on the **Customize HP OpenView** option from the **Options** menu. When you click on the **OK** button to exit the **Customize HP OpenView** dialog box, the OVWIN.EXE program updates this entry automatically.

The second entry is the [OpenViewApps] section:

```
[OpenViewApps]
XOVSRRV=C:\OV\XOVSRRV.EXE
SYSMGR=C:\OV\SYSMGR\SYSMGR.EXE
SMMSI=C:\OV\SYSMGR\MSI.EXE
SMUSRCFG=C:\OV\SYSMGR\USRCFG.EXE
```

In this section you specify the OpenView applications that you want the OVWIN.EXE program to load. Since most OVC PC programs are Visual OpenView applications, the XOVSRRV.EXE program must also be loaded. All the Visual OpenView applications are registered via the XOVSRRV.INI file. The SYSMGR.EXE, MSI.EXE and USRCFG.EXE

programs are not Visual OpenView applications, and they must be specified here.

**SYSMGR.EXE** This program does not have any user interface. The sole purpose of this program is to report the version of each \*.EXE file in the C:\OV\SYSMGR directory. When you select the OpenView About... menu item, this program examines all the C:\OV\SYSMGR\\*.EXE files specified in the XOVSRRV.INI program under the [OpenView] section. It gets the version information for each \*.EXE file and reports them via the list box in the *About HP OpenView Windows* window.

**MSI.EXE** This program provides the MSI feature for OVC. It launches the Reflection window for an MSI session.

**USRCFG.EXE** This program allows an OVC administrator to configure OVC users from the PC. The actual user records are stored on the HP 3000 management node.

---

**NOTE**

Do not remove or alter these three entries.

## C:\OV\XOVSRRV.INI

The C:\OV\XOVSRRV.INI file is used by the XOVSRRV.EXE program to register Visual OpenView applications.

The first entry is the [Programs] section:

```
[Programs]
SMCOMM=C:\OV\SYSMGR\COMMMGR.EXE,1
SMREVIEW=C:\OV\SYSMGR\REVIEW.EXE
SMMNCFG=C:\OV\SYSMGR\MNCFG.EXE
SMEMLCFG=C:\OVS\SYSMGR\EMLCFG.EXE
SMREPORT=C:\OV\SYSMGR\REPORT.EXE
SMLANCON=C:\OV\SYSMGR\LANCON.EXE,2
SMMAP=C:\OV\SYSMGR\MAPCOPY.EXE
```

This section specifies all the SysMgr Visual OpenView applications. The entries also specify the path to each application and if each should be launched at the OpenView initialization stage. The “1” after the COMMMGR.EXE means to launch the COMMMGR.EXE during XOVSRRV.EXE initialization and display its window normally. The “2” after the LANCON.EXE application means to launch the program and display its window minimized.

This is the [MenuIDs] section:

```
[MenuIDs]
SYSMGRCFG=Customize HP Sys&Mgr, OV_IDMOPTION, Configure SysMgr. SYSMGRREPT=&Report, OV_IDMMAIN, Report Generation.
```

These two entries add two cascading SysMgr menu items to the menu

structure of OpenView. One menu item is added to the OpenView main menu bar, and another is added under the Options menu. The last field contains the text that is displayed for the status field of the main OpenView window.

This is the [CommandIDs] section:

```
[CommandIDs]
SYSMGR1=SysMgr &Logoff, OV_IDMCONTROL, 0x100, OVM_DISABLED, SMCOMM
SYSMGR8=Configure &EML Database..., SYSMGRCFG, 0x101, OVM_ENABLED
SYSMGR9=SysMgr Event &Browser..., OV_IDMMONITOR, 0x101, OVM_ENABLED
SYSMGR10=&Show SysMgr Events..., OV_IDMPOPUP, 0x105, OVM_ENABLED
SYSMGR11=SysMgr &Report..., SYSMGRREPT, 0x101, OVM_ENABLED, SMREPORT
SYSMGR12=SysMgr C&opy Map, OV_IDMEDIT, 0x106, OVM_ENABLED, SMMAP
SYSMGR13=SysMgr P&aste Map..., OV_IDMEDIT, 0x107, OVM_DISABLED, SMMAP
```

---

**NOTE**

Be aware that because of the page size, each entry in this listing of the [CommandIDs] section is truncated. Print out the actual C:\OV\XOVSRVR.INI file to read the full entry for each SysMgr menu item.

The syntax for the entry is:

```
<cmdEntry>=<CmdName>,[!]<Parent MenuID>,<CmdID>,<flags>,<TaskName>,<ObjectType>[,<Status string>]
```

For a more detailed explanation, refer to the *HP Visual OpenView User's Guide*

At the bottom of the file are two entries, [AlarmSets] and [AlarmTypes]:

```
[AlarmSets]
SYSMGR=SMCOMM

[AlarmTypes]
SYSMGR=SYSMGR, 100, , , "SysMgr Event Notice"
```

These two entries are used to post OVC events to the OpenView map. *Do not* remove or alter these two entries.

## C:\OV\SYSMGR\SYSMGR.INI

The C:\OV\SYSMGR\SYSMGR.INI file is used by the OVC application to store configuration information. The remaining pages of this appendix describe several subsections of this file including:

- [SysMgr]
- [OVSymbols]
- [SMTerm]
- [Console]

[SysMgr] Section

This section contains configuration parameters used by the OVC PC software modules. The configuration parameters include:

MgmtNode=<your default management node name>

This entry defines your current HP 3000 management node. If the COMMMGR.EXE program cannot find this entry, or no node name is specified in this entry during its start-up, COMMMGR.EXE is unable to create a NOCAP link with the HP 3000 management node. In this situation, all the symbols on the map remain blue (status unknown).

Since COMMMGR.EXE inserts or updates this entry automatically upon a successful connection, you do not have to manually edit this entry.

SymbolsDir=C:\OV\SYSMGR\SYMBOLS

This entry provides the pathnames for all the OVC map symbols (the \*.ICO icon files) for the OpenView program, OVWIN.EXE.

ConsoleFont=r\_roman8, 10

This entry specifies the current font and font size used by the CONSOLE.EXE program. If this entry is missing, the default font and font size are "Terminal, 6." Since the WRQ Reflection terminal emulator is required for OVC MSI operation, use the "r\_roman8" font for the best appearance. If you do not have this font installed, you may have to install it using the Windows control panel. The CONSOLE.EXE program is not used by OVC.

ConsoleMemory=48

This entry specifies the size of display memory (lines) for each console window of the CONSOLE.EXE program. The default size is 48 lines.

OVConsole=No/Yes

This entry specifies if your OVC PC is the OV console or not. The entry also controls how OVC MSI operates. If the OVConsole= value is Yes, then only one MSI0 terminal emulator window can be launched. Otherwise, two terminal emulator windows (MSI1 and MSI2) can be launched.

SMFolder=<folder object name>

This entry specifies the default folder object used by the REVIEW.EXE program. This entry is updated automatically and requires no user modification.

BrowserShowTime=No/Yes

This entry is used by the REVIEW.EXE program to specify the initial setting for a new event window. If this entry is missing, the default is Yes.

RCLowWaterMark=25

Both the CONSOLE.EXE and REVIEW.EXE programs use the Multiple Document Interface (MDI). To prevent you from opening too many son windows (and exhausting all system resources), this entry specifies the minimal system resource (the user heap space) needed for creating new son windows. CONSOLE.EXE and REVIEW.EXE will not launch new windows when this limit is reached. The default value is 25 (%).

HotPrinterPort=2

This entry specifies the “log bottom” printer port number, a default value of 2.

EventSound=No/Yes

EventText=No/Yes

EventRun=No/Yes

These three entries specify how an OVC event gets posted to the OpenView map. If the EventSound= entry value is Yes, each symbol color update generates an alert sound. If the EventText= entry value is Yes, each symbol update adds a “SysMgr Event Notice” message to the OpenView alarm log. If the EventRun= entry value is Yes, the posted event can trigger pre-configured applications (an OpenView feature). You can change these settings using the COMMMGR program’s OpenView Alarm Setup dialog box. You do not manually change the settings with an editor.

ConsolePosition=

ReviewPosition=

ReportPosition=

UsrCfgPosition=

EMLCFGPosition=

MNCFGPosition=

CommMgrTopPos=

CommMgrLeftPos=

These entries are used by the OVC PC program to save their current position and size. No user modification is required.

LinkFailPolling=No/Yes

This entry is used by the COMMMGR.EXE program to indicate if the Link Failure Polling feature is enabled or not. The COMMMGR program updates this entry automatically.

LinkFailPollingInterval=20

This entry specifies the time interval of retries in seconds for link polling. If this entry is missing, the default value of 10 seconds is used. If the specified value is smaller than 5, then the value of 5 seconds will be used. If the specified value is greater than 60, then the value of 60 seconds is used.

[OVSymbols] Section

This section defines all the OVC map symbols.

```

4307=3 0 SOLCOMPR      "sysmgr (L) system+"
4308=3 0 SOMCOMPR     "sysmgr (M) system+"
4309=3 0 SOSCOMPR     "sysmgr (S) system+"

23CF=2 -1 S0FOLDER    "Folder (SysMgr)"
23CE=2 -1 SOMPE       "MPE/iX (SysMgr)"
23CD=2 -1 SOSYSMGR    "SysMgr (SysMgr)"
23CC=2 -1 SOENV       "Environment (SysMgr)"
4300=2 -1 SOSAFE      "Security (SysMgr)"
4301=2 -1 SOMETER     "Performance (SysMgr)"
4302=2 -1 SOCHIPS     "Hardware (SysMgr)"
4303=2 -1 SOCONS      "Console (SysMgr)"

4350=2 -1 S0JOBS      "Jobs (SysMgr)"
4351=2 -1 S0SPOOL     "Spooler (SysMgr)"
4352=2 -1 SOPNTR2     "Printer_2 (SysMgr)"
4353=2 -1 S0TAPE2     "Tape_2 (SysMgr)"

23CB=2 -1 SOPNTR1     "Printer_1 (SysMgr)"
23CA=2 -1 S0TAPE1     "Tape_1 (SysMgr)"
23C9=2 -1 SOMAIL      "Mail (SysMgr)"
23C8=2 -1 S0DISK      "Disk (SysMgr)"
23C7=2 -1 S0NETWRK    "Network (SysMgr)"
23C6=2 -1 S0DB        "Database (SysMgr)"
23C5=2 -1 SOEVENT     "Event (SysMgr)"
  
```

[SMTerm] Section

This section defines the terminal emulator used by the OVC PC software.

```

TerminalEmulator=C:\RWIN\R1WIN.EXE
UseAdvLink=No
TermEmulConfigExt=r1w
TermEmulScriptExt=rcl
  
```

You can modify the first entry if your R1WIN.EXE was moved to another directory. This is the default directory.

[Console] Section

This section stores the configuration data for the LAN Console program, C:\OV\SYSMGR\LANCON.EXE. These are the current default values:

```

COM1=NAILED, CONSOLE
COM2=DISABLED
COM3=DISABLED
COM4=DISABLED
MaxPort=4
  
```

The MaxPort= entry is used by the LANCON.EXE program to define how many COM ports are configured for OVC use. Each COM port requires a "COM?=" entry. The LANCON.EXE program uses this entry

to save the current COM port setup:

LaunchNailedConsole=No/Yes

**If the LaunchNailedConsole= entry value is Yes, the LANCON.EXE program launches a terminal session for each NAILED COM port at startup.**

TerminateWithOV=No/Yes

**If the TerminateWithOV= entry value is Yes, all the terminal sessions launched by the LANCON.EXE program are terminated when OpenView exits. Otherwise, the local console terminal emulator remains active.**

---

**D**

---

**Upgrading from OVC to HP  
OpenView SysMgr**

This appendix gives you step-by-step instructions on how to upgrade your OVC software to HP OpenView System Manager software.

## Overview

As your system requirements change, you may want to upgrade your OVC software to SysMgr software. With OVC you can manage only one HP 3000. But with SysMgr you can monitor and control multiple HP 3000s. This appendix explains how to do the upgrade with step-by-step instructions.

---

**NOTE**

When you decide to upgrade your OVC software to SysMgr software, you must obtain additional managed node SysMgr software licenses, one for every additional HP 3000 that you want to manage. The product numbers for the software licenses are: HP36937A (MPE/iX managed node software license) and HP36938A (MPE/V managed node software license).

You must also obtain two guides for completing the upgrade: *HP OpenView System Manager Manager's Guide* and *HP OpenView System Manager User's Guide*.

---

You do the upgrade in two parts:

- Upgrade your OVC PC software
- Upgrade your OVC HP 3000 software

The next sections give you the required steps for upgrading the OVC software.

---

**NOTE**

This appendix pertains only to upgrading the B.00.01 version of OVC PC software to the B.00.01 version of the OV System Manager PC software and to upgrading the A.01.03 version of the OVC HP 3000 software to the A.01.03 version of the OV System Manager HP 3000 software.

---

## Upgrading Your OVC PC Software

First you upgrade your OVC PC software. Follow these steps:

1. Start Windows on your OVC PC.
2. In the Program Manager, double-click on the StartUp group icon to open it.
3. Click on the Console program icon to select it, then press the **Del** key to delete the icon.
4. In the Program Manager, double-click on the HP OpenView program group to open it.
5. Click on the Console Setup program icon to select it, then press the **Del** key to delete the icon.
6. Use the same process to delete the OVC Read Me icon.
7. Use the File Manager application in the Main program group to delete the contents of the `c:\ov\sysmgr` directory.
8. Insert the OVC SysMgr floppy disk #1 into your A drive.
9. In the Program Manager, select the **Run** option from the **File** menu.
10. In the Command Line Box that is displayed, type:  

```
A:\SETUP
```

This command runs the installation program for OV SysMgr.
11. In the OV Console/SysMgr Setup dialog box, select **Multiple Node (HP OpenView System Manager)** as the type of installation you want to perform.
12. During the remainder of the installation, follow the instructions that are displayed to complete the SysMgr software installation.

## Upgrading Your OVC HP 3000 Software

After upgrading your OVC PC software, you then upgrade your HP 3000 software. Follow these steps:

1. Log on as MGR.SYSMGR to the HP 3000 management node that you were managing when using OV Console.

Make sure that you first shut down the OV Console software. If this software is still running, then issue this command to shut down the JCOMMON job:

```
run smutil.pub.sysmgr;info="STOP";parm=1
```

Then issue this command to stop the event collector:

```
TELLOP NMEV#EXIT
```

2. Purge all the files in the EMS.SYSMGR group.
3. Delete OVUDC from your catalog and setcatalog SMUDC.PUB.SYSMGR.

### Completing the Upgrade

To complete the upgrade, you must do these additional steps:

- Create a new map representing all your managed nodes on the PC.
- Install managed node software. (Make sure to install the managed node software on the management node also.)
- Start the management node software using the OVSTART UDC.
- Add managed node information in the configuration database using the managed node configuration dialog box on the PC.

For detailed information on these procedures, refer to the applicable sections in Chapter 3, "Installing SysMgr," in the *HP OpenView System Manager Manager's Guide*.

## A

add OVC symbols, 50  
application ID, 41  
ARS  
  Automated Response Service, 55  
  Automated Response Service  
    , 45  
ARS00000.EMS.SYSMGR, 33  
ARS00CMD.EMS.SYSMGR, 33  
ARS00EXE.EMS.SYSMGR, 33  
Automated Response Service  
  ARS, 55  
  ARS  
    , 45

## B

BAKQHEAD, 27  
BAKQPTER, 27

## C

commands  
  MPE TELLOP, 25  
  TELLOP, 42  
COMMMGR, 29  
CONFIG database, 62  
configuring  
  HP 3900 software, 21  
  OVC PC software, 19  
console.ovmNOTICE.SECURE.SYSMGR, 40  
CONVEXE, 62  
CONVEXE program, 63  
customizing maps, 40

## D

DBEMLEXE utility, 60  
DBUTIL program, 63  
DBWUB,EXE, 75  
DEF0LIST.EMS.SYSMGR, 33  
default map, 40  
definition file, 47  
DELLIST, 58

## E

EMG job, 21  
EMG process, 46  
EMG00000.EMS.SYSMGR, 33  
EMG0000X.EMS.SYSMGR, 33  
EMG00EXE.EMS.SYSMGR, 33  
EMG00JOB.EMS.SYSMGR, 33  
EMG00TPL.EMS.SYSMGR, 33

EMGINIT.EMS.SYSMGR, 34

## EML

  achieve, 27  
  database, 28  
  Event Message Log, 16  
  modify, 27  
EML database, 60  
EML0000X.EMS.SYSMGR, 34  
EMS000XL.EMS.SYSMGR, 34  
EMS00JOB.EMS.SYSMGR, 34  
EMS00TPL.EMS.SYSMGR, 34  
event collector, 40  
event gatherer, 42  
event marker, 40  
event marker format  
  , 41  
Event Message Log  
  EML, 16  
event posting, 30  
expired one, 28

## G

generate event, 40

## H

high end printer, 44

## I

INCS# marker, 42  
instrumented, 42  
intrinsic  
  MPE PRINTOP, 25  
  NMEVENT, 25, 40  
intrinsic  
  MPE PRINTOP, 42

## J

JCOMMON job, 21  
JEMLSWLG, 28

## L

LAUNCHER.EMS.SYSMGR, 34

## M

managed node, 33  
management by exception, 14  
Management Session Interface  
  MSI, 32  
management session interface  
  MSI, 24

matching events, 43  
message class, 41  
message dispatching, 43  
message severity, 41  
modify events, 44  
MPE PRINTOP intrinsic, 25, 42  
MPE TELLOP command, 25  
MPE00DEF, 43  
MPE00DEF.EMS.SYSMGR, 34  
MSGQHEAD file, 31  
MSI  
  Management Session, 56  
  Management Session Interface, 32  
  management session interface, 24

## N

NETALLOW, 56  
NETALLOW utility, 56  
NETALLOW.EMS.SYSMGR, 34  
NMEV# marker, 42  
NMEVENT intrinsic, 25, 40  
NOCAP, 29  
NTC00EXE utility, 55  
NTC00EXE.EMS.SYSMGR, 34

## O

object declaration record, 43  
OVC  
  defined, 14  
OVC icon files, 50  
OVC report generation, 28  
OVC report template, 28  
OVSEXE, 29  
OVSTART UDC, 76  
OVSTOP UDC, 76

## P

PRINTOP, 40

## R

Reflection 1, 32  
REMOVEXE  
  utility, 33  
REMOVEXE utility, 58  
report generation, 28  
requirements  
  hardware, 13  
  software, 13  
reserved symbol IDs, 52  
RSP00000.EMS.SYSMGR, 34  
RSP0000X.EMS.SYSMGR, 34

RSP000IN.EMS.SYSMGR, 34  
RSP000XL.EMS.SYSMGR, 34  
RSP000OUT.EMS.SYSMGR, 34

## S

SMCAT000.EMS.SYSMGR, 35  
SMUTIL, 65  
SMUTIL utility, 64  
standard templates  
  FORM00MN, 28  
  FORM00PC, 28  
static, 66  
symbol declaration record, 43  
symbol ID, 44  
symbols, 50  
  modify, 52  
  new map, 52  
  private declaration record, 52  
SYSEXE, 29  
SysMgr, 14  
SysMgr Communication Manager, 29  
  COMMMGR, 29  
SysMgr symbol ID, 51  
system requirements, 15

## T

technical description, 23  
TELL messages, 46  
TELLOP, 40  
TELLOP command, 42  
terminal emulator, 32

## U

UCSEXE, 29  
UCSEXE program, 29  
USER0DEF, 43  
USER0DEF.EMS.SYSMGR, 35  
USEREXE utility, 61  
utilities  
  DBEMLEXE, 60  
  NETALLOW, 56  
  REMOVEXE, 58  
  SMUTIL, 64  
  USEREXE, 61  
  VERCHECK, 57, 68  
utilitites  
  NTC00EXE, 55  
utility  
  REMOVEXE, 33  
utility programs, 54

---

## V

VERCHECK utility, 57  
VERCHECK utility, 29, 68  
VERCHECK.EMS.SYSMGR, 35  
VERCHECK.PUB.SYSMGR, 16  
version, 73  
VRCLOG00.EMS.SYSMGR, 35  
VRCLOG01.EMS.SYSMGR, 35  
VTC0000.EMS.SYSMGR, 35  
VTC000XL.EMS.SYSMGR, 35

## W

WINSOCK error, 72