System Automation for OS/390

AOC/MVS OPC Automation Operator's Guide and Scheduler Reference

Version 1 Release 4

SC23-3821-02

System Automation for OS/390

AOC/MVS OPC Automation Operator's Guide and Scheduler Reference

Version 1 Release 4

SC23-3821-02

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xi.

Third Edition (June 1999)

This edition applies to Version 1 Release 4 of the System Automation for OS/390 OPC Automation Feature (5685-151), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

FAX: (Germany) 07031-16-3456
FAX: (Other countries) (+49)+7031-16-3456

IBM Mail Exchange: DEIBMBM9 at IBMMAIL
Internet e-mail: s390id@de.ibm.com
World Wide Web: http://www.s390.ibm.com/os390/

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries:

AT	MVS/ESA	RMF
CICS	MVS/XA	System/370
DB2	NetView	System/390
IBM	PR/SM	VTAM
IMS	RACF	3090

About This Book

This book provides operator and scheduler information for the *IBM Automated Operations Control/MVS Operations Planning and Control (AOC/MVS) OPC Automation Feature, Version 1 Release 4*. Hereafter, this book refers to AOC/MVS OPC Automation as SA OS/390 OPC Automation. This is due to the fact that AOC/MVS 1.4 has been withdrawn and replaced by System Automation for OS/390 (SA OS/390).

SA OS/390 OPC Automation is a feature of SA OS/390 that brings together batch and online console automation into a common focal point. This automation feature automates, simplifies, and standardizes console operations and the management of component, application, and production related tasks.

Who Should Use This Book

The audience for this book includes:

- OPC administrators or schedulers
- Operators
- Programmers
- Evaluators

OPC Administrators or Schedulers

OPC administrators or schedulers use OPC to define information required for the automation interface. Chapter 2, “Scheduler’s Guide” on page 17 describes OPC Automation tasks performed by the administrators or schedulers.

Operators

Operators use OPC Automation commands and panels to:

- Handle OPC alerts
- Monitor systems and jobs
- Submit OPC requests

Chapter 3, “Operator’s Guide” on page 33 describes OPC Automation tasks performed by the operators.

The major difference between the scheduler and the operator is that the operator uses OPC Automation to perform OPC Automation tasks and the scheduler uses OPC to define OPC Automation required information.

Programmers

Programmers configure OPC Automation. Although the *AOC/MVS OPC Automation Programmer’s Reference and Installation Guide* was written especially for programmers, they may also need to perform some of the tasks described in this operator and scheduler reference.

Evaluators

Although the *AOC/MVS OPC Automation General Information* was written especially for evaluators, they may want to read this operator and scheduler reference to acquire a better understanding of SA OS/390 OPC Automation.

Prerequisite Knowledge

Familiarity with the functions and components of SA OS/390 as well as OPC would prove beneficial to understanding the concepts discussed in this manual. These IBM offerings work closely together to perform the automated console operations that are described.

What's in This Book?

This book contains the following chapters and appendix:

Chapter 1, OPC Automation Solution
Describes the benefits of SA OS/390 OPC Automation, which include:

- SA OS/390 OPC Automation's approach to automation
- What automation can do for your I/S operation
- The components of SA OS/390 OPC Automation

Read this chapter whether you are in production control or in operations.

Chapter 2, Scheduler's Guide
Describes the OPC tasks that scheduling or productions control staff perform using SA OS/390 OPC Automation. In cases where the scheduling staff also has NetView access, review Chapter 3, "Operator's Guide" on page 33.

Chapter 3, Operator's Guide
Describes the tasks that the operations staff does. In cases where the operations staff also has OPC access, review Chapter 2, "Scheduler's Guide" on page 17.

Chapter 4, Resynchronization and Recovery Considerations
Describes how interruptions are handled.

Appendix, Messages and Codes
Contains OPC Automation messages and codes.

A glossary of related terms and an index are also included.

What's New in This Book?

Changes to this book since Release 3:

- Modifications to support TME OPC Version 2 (APAR number OW35607)
- Modifications supporting Release 4 of SA OS/390 OPC Automation
- Technical changes reflecting service updates

A vertical bar (|) in the left margin indicates changes to the text and illustrations.

Related Publications

The SA OS/390 Library

The following table shows the information units in the SA OS/390 library:

Table 1. SA OS/390 Library

Title	Order Number
SA OS/390 General Information	GC28-1541
SA OS/390 Licensed Program Specifications	GC28-1540
SA OS/390 Planning and Installation	GC28-1549
SA OS/390 Customization	GC28-1566
SA OS/390 Operations	GC28-1550
SA OS/390 Messages and Codes	GC28-1569
SA OS/390 Technical Reference	GC28-1593
AOC/MVS CICS Automation General Information	GC23-3813
AOC/MVS CICS Automation Operator's Guide	SC23-3815
AOC/MVS CICS Automation Programmer's Reference and Installation Guide	SC23-3814
AOC/MVS IMS Automation General Information	GC23-3816
AOC/MVS IMS Automation Operator's Guide	SC23-3818
AOC/MVS IMS Automation Programmer's Reference and Installation Guide	SC23-3817
AOC/MVS OPC Automation General Information	GC23-3819
AOC/MVS OPC Automation Operator's Guide and Scheduler's Reference	SC23-3821
AOC/MVS OPC Automation Programmer's Reference and Installation Guide	SC23-3820

The System Automation for OS/390 books (except Licensed Program Specifications) are also available on CD-ROM as part of the following collection kits:

- IBM Online Library OS/390 Collection (SK2T-6700)
- IBM Online Library Networking Collection (SK2T-6012)

These softcopy collections include the IBM Library Reader, a program that enables you to view online documentation.

SA OS/390 Homepage

For the latest news on SA OS/390, visit the SA OS/390 homepage at http://www.s390.ibm.com/products/sa/

Related Product Information for the Base Program

The following table shows the books in the related product libraries that you may find useful for support of the SA OS/390 base program.

Table 2 (Page 1 of 2). Related Product Books

Title	Order Number
MVS/ESA MVS Configuration Program Guide and Reference	GC28-1817
MVS/ESA Planning: Dynamic I/O Configuration	GC28-1674
MVS/ESA Support for the Enterprise Systems Connection	GC28-1140
MVS/ESA Planning: APPC Management	GC28-1110
MVS/ESA Application Development Macro Reference	GC28-1822
MVS/ESA SP V5 System Commands	GC28-1442
MVS/ESA SPL Application Development Macro Reference	GC28-1857
NetView for MVS V3R1 Administration and Security Reference	SC31-8045
NetView for MVS V3R1 Automation Implementation	SC31-8050
NetView for MVS V3R1 Automation Planning	SC31-8051
NetView for MVS V3R1 Command Reference	SC31-8047
NetView for MVS V3R1 Customization Guide	SC31-8052
NetView for MVS V3R1 Customization: Writing Command Lists	SC31-8055
NetView for MVS V3R1 Installation and Administration Guide	SC31-8043
NetView for MVS V3R1 RODM and GMFHS Programming Guide	SC31-8049
NetView for MVS V3R1 User's Guide	SC31-8056
NetView for MVS V3R1 Tuning Guide	SC31-8048
OS/390 Hardware Configuration Definition: User's Guide	SC28-1848
OS/390 Information Roadmap	GC28-1727
OS/390 Information Transformation	GC28-1985
OS/390 Introduction and Release Guide	GC28-1725
OS/390 V1R2.0 JES Commands Summary	GX22-0041
OS/390 Licensed Program Specifications	GC28-1728
OS/390 Printing Softcopy Books	S544-5354
OS/390 Starting Up a Sysplex	GC28-1779
OS/390 Up and Running!	GC28-1726
Planning for the 9032 Model 3 and 9033 Enterprise Systems Connection Director	SA26-6100
Resource Access Control Facility (RACF) Command Language Reference	SC28-0733
S/390 MVS Sysplex Overview – An Introduction to Data Sharing and Parallelism	GC23-1208
S/390 MVS Sysplex Systems Management	GC23-1209
S/390 Sysplex Hardware and Software Migration	GC23-1210
S/390 MVS Sysplex Application Migration	GC23-1211

Table 2 (Page 2 of 2). Related Product Books	
Title	**Order Number**
S/390 Managing Your Processors	GC38-0452
TSO/E REXX/MVS Users Guide	SC28-1882
TSO/E REXX/MVS Reference	SC28-1883
VSE/SP Unattended Node Support	SC33-6412
VSE/ESA 1.1.0 Unattended Node Support	SC33-6512
VTAM Version 3 Release 3 Network Implementation Guide	SC31-6404
VTAM Version 3 Release 4 Network Implementation Guide	SC31-6434

Related Product Information for Workstation Operations

The following are the books in the related product libraries that you may find useful for support of SA OS/390 workstation operations.

Table 3. Related Product Books	
Title	**Order Number**
APPC System Definitions in MVS/ESA and OS/2	GG66-3224
APPC Programming Considerations	GG24-3818
APPC Application Examples	GG24-3819
Distributed Console Access Facility User's Guide	GE13-0061
IBM Communications Manager/2 Version 1.1	G221-3630
IBM Communications Manager/2 Version 1.1 Information and Planning Guide	SC31-7007
IBM Communications Manager/2 Version 1.1 Workstation Installation Guide	SC31-6169
IBM Communications Manager/2 Version 1.1 Configuration Guide	SC31-6171
IBM Communications Manager/2 Version 1.1 User's Guide	SC31-6108
IBM Operating System/2 Version 2.1 Using the Operating System	S61G-0905
IBM Operating System/2 Warp	SR28-5668
NetView for MVS V3R1 Graphic Monitor Facility User's Guide	SC31-8095
Official Guide to Using OS/2 Warp	SR28-5659
Personal Communications Programmer's Guide	SC31-8660
Personal Communications Reference	SC31-8259
Personal Communications Tell Me About OS/2 Access Feature	SC31-8257
Personal Communications Up and Running	SC31-8258

Chapter 1. OPC Automation Solution

Operations Planning and Control/Advanced (OPC/A), Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA) or TME 10 Operations Planning and Control (TME 10 OPC) submits, tracks, and recovers the execution of batch work through a job entry system (JES) interface. NetView implements SA OS/390, and serves as the basis for automated console operations. SA OS/390 OPC Automation capitalizes on the strengths of NetView, SA OS/390, and OPC by providing the ability to greatly expand job execution, scheduling, monitoring, and alert notification capabilities.

Note: For consistency and clarity, this document uses the term OPC to refer to OPC/A, OPC/ESA and TME 10 OPC. Similarly the term OPC Controller refers to OPC/A PCS, OPC/ESA Controller or TME 10 OPC Controller. AOC OPC Automation supports all products.

OPC Automation's Approach to Automation

With OPC Automation, NetView can use OPC calendar information to achieve a single-calendar definition that handles multiple systems and sites. A change in the OPC calendar can affect all the systems, ensuring consistency throughout the systems complex.

System Automation for OS/390 (SA OS/390) automates MVS console operations and provides the base for further automation when used with the NetView Solutions family of program offerings. SA OS/390 is a powerful application, designed to greatly reduce the time and effort required to meet automation objectives.

This approach to automation combines NetView and OPC with an SA OS/390 and OPC Automation. It provides a function that does not exist in any of these applications alone. Thus, the end result of combining these applications in an automated environment far exceeds the capabilities of these products when used individually.

Basic Concept

Large, complex systems frequently require comprehensive schedules. There are regular workdays, other workdays (weekends), and complicated business cycles that take into consideration events such as:

- Holidays
- Financial quarter-end processing
- Sales promotions
- Maintenance
- Product development phases
- Testing

NetView does not easily lend itself to implementing these types of calendars. However, OPC has excellent calendar-management capabilities.

OPC Automation's basic concept consists of moving the management of functions that require calendar control from NetView and SA OS/390, to OPC, even if no batch component exists. This ensures a single point of control and eliminates

problems resulting from a loss of synchronization in calendars between OPC and NetView.

Examples of Placing Calendar Control

You can control startup and shutdown with SA OS/390 or with the OPC calendar functions. The following are examples of where to place this control:

- In the first situation, TSO is scheduled for availability at all times, regardless of dates or time of day. Here TSO is defined using the SA OS/390 control file, since calendar-specific control is not required.
- In the second case, TSO is required for specific hours on business days and different hours on weekends and holidays. Since SA OS/390 alone does not easily encompass calendar-specific events, you should define this in OPC and tie into SA OS/390 with OPC Automation. This approach offers a single point of control for all automated events.

SA OS/390

SA OS/390 provides automated console operation functions that are implemented through NetView CLISTs, command processors, message tables, and panels. These automation capabilities address the majority of subsystem and component automation requirements. They are an integral part of NetView and the MVS operating system. The automation of local resources in the operating system provides the primary focus of this approach.

When multiple MVS systems are interconnected and require consolidated operations at one focal-point system, you can configure SA OS/390 to communicate automation-related status and commands to and from that focal-point system. This enables you to view status from multiple systems on a single system which acts as a focal-point.

SA OS/390 provides the capability to automatically start, monitor, and terminate MVS subsystems, components, and applications, such as JES2, JES3, VTAM, TSO/E, IMS, CICS, DB2, RMF, NetView, and many others.

OPC defines a workstation as a unit or place that performs a specific data processing function. Examples of workstations include JCL preparation, data entry, CPUs, and printers. Activities that occur on workstations are referred to as operations. AOC OPC Automation extends the idea of workstations to include NetView. Each NetView with AOC/MVS in your enterprise is represented by an OPC workstation. An OPC workstation may also represent all NetViews running SA OS/390 within the same sysplex where the OPC controller is running. These NetView workstations then schedule and perform operations on behalf of batch applications.

SA OS/390 Control File

A control file defines the scope of the automation that is performed. The control file supplied with SA OS/390 contains a basic sample set of subsystem and component definitions. This control file is designed to be easily expanded for specification of additional subsystems and components.

OPC Automation is an extension of SA OS/390. To implement OPC Automation, the following control file entries were added:

- OPCA CODE
- OPCACMD
- OPCA DOMAINID
- OPCAPARM
- OPCA PCS
- ENVIRON OPCAO

OPCA CODE

The OPCA CODE entry defines the parameters used for various requests. This entry is coded for each subsystem. For example:

```
RMF  OPCA,CODE=(START,,,'UP,3,RMFUTMER')
```

OPCACMD

The OPCACMD entry specifies the actual automation command that is issued for a request. This entry is coded for each subsystem. For example:

```
RMF  OPCACMD,CMD=(START,,'SETSTATE RMF,RESTART,START=YES')
```

OPCA DOMAINID

| The OPCA DOMAINID entry relates an OPC automatic workstation to either a
| specific NetView domain ID or collectively to all NetView domains on the local
| sysplex. For example: :

```
OPCA    DOMAINID,
            CODE=(NV06,,,AOF06),
            CODE=(NV00,,,AOF01),
            CODE=(NV08,,,SYSPLEX),
            CODE=(NV01,,,XBAOF)
```

OPCAPARM

The OPCAPARM entry defines the parameters used for various requests. This entry is coded for each subsystem. For example:

```
RMF  OPCAPARM,CODE=(START,,,',,')
```

OPCA PCS

| The OPCA PCS entry specifies either the NetView domain on which the OPC
| Controller resides or that the local sysplex is to be searched for the active controller
| when required. It must also specify the MVS subsystem name for the OPC
| controller. For example:

```
OPCA   PCS,
        DOMAIN=AOF01,
        SUBSYS=OPCC
```

ENVIRON OPCAO

The ENVIRON OPCAO entry specifies certain system-wide defaults, for:

- Retention of critical messages (MSGKEEP)
- Determining if operations can be reset after NetView has been unavailable (OPCRESET)
- Checking subsystem status before allowing requests to proceed (REQSTAT)

For example:

```
ENVIRON OPCAO,REQSTAT=YES,
              MSGKEEP=04:00,
              OPRESET=00:30
```

Refer to *AOC/MVS OPC Automation Programmer's Reference and Installation Guide*, for a complete description of the OPC automation control file entries.

For a complete description of the base SA OS/390 automation control file entries, refer to *System Automation for OS/390 Customization*.

NetView Automation Table

SA OS/390 monitors messages received and compares them with those in the NetView automation table, formerly called the *message table*. When a message occurs that ordinarily requires manual operator intervention, such as responding to an outstanding WTOR, the control file directs a predefined response to the MVS console without operator intervention.

OPC Automation Main Menu Panel and Tutorial

Although you can enter OPC Automation commands from any NetView panel, the OPC Automation Main Menu panel is helpful since it lists the most commonly used commands and provides access to tutorial information.

To display the OPC Automation Main Menu panel, type the following from any NetView command line:

```
OPC
```

After you press ENTER, OPC Automation displays the AOC OPC Main Menu panel, as shown in Figure 1 on page 5.

```
EVJT0000                A O C   O P C    M A I N    M E N U

   TYPE    COMMAND             DESCRIPTION                      TUTORIAL

                               AOC OPC Automation Feature          1

    P      SDF                 Display facility additions          2

    P      OPCAQRY             Display status of operations        3
    L      OPCAPOST            Manually post status to OPC         4
    P      OPCACMD             Display and modify OPC data         5
    L      SRSTAT              Update special resource status      6

    L      UP                  Start subsystem                     7
    L      DOWN                Terminate subsystem                 8
    L      $RESTART            Recycle subsystem                   9

Note: Commands of type L (linemode) do not have an input panel
      and must be invoked with all required parameters specified.


Enter a Tutorial Number or Command ==>
          PF1= Help   PF2= End   PF3= Return
          PF6= Roll
```

Figure 1. AOC OPC Main Menu Panel

Using the AOC OPC Main Menu Panel and Tutorial

Accessing the Tutorial

Each number on the AOC OPC Main Menu panel corresponds to an adjacent topic displayed on the panel under the description column as shown in Figure 1 on page 5. To use the OPC Automation tutorial, type a number (1–9) on the command line and press ENTER.

Issuing OPC Automation Commands

To issue an OPC Automation command, enter the command on the command line of the Main Menu panel. To assist you, this panel displays the most commonly used commands under the command column, as shown in Figure 1 on page 5.

OPC Automation commands are used to:

- Start up, shut down, or restart subsystems.
- Review the status of an entire system complex. This is described in “Status Display Facility” on page 12.
- Communicate with OPC.

All of the OPC Automation commands are described in this operator and scheduler reference.

Automating I/S Operations

Once installed, OPC Automation allows OPC to issue requests that perform complex startup, shutdown, or restart activities that are not handled efficiently by OPC alone. In addition, SA OS/390 utilizes the far more sophisticated scheduling capabilities of OPC, thus ensuring comprehensive event management with OPC-defined rules and calendars. OPC Automation also extends the SA OS/390 Status Display Facility to include status information on components, such as tapes, batch jobs, or OPC-detected errors and alerts. Operations personnel can then use a single NetView monitor to observe the health of an entire complex.

The major strength of OPC is its ability to control multiple streams of related activities without logical limits on the number of streams or the time required to execute those streams. With dependency control, OPC can separate each stream into many parallel substreams and, at a later time, bring them back together into a smaller number of streams or a single stream. During this processing, sophisticated mechanisms detect deviations and either correct them or bring them to the attention of the administrator.

Another important strength of OPC is that it uses an intuitive dialog, which allows nontechnical personnel access to a controlled set of scheduling functions.

NetView has a powerful and integrated interface to MVS console services. NetView communicates with the operating system or subsystems, issues commands, analyzes the responses, and reacts to unsolicited messages.

SA OS/390, a comprehensive automation package, uses basic NetView capabilities and replaces many of the functions previously performed by console operators.

However, OPC lacks the ability to interact with console services. Although you can issue commands to the system console, OPC cannot receive responses and evaluate them. OPC can ask for an action, but cannot determine if the action is completed successfully or if problems are encountered.

On the other hand, with only SA OS/390, NetView is not designed to manage complex streams of related operations in the same way that OPC is. NetView manages simple streams, such as single sets of procedures, by keeping control data in variables. This methodology becomes cumbersome when the streams are complex. Also, SA OS/390's scheduling capabilities are less sophisticated than those provided with OPC.

SA OS/390's strengths include operating and controlling the production environment. OPC provides long-term planning and control capabilities for production work load management. With SA OS/390 OPC Automation, these functions work together to enable you to expand automation capabilities as described in the following topics.

Reduce Operator Intervention

SA OS/390 OPC Automation aids in reducing or eliminating operator intervention. You can now fully automate to assure correct termination of online systems prior to batch processing. SA OS/390 is designed to automate the normal operation of OPC.

Simplify Complex Operations

Large, complex systems operations often utilize two distinct groups of personnel to perform the following:

- Control production batch work and direct its scheduling
- Manage the production online systems that are generally controlled by master terminal operators

Changes in either group can dramatically affect the working conditions of the other. Untimely or unclear communications sometimes occur between these two groups. Unfortunately, this can lead to errors, loss in productivity, and frustration.

With OPC Automation, either group can modify a common control facility that implements changes automatically in either process. No interoperator communication is required. In fact, in a highly automated environment, no operator intervention or awareness of these interrelated changes is necessary.

Simplify Tasks Across Entire Complexes

Across entire system complexes, you can easily implement the following time-consuming tasks:

- Changes to the online hours of availability of specific applications
- Cycling of individual online databases
- Changes to scheduled test times

With proper planning, OPC Automation allows users to define disaster recovery procedures that are totally automated.

System Initialization with OPC Automation

JES starts OPC, which is usually operational at all times, as a task without SA OS/390. OPC Automation then transfers the responsibility of starting OPC from JES to SA OS/390, as described in the following scenario:

- The OPC Tracker has JES as a parent.
- During the IPL process, as soon as JES is running, SA OS/390 issues a start command for the Tracker subsystem.
- Once the Tracker has started, SA OS/390 issues a start command for the Controller on the control host(s) only.
- Automation continues to initialize the rest of the tasks that are defined to it. SA OS/390 OPC Automation restores the status of any OPC-controlled tasks to the last status requested by OPC and waits for OPC to issue new requests.

NetView Interface to OPC Automation

The program-to-program interface (PPI), a high-performance interface, provides synchronization and bidirectional command and message flow between NetView and other applications. OPC provides additional application programming interfaces (APIs), which allow it to be updated by other programs.

The implementation of this interface in OPC Automation also provides the following capabilities:

- Automation of OPC startup and termination
- Interception of OPC alerts for analysis by the alert operator
- Expansion of the Status Display Facility to provide information about TSO users, batch jobs, critical messages, outstanding tape mounts, and OPC errors
- Implementation of a two-way interface between OPC and NetView with SA OS/390:
  - OPC defines and controls interactive applications. Support is provided to start and stop subsystems that are defined to SA OS/390 application.
  - Database tasks can run in both interactive and batch systems with full synchronization between the activities.
  - SA OS/390 can access OPC calendars and other information.
  - NetView operators can access and update OPC-defined applications without the need to log on to OPC.

- Two user extensions:
  - OPCACOMP allows the startup and shutdown of subsystems not controlled by SA OS/390.
  - UX*nnnnnn* allows automation of activities not associated with a specific subsystem.

OPC Automation provides commands and panels that allow a NetView operator to make inquiries and issue requests to OPC without actually logging on to OPC.

For example, if you wanted to display OPC detail information from a NetView console, enter the following command from any NetView command line:

OPCACMD

After you press ENTER, OPC Automation displays the AOC OPC: Display or Modify OPC Data panel (EVJKAC01), as shown in Figure 2.

```
EVJKAC01         AOC OPC: Display or Modify OPC Data
                                                          Date: 05/17/95
                                                          Time: 14:36:36

Specify search criteria and press ENTER

  Subsystem    :  OPCC                From ACF-file

  Application  :  rmf*_____________   Can be generic
  Opno         :  ____                Numeric
  Jobname      :  ________            Can be generic
  Wsname       :  ____                Can be generic
  Group        :  ________            Can be generic
  Owner        :  ________            Can be generic
  Priority     :  _                   1-9 (1=low, 9=high)
  Errcode      :  ____                Can be generic
  Status       :  _                   A/W/S/R/C/I/E/U



Action===>
F1= Help     F2= End      F3= Return                          F6=Roll
```

Figure 2. AOC OPC: Display or Modify OPC Data Panel

Note: This presentation shows how AOC OPC panels interface with OPC. For further details, see “SDF—Status Display Facility Panel” on page 44 and “OPCACMD—Interacting Dynamically with OPC” on page 56.

To list all of the applications defined to OPC on this system, type an asterisk (*****) in the application field, as shown in Figure 3.

```
Application  :  *_____________    Can be generic
```

Figure 3. Generic Search Function

Although the generic search function is used here for purposes of illustration, this operator and scheduler reference can help you select your list more specifically.

After you press ENTER, OPC Automation displays the Display or Modify OPC Data panel (EVJKAC03), as shown in Figure 4.

```
EVJKAC03             Display or Modify OPC Data              Page:   1  of   1
                                                             Date: 05/17/95
CMD: C change  B browse                                      Time: 14:36:42
     H hold  R release  N no-op  U unno-op
CMD  Application      Jobname   Ws  Opno St   Inp. Arr   Description  H    N
---  ---------------- -------- ---- ---- -    ---------- -----------  -    -
 _   RMFDLY           RMF      NV11 0001 C    9505262230 STOP         N    N
 b   RMFDLY           RMFMAINT CPU1 0002 E    9505262230              N    N
 _   RMFDLY           RMF      NV11 0003 W    9505262230 START        N    N
 _   RMFDLY           RMF      NV11 0001 A    9505302200 STOP         N    N
 _   RMFDLY           RMFMAINT CPU1 0002 W    9505302200              N    N
 _   RMFDLY           RMF      NV11 0003 W    9505302200 START        N    N
 _   RMFDLY           RMF      NV11 0001 A    9506062200 STOP         N    N
 _   RMFDLY           RMFMAINT CPU1 0002 W    9506062200              N    N
 _   RMFDLY           RMF      NV11 0003 W    9506062200 START        N    N
 _




Action===>
F1= Help     F2= End      F3= Return                 F5= Refresh   F6= Roll
```

Figure 4. Display or Modify OPC Data Panel

To display detail information about application MAINT, type **B** in the CMD field.

After you press ENTER, OPC Automation displays the OPC Occurrence Data panel (EVJKAC02), as shown in Figure 5.

```
EVJKAC02             OPC Occurrence Data
                                                          Date: 05/17/95
                                                          Time: 14:38:30

Subsystem    : OPCC              Deadline  : 95/05/26 23:30
Application  : RMFDLY            OPIA date :   /  /     OPIA time :   :
Opno         : 0002              Edur      : 0005       PS reqd   : 0001
Inp arrival  : 95/05/26 22:30    Job Class : P          R1 reqd   : 0000
Jobname      : RMFMAINT          Auto Sub  : Y          R2 reqd   : 0000
Wsname       : CPU1              AJR       : Y          A E C     : Y    Y/N
                                 Clate     : N          Timedep   : N
                                 Form no   : ____       Hi RC     : 0000
Priority     : 5                 Reroute   : N          Restart   : Y
Error Code   : SB37              DeadWTO   : N          Cat mgt   : N
Status       : E               Manual hold : N          NOP       : N
                               Description : ________________________
                               User data   : ________________



Action===>
F1= Help     F2= End      F3= Return                          F6=Roll
```

Figure 5. OPC Occurrence Data Panel

Status Display Facility

SA OS/390 uses the Status Display Facility to provide a central focus of information. The Status Display Facility describes the status of all automated systems and applications of a NetView complex, such as:

- A consolidated, hierarchical view of an entire operating environment with detailed information where required.
- Dynamically updated status panels that use color, representing system and application status to enhance usability and to expedite comprehension of priority information.
- A central repository for the status of automated resources and components.
- A facility for viewing the status of multiple target systems by a focal-point system operator.
- Simplified techniques for presenting and maintaining resource status for multiple systems, subsystems, and applications. For example:
  - Multiple programs can asynchronously update the Status Display Facility with subsystem status without concern for sequence or priorities posted by other programs.
  - The Status Display Facility resolves the priority of conflicting statuses and displays the most severe status.
  - When a problem is resolved, the program resolving the problem can update or clear the condition previously set without regard for any other status posted by other programs. The status of the application is automatically updated to the current status or most serious problem as appropriate.

OPC Automation provides the following additional fields and detail panels for the Status Display Facility:

CRITMSG Critical messages

TAPES Outstanding tape mounts

TSOUSERS TSO users logged on

BATCH Batch jobs being executed

OPCERR OPC-detected errors

To use the Status Display Facility, refer to “Monitoring the Health of the System” on page 34.

Request and Confirmation Transaction Flow

Figure 6 shows the flow from an OPC application requested action through to NetView and the return confirmation of the action. This example illustrates the request to start the resource management facility (RMF), located in a remote host with a NetView domain identifier of NVREG. OPC contains a representation of this host with a workstation definition of NV04. The request to start RMF is part of an OPC application known as MAINT. In Figure 6, the jobname is specified as RMF and the operation text is START.

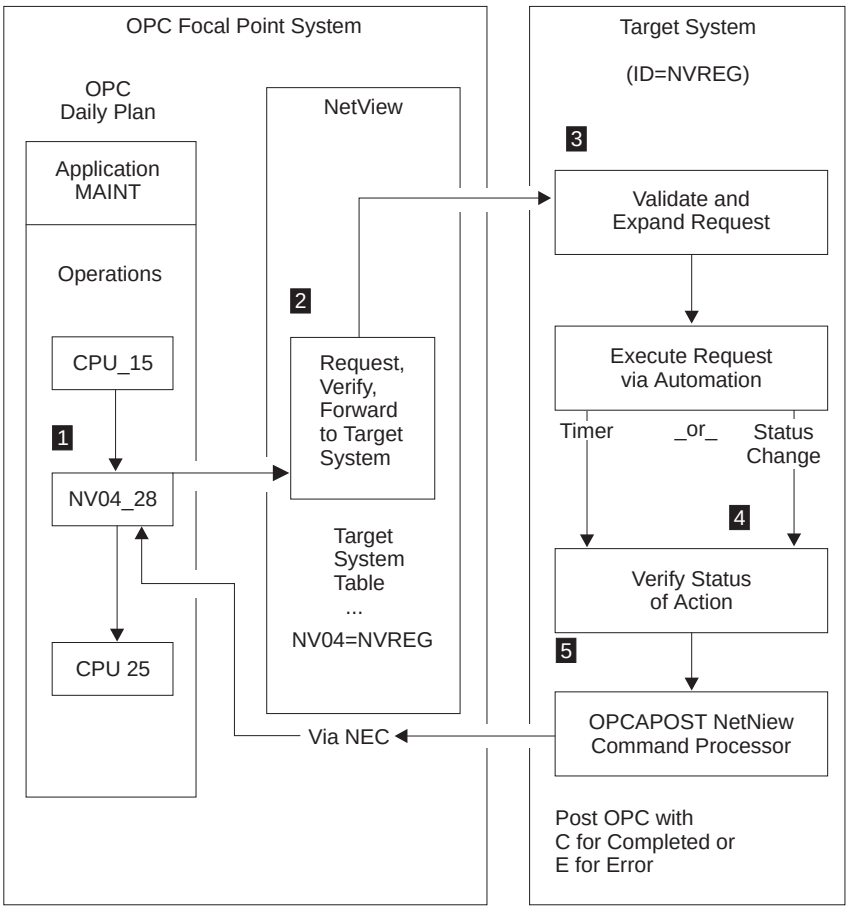

Figure 6. NetView–OPC Interface Flow. Syntax and definition errors, target system availability, recovery, and resynchronization via OPC API and NetView PPI are not shown in this example.

The OPC application named MAINT is defined to OPC, using dependency control, to ensure an orderly flow of operations. NV04 defines an OPC automatic general workstation which is resolved by NetView into the target NetView domain ID through OPC Automation parameter definitions.

Figure 6 shows CPU_15 as the last batch step which needs processing prior to starting RMF. Once this completes properly, OPC dependency control makes the NV04_20 operation ready on the NV04 workstation. This causes the request to start RMF which is then forwarded to NetView Domain NVREG.

OPC Automation uses the NetView PPI to transfer the request from OPC to NetView. This transfer of the NetView request from OPC to NetView is through the use of the status change exit (exit 7) in OPC.

1 The NetView PPI passes the request buffer to the PPI dispatcher task in SA OS/390. This task dispatches the request to the OPC Automation verify routine, which translates the workstation name into the NetView domain ID through definitions in the SA OS/390 control file.

2 The request is forwarded to the appropriate NetView domain for execution.

3 The target NetView translates the request text into MVS console commands using information stored in the automation control file. In Figure 6 on page 13, the request function translates the request buffer to the SA OS/390 function, which then starts RMF, but not an MVS START command. The automation control file entry to start RMF is:

```
RMF OPCACMD,CMD=(START,,'SETSTATE RMF,RESTART,START=YES')
```

Functions other than START and STOP of systems based on SA OS/390 may require user programming.

The command is dynamically generated using definitions in the SA OS/390 control file. A check determines whether the command is properly accepted. During this process, WTOs and the OPCAPOST command report errors. OPCAPOST sends an error indication back to OPC. If the command is issued correctly, a timer request is made. The timer intercepts a condition, where the request does not execute in a reasonable amount of time, which is user-selectable.

4 A change-of-status SA OS/390 function intercepts all changes-of-status. This allows the completion of outstanding requests as soon as the request is executed.

5 When the request is completed, the OPCAPOST command processor is invoked. OPCAPOST calls EQQUSINT/DRKUSINT which passes the completion code to the Tracker (EMS) on this system. The Tracker (EMS) forwards the completion code to the OPC Controller.

In a user-supported function, the timer and completion validation are a user responsibility. Once the user code determines that the function is completed, the OPCACOMP function is called. This function assures that actions are accomplished in the correct sequence, performs some housekeeping, returns a good or bad completion code, and calls the OPCAPOST command processor.

This terminates the processing for this specific OPC operation. If the request is executed without problems, the operations status is set to C (completed) and normal OPC dependency control allows the next operation to start. See the CPU_25 batch job in Figure 6 on page 13.

If the operation completes in error, an E status and a 4-character return code is set, and the application does not continue processing until a person or OPC intervenes.

Errors reset by OPC Automation are the result of regained availability of a target NetView domain to which communications are lost. The error codes are set to:

- U*xxx* when human intervention is required.
- S*xxx* when OPC Automation attempts to recover. This occurs when an operation that did not complete properly is resolved and completed.

OPC Automation Log Entries

Before processing takes place, the request buffers received by OPC Automation from OPC are copied to the NetView log for tracking purposes, such as verification of correct operations and error logging.

Figure 7 shows a request buffer log entry. In this example, OPC is processing the MAINT application. An OPC Controller running on domain NVDOM has made a request to SA OS/390 to perform START for the RMF subsystem. When this action is completed, SA OS/390 changes the status of the NV00_10 operation in the MAINT application to C (completed).

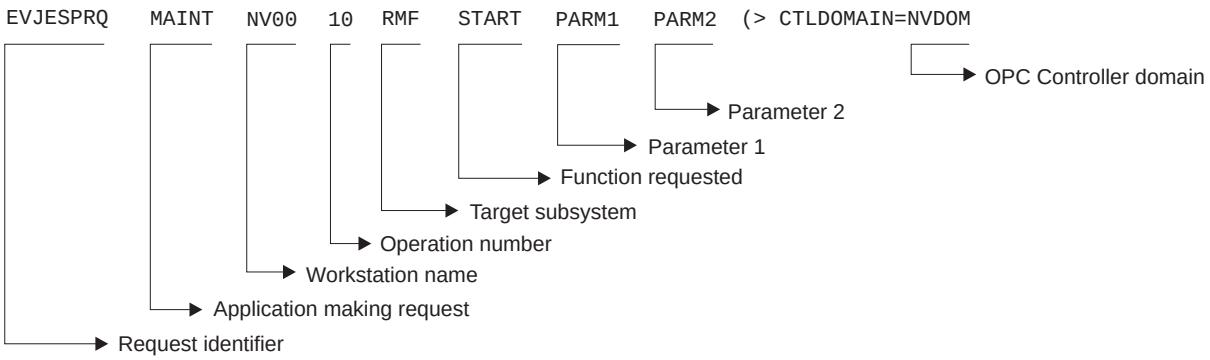

Figure 7. NetView Log Entry of an OPC Generated Request

Chapter 2. Scheduler's Guide

This chapter describes what you need to do when you define an application in OPC that you can automate using OPC Automation.

Note: This chapter contains some specific details about how to define applications and other items to OPC. However, it does not explain basic OPC functions because it assumes that you have a prerequisite knowledge of OPC and will refer to the OPC documentation when necessary. See "Prerequisite Knowledge" on page xiv.

In cases where the scheduling staff also has NetView access, review Chapter 3, "Operator's Guide" on page 33. This will help you to see the correlation between the information you define in OPC and how that information is used in OPC Automation.

Defining OPC Automation Required Information in OPC

The information that is passed to OPC Automation is entered in standard OPC description fields. This information is used to route requests where they are verified and executed. When the request is completed, a status change for the operation is sent back to OPC. The minimum information that needs transferring to accomplish this is listed in Table 4.

Table 4. OPC Automation Items defined in OPC

Definition in OPC	Information item	Refer to
General reporting workstations that represent target NetView domains	OPC workstation ID representing the NetView domain ID	"Defining the Target NetView Domains" on page 18
OPC-defined application making requests to SA OS/390 OPC Automation	Application name	Application field in Figure 10 on page 20
Target subsystem, such as RMF, for which this function is executed	Job name	Job name field in Figure 10 on page 20
Operations executed within a job	Operation number or numbers	No. field in Figure 10 on page 20
Task and task parameters to be performed for the specific operation	A task, such as STOP or START, and optional parameters	Operator text field in Figure 11 on page 21 and Figure 12 on page 21

Defining the Target NetView Domains

To define an OPC request that OPC Automation can use, a workstation representing the target NetView domain is required. This workstation, which must begin with the letters 'NV', should be a general, automatic reporting workstation defined with OPC using the standard OPC dialogs. Use the following naming convention where *xx* are any two characters:

NV*xx*

Note: Reserve NV*xx* workstations for OPC Automation. Unpredictable and undesirable results may occur if these workstation names are used other for workstations.

Figure 8 shows a typical definition.

```
---------------------- BROWSING A WORK STATION DESCRIPTION ----------------------
 Command ===>

 Enter the command R for resources or A for availability above.

 Work station          : NV00
 Description           : NetView AOFS6 for test

 Work station type     : General
 Reporting attribute   : Automatic
 Printout routing      : SYSPRINT
 Control on servers    : No

 Splittable            : No
 Job setup             : No
 Sub/Rel data set      :

 Transport time        :  0:00
 Duration              :

 Last updated by       : COLEY    on 05/08/95 at 09:54
```

Figure 8. Sample NVnn Workstation Definition

Entries in the SA OS/390 control file translate NV*nn* to an actual NetView domain ID. The programmer defines these entries. In this manner, the scheduler does not need to know the NetView domain ID names, but rather works with the workstation representation of those names. This allows changes to the relationship of workstations to NetView domain IDs without modifying the OPC definitions.

Figure 9 shows an example for a 4-processor environment.

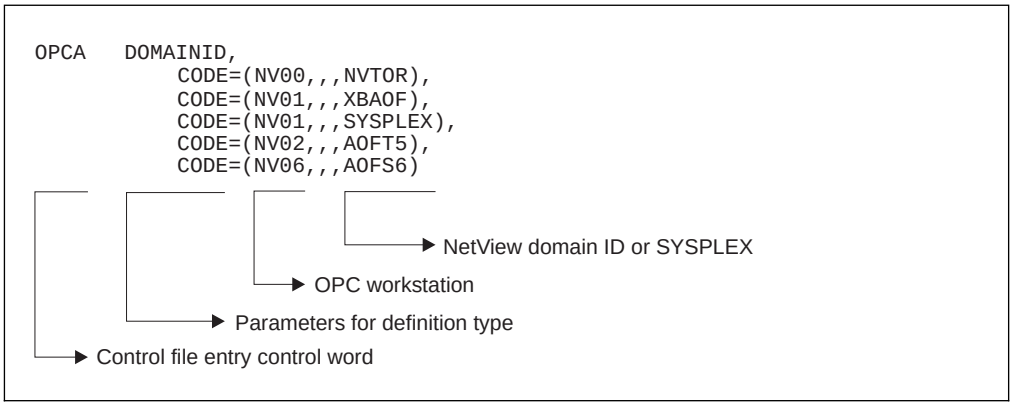

Figure 9. Control File Entry Defining a NetView Domain. In this example, NetView domain ID NVTOR is defined in OPC as NV01.

Note: You may use OPC database management dialogs or batch loader jobs to define the NV*nn* workstations.

Defining Requests in OPC

Standard OPC application description panels are used to define OPC Automation requests. The following items are defined:

- Application making the request
- Function requested

Application Making the Request

The application making the request is defined to OPC with operations specified on the NV*nn* workstation, as shown in Figure 10.

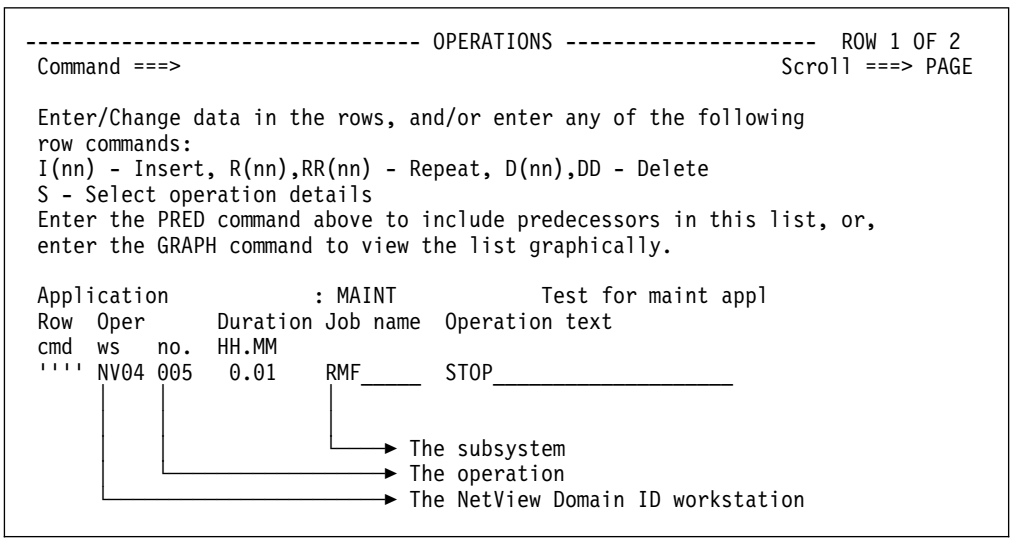

Figure 10. Defining the MAINT Application in OPC Automation

Function Requested

The function requested and the request parameters, if any, are not standard OPC definitions. Enter these fields in the operation text field using blanks as delimiters.

Figure 11 shows an example of how to define a request to stop and start RMF.

```
--------------------------------- OPERATIONS ---------------------  ROW 1 OF 2
Command ===>                                                    Scroll ===> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
S - Select operation details
Enter the PRED command above to include predecessors in this list, or,
enter the GRAPH command to view the list graphically.

Application            : MAINT            Test for maint appl

Row  Oper      Duration Job name  Operation text
cmd  ws   no.  HH.MM
'''' NV04 005   0.01    RMF_____  STOP____________________
'''' NV04 010   0.01    RMF_____  START___________________
```

Figure 11. Operations Text OPC Panel Showing OPC Automation Requests

OPC Automation permits the inclusion of two optional parameters in the request buffer. The target system builds the required command with these parameters, using information contained in the SA OS/390 control file. Alternatively, the parameters pass control information to optional user-written modules. Figure 12 shows an example of the request using these optional parameters.

```
--------------------------------- OPERATIONS ---------------------  ROW 1 OF 2
Command ===>                                                    Scroll ===> PAGE

Enter/Change data in the rows, and/or enter any of the following
row commands:
I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
S - Select operation details
Enter the PRED command above to include predecessors in this list, or,
enter the GRAPH command to view the list graphically.

Application            : FORCE            Test for maint appl

Row  Oper      Duration Job name  Operation text
cmd  ws   no.  HH.MM
'''' NV04 005   0.01    RMF_____  STOP FORCE IMM__________
```

Figure 12. Request Using Optional Parameters

Displaying OPC Automation Requests in OPC

Since OPC Automation requests are stored as operator text, you can view them in OPC, as shown in Figure 13.

```
------------------------------ BROWSING OPERATIONS ----------------  ROW 1 OF 2
Command ===>                                                      Scroll ===> PAGE

Enter the PRED command above to include predecessors in this list, or,
enter the GRAPH command above to view operations graphically.
Enter the row command S to select the details of an operation.

Application            : RMFBKUP           RMF Backup Processing

Row  Oper      Duration Job name  Operation text
cmd  ws   no.  time
'    NV04 005   0:01    RMF       STOP
'    NV04 010   0:01    RMF       START
****************************** BOTTOM OF DATA ********************************
```

Figure 13. Browsing Operations Including OPC Automation Requests

The following list defines several of the fields that are shown on the panel in Figure 13.

NV04	Represents the target NetView domain or the sysplex the request is sent to.
RMFBKUP	The application submitting the request to OPC Automation.
RMF	Target subsystem. This looks like a job to OPC, but is actually a subsystem.
005 and 010	Standard operation sequence numbers used by OPC.
STOP and START	Requested function. There are no parameters in this example.

Example of a Request Definition

This section provides an example of how an application is defined.

The application name is MAINT. This application consists of three operations:

- Stop RMF on the target system
- Schedule a batch job
- Restart RMF on successful completion of the batch job

Figure 14 shows the initial panel in the application creation process.

```
--------------------------- CREATING AN APPLICATION ---------------------------
 Command ===> oper

 Enter/Change data below:
 Enter the RUN command above to select run cycles or enter the OPER command
 to select operations.

 Application id         : MAINT
 Valid from - to        : 95/04/17 - 99/12/31

 APPLICATION TEXT   ===> RMF maintenance_________
                                  Descriptive text of application
 Owner:
  ID                ===> NSC02___________
  TEXT              ===> ____________________________
                                  Descriptive text of application owner
 PRIORITY           ===> 5           A digit 1 to 9 , 1=low, 8=high, 9=urgent
 VALID FROM         ===> 95/04/17    Date in the format YY/MM/DD
 STATUS             ===> A           A - Active, P - Pending
 AUTHORITY GROUP ID ===> ________    Authorization group ID
 CALENDAR ID        ===> ________________
                                  For calculation of work and free day
```

Figure 14. RMF Maintenance Application Primary Panel

In Figure 14, certain fields, such as calendar ID, are not used. However, OPC Automation does not preclude the use of normal application and operation functions.

Selecting OPER as a primary command allows the entry of the individual operations for this application.

In Figure 15, three operations are defined. The first and third send requests to OPC Automation in the NetView associated with the NV00 workstation. The second is a batch job named RMFMAINT that performs the batch maintenance tasks.

```
--------------------------------- OPERATIONS ---------------------  ROW 1 OF 2
 Command ===> text                                              Scroll ===> PAGE

 Enter/Change data in the rows, and/or enter any of the following
 row commands:
 I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
 S - Select operation details
 Enter the TEXT command above to include operation text in this list, or,
 enter the GRAPH command to view the list graphically.

 Application            : MAINT            RMF maintenance

 Row  Oper      Duration Job name  Internal predecessors           More preds
 cmd  ws   no.  HH.MM                                             -Int-Ext-
 '''' NV00 005   0.01    RMF_____  ___ ___ ___ ___ ___ ___ ___ ___   0   1
 '''' CPU1 010   0.10    RMFMAINT  005 ___ ___ ___ ___ ___ ___ ___   0   0
 '''' NV00 015   0.01    RMF_____  010 ___ ___ ___ ___ ___ ___ ___   0   0
 ****************************** BOTTOM OF DATA ********************************
```

Figure 15. Operations in the MAINT Application

Selecting TEXT as a primary command allows entry of the operator text. OPC Automation uses the operator text field to contain the request and optional parameters for operations with the workstation defined for OPC Automation. OPC Automation then passes up to two additional parameters from the operations text in OPC. Figure 16 shows the resulting operations text detail panel.

```
--------------------------------- OPERATIONS ---------------------  ROW 1 OF 2
 Command ===>                                                   Scroll ===> PAGE

 Enter/Change data in the rows, and/or enter any of the following
 row commands:
 I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
 S - Select operation details
 Enter the PRED command above to include predecessors in this list, or
 enter the GRAPH command to view the list graphically.

 Application            : MAINT            RMF maintenance

 Row  Oper      Duration Job name  Operation text
 cmd  ws   no.  HH.MM
 '''' NV00 005   0.01    RMF_____  STOP____________________
 '''' CPU1 010   0.11    RMFMAINT  ____________________
 '''' NV00 015   0.01    RMF_____  START___________________
 ****************************** BOTTOM OF DATA ********************************
```

Figure 16. Operations Text Detail Panel

In the applications, OPC Automation defines OPC requests in a generic manner. Figure 16 shows the MAINT application with the first and last operations defined for the NetView workstation NV00. The requests that are forwarded to the NetView workstation NV00 are STOP and START. These requests are expanded by definitions in the control file into specific SA OS/390 commands.

Handling Time Dependencies

If you require a time dependency, do not place the time consideration on the operation defined by NV*nn*. First define a dummy nonreporting workstation to implement time dependencies. Then define the time dependency on the dummy workstation. Define any dependencies other than time on the dummy workstation, which precedes the NV*nn* workstation.

Once you satisfy all dependencies, including the time dependency, the dummy workstation completes immediately and the NV*nn* workstation is driven. As an example, redefine the MAINT application shown in Figure 14 on page 23, and Figure 15 on page 24 and Figure 16 on page 24, with a timer dummy workstation (TIMR) as the first operation of the application. The panel in Figure 17 shows this new definition.

```
--------------------------------- OPERATIONS ---------------------  ROW 1 OF 2
 Command ===>                                                  Scroll ===> PAGE

 Enter/Change data in the rows, and/or enter any of the following
 row commands:
 I(nn) - Insert, R(nn),RR(nn) - Repeat, D(nn),DD - Delete
 S - Select operation details
 Enter the TEXT command above to include operation text in this list, or,
 enter the GRAPH command to view the list graphically.

 Application            : MAINT            RMF maintenance

 Row  Oper      Duration Job name  Internal predecessors           More preds
 cmd  ws   no.  HH.MM                                             -Int-Ext-
 '''' TIMR 005   0.01    ________  ___ ___ ___ ___ ___ ___ ___ ___   0   1
 '''' NV00 010   0.01    RMF_____  005 ___ ___ ___ ___ ___ ___ ___   0   0
 '''' CPU1 015   0.10    RMFMAINT  010 ___ ___ ___ ___ ___ ___ ___   0   0
 '''' NV00 020   0.01    RMF_____  015 ___ ___ ___ ___ ___ ___ ___   0   0
 ****************************** BOTTOM OF DATA ********************************
```

Figure 17. Using Time as a Dependency

With this type of structure, OPC Automation can schedule the NV*nn* operation, rather than the timer-dependent dummy workstation, if the application needs scheduling on demand or restarting. This manually initiated procedure is independent of the time consideration, if appropriate.

Changes to the Status of the Operation

The operation with the NV*nn* workstation goes through several status changes as the request defined in the operator text is processed. The initial trigger is one of three status changes. When the operation moves to the A (arrival), R (ready), or * (ready with nonreporting predecessor) status, the OPC Automation function in the EQQUX007/DRKUX007 exit is triggered. The exit then examines the request. If the request is valid, the exit transfers it to the target NetView.

If any definition problems are determined, OPC Automation updates the status to E with an error code of U*xxx*. See “Codes Posted to OPC by OPC Automation” on page 89. OPC Automation takes no further action. The user is then responsible for correcting the error and restarting the application at the failed operation.

If OPC Automation encounters a connectivity problem, it marks the operation with a status of E (error) and an error code of S*xxx*. If this happens, OPC Automation automatically restarts the operation once the connectivity problem is resolved. However, if the operation status is changed manually, the automatic restart is suppressed.

After OPC Automation resolves the request and verifies it, the status is updated to S (started) before OPC Automation submits it. Once the request is submitted and action is requested, OPC Automation updates the status to C (completed).

If the desired result did not occur within the time period specified, the operation ends with an E status and a U*xxx* code, indicating that user intervention is required.

Extending the Daily Plan

OPC Automation does not call EQQUX007/DRKUX007 for time-delay operations added at daily planning. To provide time-delay operations added at daily planning, you need to define an operation on a dummy workstation as a predecessor to the NV*nn* workstation. The operation on that workstation completes immediately after the daily plan is extended, and the operation on the NetView workstation is READY when all of its other dependencies are satisfied. See “Handling Time Dependencies” on page 25 for more information and an example of this technique.

Defining an operation on a dummy workstation is required because the operation-status-change exit is called whenever an operation in the current plan changes status. That exit is also called when a new operation has been added to the current plan by a function other than daily planning jobs, for example, by PIF or by the MCP dialog. The exit is called when the operation is added either to an existing occurrence or as a result of a new occurrence being added to the current plan.

Sending a Request to Optional Installation-Provided Functions

OPC Automation allows installation-specific extensions through optional user-provided modules. Two types of functions are supported:

- Issuing a nonSA OS/390 command or request
- Sending a request through to OPC Automation to an installation extension

Because these user-provided modules are unique to your installation, you need to obtain information from your systems programmer/analyst on the use and syntax of these functions.

Possible Uses of OPC Automation

In data centers, certain groups assume responsibility for the daily operation of the systems. Frequently, these groups are split into these two areas that perform the following tasks:

- Controlling online systems
- Processing all batch work

User requests for hours of service form the basis for online planning decisions. The time available to process the jobs required for online systems for the next day, as well as requests for other batch work, determines batch processing.

A system using OPC executes the current plan (CP), which contains the information for batch processing. A help desk, hotline, or service-contact point merge user-change requests into the overall schedule. While processing control executes batch processing, operations or master terminal operators control the online systems, thus adding to the confusion. Changes to online availability are frequently manual in nature. For example, instructions to change online availability often consist of slips of paper or phone calls to the operator.

OPC Automation allows changes which influence both batch and online systems through simple OPC dialogs. Because OPC manages both batch and online systems, these changes are needed only in one place. Because the processes are automated with SA OS/390 and OPC, no interoperator communications are required. In fact, in a highly automated environment, no operator intervention or awareness of these user-requested changes is necessary.

OPC Automation can automate some of the more complex operator procedures and thereby provide several new functions. The following topics give some examples and scenarios which demonstrate these functions.

Changing Online Hours of Availability

Several possible methods exist for changing the hours of availability of online services. To illustrate these methods, consider the example of a service such as IMS. Assume that the scheduled hours of availability for the IMS online service are 7 a.m. to 6 p.m. In NetView, under control of the current plan, OPC Automation performs the timed start and stop events.

- The help desk gets a request from a user group to extend the IMS hours of availability, for today only, from the original plan of 6 p.m. to 8 p.m. (extended service period).
- The help desk ensures that this extended service is acceptable within the service level agreement for this user group.
- The help desk now makes a change to the OPC current plan to reflect this extended IMS period.
- When the revised scheduled time is reached, now two hours later than usual, OPC executes the operation, requesting that OPC Automation stop IMS.
- OPC Automation requests that the SA OS/390 application stop IMS.
- Once the SA OS/390 application successfully stops IMS, OPC Automation returns an operation-ended status to OPC, fulfilling OPC's dependencies on the online IMS.
- Jobs dependent on the termination of IMS are now released for execution.

The above scenario requires no intervention from the IMS/MTO or system console operators. No restructuring of the batch processing is necessary if the request is within planned service bounds.

Cycling Individual Online Databases

OPC Automation allows OPC to interact not only with the SA OS/390 functions, but also with the MVS and MTO consoles, which enables the scheduling of interrelated sequences of events. For example, it is possible to cycle individual databases rather than the complete online system. Figure 18 shows how this scheduling results in minimum disruptions to online applications.

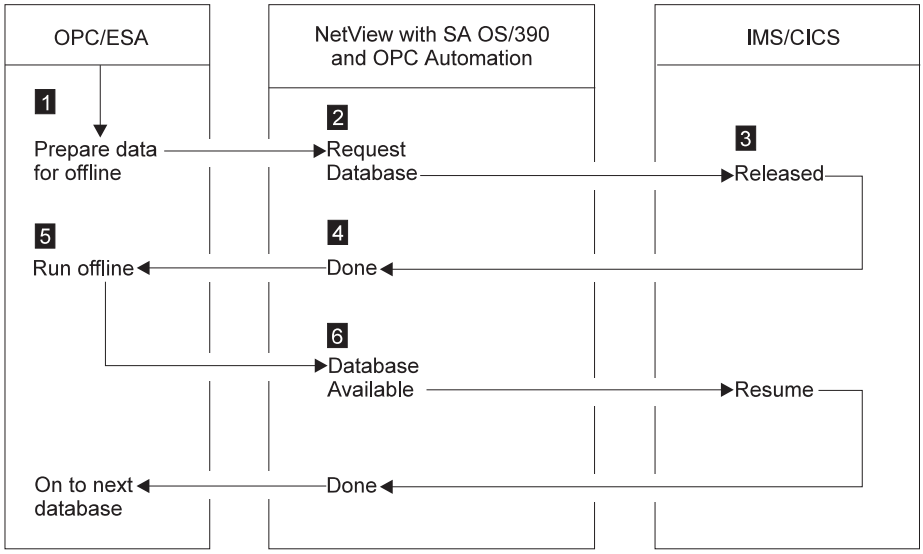

Figure 18. Example of Cycling Individual Online Databases

In Figure 18, the online databases are structured so that you can vary specific ones offline, without an impact to the system, as in the case of databases structured on a geographic or application basis. This process flows as follows:

1. Based on the current plan, OPC begins the READY TO START DATABASE UPDATE job.
2. A request is sent to OPC Automation to issue the command required to vary the subject database offline to allow for batch processing.
3. The request is issued through the MTO interface.
4. OPC Automation ensures that the database is offline.
5. OPC Automation posts the operation as completed in OPC.
6. With the operation completed, OPC dependency starts the batch processing for this database.
7. When the batch process is completed, OPC once again triggers OPC Automation, and the proper MTO command is issued to vary the database online to IMS.

When individual databases accomplish this type of process, the database/data communications (DB/DC) system is always up, and certain small portions of the data is unavailable for short periods. In some cases, you can restructure the databases to further shorten periods of data unavailability.

OPC Automation does not directly support the preceding example, which requires some user-written modules. (See “Interaction with AOC CICS Automation” and “Interaction with AOC IMS Automation” in the *AOC/MVS OPC Automation Programmer’s Reference and Installation Guide* for some examples of this type of user-written module. OPC Automation transports the request to the appropriate system and prepares the information for the user code. OPC Automation then returns the resulting status to the operation in OPC. OPC Automation also ensures that the actions requested are serialized with other requests to that specific target subsystem and that the status of the subsystem is such that it can accept the requests.

Scheduling Time for Testing

Another example is an automated mechanism that prepares a logically partitioned mode (LPAR) on a process resource/system management (PR/SM) complex for testing periods.

In this example, a system programmer or application developer makes a request through the help desk for testing. The help desk checks that the resources for the test period are available and invokes a prepared OPC-controlled application, updating the information required to set up the time and duration of the test. No other action is needed.

At the proper time, OPC begins execution. It sends the requests to the target system control facility (TSCF)[1] application to set up the LPAR for the test period, and to IML and IPL the PR/SM partition. If the requestor of the test period prepares the test system, so it is ready and waiting at the start of the test period, there is no waiting for an operator to set up the test environment or to structure the system as required by the testing.

Distributing and Updating Data Across Multiple Systems

As centralization of operations and support progresses, preparing data at a central site and then distributing it to other systems becomes necessary. Controlled execution of batch utilities is often required to update the target systems.

Installation of system maintenance provides an example of this type of distribution. Program temporary fixes (PTFs) are installed and tested at a central site. The PTFs are then shipped to target systems and applied with a system modification program (SMP or SMP/E). Frequently, a system programmer performs this by logging on to the target system and executing the job streams manually.

Another example is the creation of office system files on a central system, such as electronic telephone directories. These files are then distributed to the target systems.

OPC can control network job entry (NJE) jobs for the distribution of data, and thus controls the execution of the jobs on the target systems, to apply the data, using dependency control, if required.

---

[1] TSCF is a product primarily used to remotely IPL target 3090 processors from a focal point system. For more information, see your IBM representative.

OPC Automation extends this OPC capability and allows necessary cycling of the target system application once the maintenance is applied successfully. You can schedule this in such a way as to minimize any impact on the end user community. The following is a typical scenario:

1. A PTF is installed, tested, and found acceptable. This PTF is then applied to all copies of TSO in a multisystem environment.
2. The application is defined to OPC. In most cases, the application is simply updated since it is already defined.
3. OPC presents the batch jobs that control the SMP/E process to the systems programmer for modifications, if required.
4. OPC schedules the transmission of the jobs to the target systems using NJE. The scheduling can use a time when network traffic is low.
5. Once the jobs are in the target system, OPC dependency control is used to schedule the SMP/E job execution.
6. OPC ensures that the SMP/E jobs run correctly. If OPC encounters problems, the OPC application provides backout procedures.
7. After installing the PTFs, OPC selects the appropriate time to issue a request to OPC Automation to restart TSO.
8. Since OPC fully controls the process for this PTF update, you can inquire at any time to see the progress of the operations. If errors or problems occur, OPC Automation informs the SA OS/390 notification operator.

Complex Application Recovery

As computer applications become more critical to the daily operation of your enterprise, disaster recovery takes on an added significance. Usually, installations have the necessary equipment and facilities for disaster recovery, but the operational processes are so complicated that the chance of a successful backup in a short period, lasting from many minutes to no more than a few hours, is highly unlikely.

OPC Automation allows full or partial automation of this type of activity between systems and sites. In some cases, changing the NV*nn*-to-NetView domain ID relationship is adequate to transfer the control of the work load to a different system. However, the change may require some manual intervention for synchronization. Chapter 4, “Resynchronization and Recovery Considerations” on page 69 discusses several scenarios and the process of synchronization.

Although not all steps are required each time, most recoveries consist of three major operational steps that are executed sequentially and provide the following functions:

Step 1: Preparing the recovery system
This may require stopping some or all the applications on the recovery systems, unloading data from disk-storage devices to tape, and reconfiguring the recovery system.

Step 2: Starting the critical applications on the recovery system.
This can include the following:

- Loading databases and applications from tape-to-disk devices
- Starting the recovery system
- Updating data from checkpoint data, logs, or other sources
- Starting critical applications.

Step 3: Returning to the original production system
This is a reversal of the recovery process. These procedures are as complex as the original recovery process, but are scheduled and do not have the urgency of the original recovery.

In the following example, a series of applications need starting on a system after the failure of the original system or possibly even the site. Assume that the installation has prepared properly for this type of problem. This implies tested procedures, current levels of the affected applications and operating environment, and data at the backup site. To simplify this example, assume that the database at the recovery site is adequate for a contingency recovery situation.

- Prior to the need, a series of interdependent recovery applications are defined to OPC, but not scheduled.
- The decision to recover the critical applications at the backup site is made. The scheduler uses normal OPC panels to modify the current plan to schedule the first backup application.
- Before recovery, several factors, which can result in modifications to procedures and JCL, need considering. These modifications are then presented to operators at manual workstations with instructions in the operator instruction files of OPC. They are also presented to systems programmers at JCL workstations.
- The work load on the recovery system is stopped by scheduling a request to SA OS/390 to stop all subsystems other than JES.
- Once the subsystems are stopped, a series of jobs are scheduled to transfer data from disk-to-tape to accommodate the requirements of the critical applications that are recovered.
- Depending on the situation, the same system is reused or restructured, and then followed by an IML and IPL of the recovery system. If this is the case, the focal point implementation option of SA OS/390 is used to partially or completely automate this phase of the recovery. Regardless of the specifics, the result is an operating system platform ready to accept the recovery environment.
- OPC schedules a series of JES jobs that restore the databases from backup.

- OPC triggers NetView to issue the appropriate commands to start the subsystem.
- In some cases, NetView requires access to MTO functions to issue specific procedures before the DB/DC system can resume transaction processing. If that is the case, user-provided modules are required to fully automate the recovery.

At this point, the recovery is completed. Normal operating procedures should apply to the environment. Because a recovery situation creates an environment where resources are scarce, the actual applications that are offered are frequently a subset of the normal applications. To accommodate this environment, OPC and OPC Automation may need to change the scheduling of some of the applications controlled by OPC.

After recovery occurs and you resolve the problems which forced the original backup, the applications should be moved back to the original system. The scenario for this move is similar to the one above except that this move is planned instead of forced. This allows you to move specific applications one at a time, as opposed to the all-at-once scenario that a critical situation requires. The fact that some of the applications are moved to an already working system makes the takeback more complex than the original recovery.

Give special consideration to any synchronization procedure in an OPC Automation environment. For more information on the synchronization process, see Chapter 4, “Resynchronization and Recovery Considerations” on page 69.

Chapter 3. Operator's Guide

This chapter describes OPC Automation typically carried out by the operations staff.

Note: In cases where the operations staff has OPC access, review Chapter 2, "Scheduler's Guide" on page 17.

This chapter contains two main sections:

- "How the Operator Uses OPC Automation" describes the two most common tasks of an operator.
- "OPC Automation Commands" on page 41 explains all of OPC Automation commands.

How the Operator Uses OPC Automation

The OPC Automation operator performs two basic tasks:

- Monitoring the health of the complex using the Status Display Facility
- Responding to alerts or other error conditions that are not resolved by OPC, SA OS/390, or OPC Automation.

OPC Automation and SA OS/390 work by capturing messages and taking actions based on message contents. Some of those actions are coded by the programmer when OPC Automation and SA OS/390 are installed and implemented.

Messages fall into the following two categories:

- Problems encountered by OPC
- OPC-controlled applications

To ensure that an operator is made aware of the event, OPC Automation uses the Status Display Facility to show the status of the systems, subsystems, and applications.

Monitoring the Health of the System

OPC Automation provides enhancements to the SA OS/390 Status Display Facility so that production-control environment information is shown with operational environment information. This gives you a comprehensive view of your installation.

These enhancements include the following:

- Tape mounts
- Batch job monitoring
- TSO user monitoring
- Critical message alerting
- OPC ended-in-error alerting

To use the Status Display Facility, type **SDF** and then press ENTER. See “Status Display Facility Panel Array” on page 43 and “SDF—Status Display Facility Panel” on page 44 for details on using the Status Display Facility.

Notes:

1. The panel examples shown in this section are the default panels provided with OPC Automation. Your system programmer may have changed these panels during implementation. This is an option that OPC Automation provides so that you can tailor your panels to your specific needs.
2. The Status Display Facility requires that the terminal from which the command is entered support 3x79 terminal extended attributes.
3. The definitions of color and highlighting depend on how SA OS/390 and OPC Automation are implemented at your location.
4. To fully understand the Status Display Facility, refer to the appropriate section in the SA OS/390 operator manuals.

Problems in an OPC-Defined Application

If an application under OPC control ends in error, OPC issues a message which NetView intercepts. OPC Automation uses the message to update the Status Display Facility panel. The subsystem name turns the appropriate color, probably red, which is the default.

Note: When the problem is in OPC (the Controller, the Tracker, or both), the subsystem fields change color.

Your systems programmer will have customized the display facility to look something like one of the two following figures, either the SY1 System Status—Single-System Summary panel or the EVJD0001 OPC Automation Monitor panel. (For more details on what the fields in these panels mean, see either “SY1 System Status—Single-System Summary Panel” on page 45 or “EVJD0001 OPC Automation Monitor Panel” on page 46.) In either case we will be looking at OPC applications in error. For this scenario, the OPCERR field or the “Applications in Error” field has changed to an alert-condition color. This means that an error has occurred in an OPC-controlled subsystem.

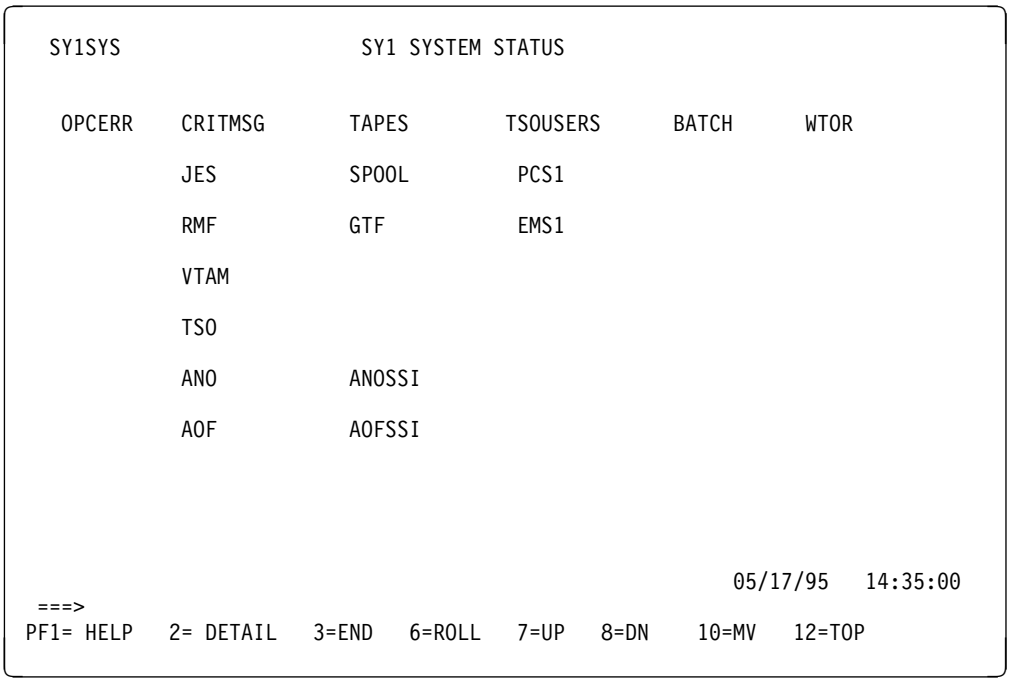

Figure 19. SY1 System Status—Single-System Summary Panel

If your panel looks like Figure 19, then place the cursor under OPCERR. After you press PF8, OPC Automation displays the OPC Automation Applications in Error panel, as shown in Figure 21 on page 36.

```
EVJD0001            OPC AUTOMATION MONITOR PANEL

         SY1 Critical Messages

         Tape Drives and Mounts

         Applications in Error

         Batch Jobs

         Tso Users


                                                              05/17/95 14:36:15
===>
F1=HELP 2=DETAIL 3=END 4=HWD  6=ROLL 7=UP 8=DN                      12=TOP
```

Figure 20. OPC Automation Monitor Panel

If your panel looks like Figure 20, then place the cursor under “Applications in Error.” After you press PF8, OPC Automation displays the OPC Automation Applications in Error panel.

The OPC Automation Applications in Error panel, as shown in Figure 21, lists the application or applications that initiated the error. See “OPC Automation: Applications in Error Panel” on page 47 for more discussion of this panel.

```
SY1OPCA         OPC Automation: Applications in Error

   RMFDLY_RMFMAINT
   APPL2_25

                                                          05/17/95   14:36:30
 ===>
PF1= HELP  2= DETAIL  3=END         6=ROLL  7=UP  8=DN    10=LF  11=RT  12=TOP
```

Figure 21. OPC Automation: Applications in Error Panel

Now that you know which application has failed, you are ready to determine your actions. Type **OPCACMD** on the command line of any NetView panel. After you press ENTER, OPC Automation displays the AOC OPC: Display or Modify OPC Data panel (EVJKAC01), as shown in Figure 22.

```
EVJKAC01         AOC OPC: Display or Modify OPC Data
                                                          Date: 05/17/95
                                                          Time: 14:36:36

Specify search criteria and press ENTER

  Subsystem    :  OPCC                From ACF-file

  Application  :  rmf*_____________   Can be generic
  Opno         :  ____                Numeric
  Jobname      :  ________            Can be generic
  Wsname       :  ____                Can be generic
  Group        :  ________            Can be generic
  Owner        :  ________            Can be generic
  Priority     :  _                   1-9 (1=low, 9=high)
  Errcode      :  ____                Can be generic
  Status       :  _                   A/W/S/R/C/I/E/U



Action===>
F1= Help     F2= End      F3= Return                          F6=Roll
```

Figure 22. AOC OPC: Display or Modify OPC Data Panel

In cases where the reported problem concerns an OPC-defined application, you have the option of logging on to OPC or using the OPCACMD command to access panels that allow you to browse or change the application information from OPC Automation. Using OPCACMD has the advantage of allowing you to remain in NetView and to jump between the OPC information and other NetView components.

You can enter the search criteria for the application directly onto this panel. Since RMFDLY_RMFMAINT is the failing application and job, type **RMFDLY** in the application field, as shown in Figure 23.

```
Application  :  RMFDLY_____________   Can be generic
```

Figure 23. Entering MAINT in the Application Field

Note: You can enter other parameters onto the OPCACMD panel, such as job name and operation number. As in this example, the application does not need to be in error to be accessible by the OPCACMD module.

After you press ENTER, OPC Automation displays the Display or Modify OPC Data panel (EVJKAC03), as shown in Figure 24.

```
EVJKAC03            Display or Modify OPC Data              Page:   1  of   1
                                                            Date: 05/17/95
CMD: C change  B browse                                     Time: 14:36:42
     H hold  R release  N no-op  U unno-op
CMD  Application      Jobname   Ws  Opno St   Inp. Arr   Description  H    N
---  ---------------- --------- ----  ---- -   ----------  -----------  -    -
 _   RMFDLY           RMF       NV11 0001 C    9505262230 STOP         N    N
 b   RMFDLY           RMFMAINT  CPU1 0002 E    9505262230              N    N
 _   RMFDLY           RMF       NV11 0003 W    9505262230 START        N    N
 _   RMFDLY           RMF       NV11 0001 A    9505302200 STOP         N    N
 _   RMFDLY           RMFMAINT  CPU1 0002 W    9505302200              N    N
 _   RMFDLY           RMF       NV11 0003 W    9505302200 START        N    N
 _   RMFDLY           RMF       NV11 0001 A    9506062200 STOP         N    N
 _   RMFDLY           RMFMAINT  CPU1 0002 W    9506062200              N    N
 _   RMFDLY           RMF       NV11 0003 W    9506062200 START        N    N
 _



Action===>
F1= Help     F2= End      F3= Return                 F5= Refresh   F6= Roll
```

Figure 24. Display or Modify OPC Data Panel

Figure 24 presents the operations in the requested application. The panel displays detailed information about a specific operation, enabling you to review the information or to make changes. The operation with the RMFMAINT batch job shows that the status is ended in error (E).

To obtain additional information on the type of error, type **B** in the CMD field, as shown in Figure 25.

```
 B   RMFDLY       RMFMAINT     CPU1  0002  E     95/05/17 14:36:42
```

Figure 25. Entering B (Browse) in the CMD Field

After you press ENTER, OPC Automation displays the OPC Occurrence Data panel (EVJKAC02).

```
EVJKAC02             OPC Occurrence Data
                                                          Date: 05/17/95
                                                          Time: 14:38:30

Subsystem    : OPCC               Deadline  : 95/05/26 23:30
Application  : RMFDLY             OPIA date :   /  /     OPIA time :   :
Opno         : 0002               Edur      : 0005       PS reqd   : 0001
Inp arrival  : 95/05/26 22:30     Job Class : P          R1 reqd   : 0000
Jobname      : RMFMAINT           Auto Sub  : Y          R2 reqd   : 0000
Wsname       : CPU1               AJR       : Y          A E C     : Y    Y/N
                                  Clate     : N          Timedep   : N
                                  Form no   : ____       Hi RC     : 0000
Priority     : 5                  Reroute   : N          Restart   : Y
Error Code   : SB37               DeadWTO   : N          Cat mgt   : N
Status       : E                Manual hold : N          NOP       : N
                                  Description : ________________________
                                  User data   : ________________


Action===>
F1= Help     F2= End      F3= Return                          F6=Roll
```

Figure 26. OPC Occurrence Data Panel

Note: From the panel in Figure 26, you can determine that the problem is a SB37 abend. Assume that the conditions causing this abend are fixed and that the recovery is to rerun the RMFMAINT job.

Press PF3 to return to the previous panel. This time type **C** in the CMD field, as shown Figure 27.

```
 C   RMFDLY       RMFMAINT     CPU1  0002  E     95/05/17 14:38:30
```

Figure 27. Entering C (Change) in the CMD Field

The resulting panel is similar to the one in Figure 24 on page 38 except that you can overwrite the fields with new information since this panel is in change mode rather than browse mode. Recovery consists of restarting the application at this operation. To do this, you overtype the C status field with an **R**, indicating a rerun request, and press ENTER. OPC Automation then examines the changes and sends them to OPC through the OPC API. The bottom left of the panel is updated with a message indicating that OPC has accepted the request or that there is an error.

Although the above scenario is a simple one, you can handle more complex situations without logging on to OPC to perform the actions. You can only use this interface to fix existing problems. You cannot use it to define new applications or workstations. If these functions are needed, you are required to log on to OPC directly, from TSO/ISPF.

Note: The OPCACMD command and this panel are described in detail in “OPCACMD—Interacting Dynamically with OPC” on page 56.

OPC Automation Commands

The following commands are used with OPC Automation:

Table 5. OPC Automation Commands

Command	Description
OPCA	Displays the OPC Automation Main Menu (EVJT0000) that lists the most commonly used commands and provides access to the tutorials. See “OPC Automation Main Menu and Tutorials” on page 42.
SDF	Accesses the Status Display Facility. This is actually an SA OS/390 command. However, sample Status Display Facility panels are supplied with OPC Automation. This includes the following changed or additional panels: • Status Display Facility panel • SY1 System Status—Single-System Summary panel • OPC Automation: Applications in Error panel • SY1 Critical Messages panel • SY1 System Status—Subsystem Messages panel • SY1 Tapes panel • OPC Automation: TSO Users Status panel • OPC Automation: Batch Jobs Executing on SY1 panel • OPC Automation Main Menu • OPC Automation: Display or Modify OPC Data panel (EVJKAC01) • OPC Automation: OPC Occurrence Data panel (EVJKAC02) • OPC Automation: Display or Modify OPC Data panel (EVJKAC03) See “SDF—Status Display Facility Panel” on page 44.
OPCACMD	Displays the OPC Automation: Display or Modify OPC data (EVJKAC01) panel that is used to interact dynamically with OPC. See “OPCACMD—Interacting Dynamically with OPC” on page 56.
OPCAQRY	Lists pending NetView operations. See “OPCAQRY—Display Status of Operations” on page 58.
OPCAPOST	Posts an operation in OPC from SA OS/390. See “OPCAPOST—Posting an OPC Operation from SA OS/390” on page 61.
SRSTAT	Determining status of OPC special resources. See “SRSTAT—Determining OPC Special Resource Status” on page 62.
DOWN	Terminates a subsystem. See “DOWN—Terminating a Subsystem” on page 63.
$RESTART	Terminates a subsystem and restart it. See “$RESTART—Terminating and Restarting a Subsystem” on page 64.
UP	Starting a subsystem. See “UP—Starting a Subsystem” on page 65.
EVJESPIN	Normally used only during initialization by SA OS/390. Can use this command manually to create a new status file record or resynchronize an existing one. The actions available with this command are INIT, RESET, SYNC, and CREATE. See “EVJESPIN—Initialization” on page 66.

OPC Automation Main Menu and Tutorials

Type **OPCA** on the command line. After you press ENTER, OPC Automation displays the AOC OPC Main Menu, as shown in Figure 28.

```
EVJT0000               A O C   O P C    M A I N    M E N U

   TYPE    COMMAND             DESCRIPTION                    TUTORIAL

                               AOC OPC Automation Feature         1

    P      SDF                 Display facility additions         2

    P      OPCAQRY             Display status of operations       3
    L      OPCAPOST            Manually post status to OPC        4
    P      OPCACMD             Display and modify OPC data        5
    L      SRSTAT              Update special resource status     6

    L      UP                  Start subsystem                    7
    L      DOWN                Terminate subsystem                8
    L      $RESTART            Recycle subsystem                  9

Note: Commands of type L (linemode) do not have an input panel
      and must be invoked with all required parameters specified.


Enter a Tutorial Number or Command ==>
          PF1= Help   PF2= End   PF3= Return
          PF6= Roll
```

Figure 28. AOC OPC Main Menu

To obtain information on using this panel, enter the tutorial number adjacent to the command.

Status Display Facility Panel Array

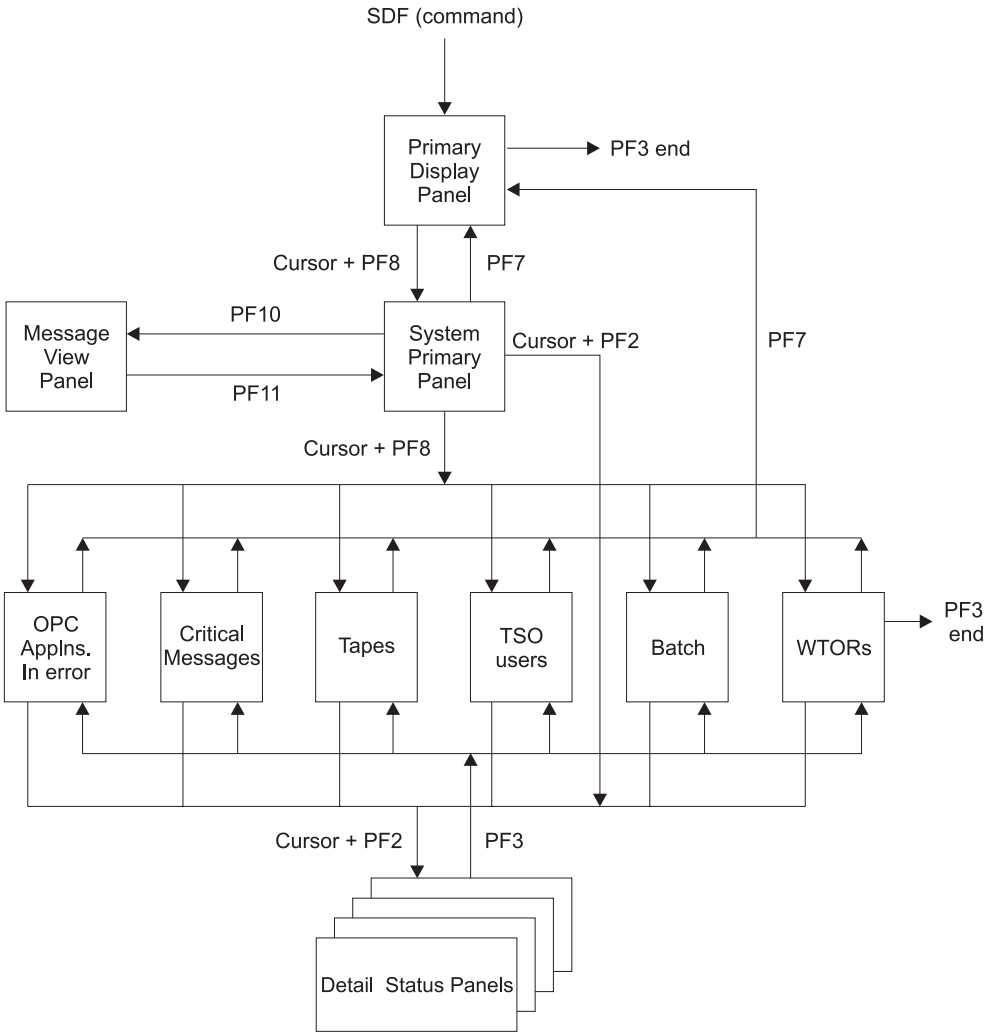

Figure 29. Enhanced Status Display Facility Panel Array

Figure 29 shows the Status Display Facility Panel Array, which provides an overview of all the installation's systems. This example shows two defined systems, SY1 and SY2. The color of the system shows its status. Status is propagated upwards from detail panels in a priority manner. In this way, the color represents the highest priority condition on that system.

SDF—Status Display Facility Panel

To use the Status Display Facility, type **SDF** and then press ENTER. Depending on how your systems programmer has customized SDF, you will see either a display similar to the AOC/MVS Support Systems—Status Display Facility panel in Figure 30, or else a display similar to the SY1 System Status—Single-System Summary panel in Figure 31 on page 45.

```
SYSTEM                AOC/MVS SUPPORT SYSTEMS

   System   Subsystems   WTORs     Gateways   Spool        MVS Comps  Features

   DALLAS                                                             I  C  O

   RALEIGH                                                               C

   CHICAGO                                                            I

   DENVER                                                                  O

   PHOENIX  FPIMSEA      IMS401E   AOF05I     JES                   I  C  O
```

Figure 30. AOC/MVS Support Systems—Status Display Facility Panel

Use the AOC/MVS Support Systems panel, shown in Figure 30, as follows:

- Tab to a system name and press PF8 to display the next level of detail.
- Press PF2 to display the detail screen with the highest level of priority for that system.
- Tab to the ‘O’ in the same row as the system and press PF8. You will go to the OPC Automation Monitor Panel, as seen in Figure 32 on page 46.

SY1 System Status—Single-System Summary Panel

If your systems programmer has customized SDF this way, then you will see OPC Automation categories displayed on the same panel as subsystem status, similar to Figure 31 below.

```
 SY1SYS                    SY1 SYSTEM STATUS

  OPCERR     CRITMSG       TAPES        TSOUSERS      BATCH      WTOR
             JES           SPOOL         PCS1
             RMF           GTF           EMS1
             VTAM
             TSO
             ANO           ANOSSI
             AOF           AOFSSI




                                                      05/17/95   14:35:00
 ===>
PF1= HELP   2= DETAIL   3=END   6=ROLL   7=UP   8=DN    10=MV   12=TOP
```

Figure 31. SY1 System Status—Single-System Summary Panel

The following fields and functions are added with OPC Automation:

OPCERR OPC operations ended-in-error alerts. Place the cursor on this field and press PF8 to display the screen described in “OPC Automation: Applications in Error Panel” on page 47.

CRITMSG Critical messages. Place the cursor on this field and press PF8 to display the screen described in “SY1 Critical Messages Panel” on page 48.

TAPES Tape mount requests and tape units varied online. Place the cursor on this field and press PF8 to display the screen described in “SY1 Tapes Panel” on page 51.

TSOUSERS TSO users logged on. Place the cursor on this field and press PF8 to display the screen described in “TSO Users Status Panel” on page 53.

BATCH Batch jobs executing. Place the cursor on this field and press PF8 to display the screen described in “Batch Display Panel” on page 55.

PF10—Message View Displays SY1 System Status—Subsystem Messages panel that lists subsystem messages for this system.

EVJD0001 OPC Automation Monitor Panel

If you press PF8 to get a “down” panel from the “O” field on the initial Status Display Facility panel, you will see panel EVJD0001 OPC Automation Monitor Panel. Figure 32 shows this panel.

```
EVJD0001          OPC AUTOMATION MONITOR PANEL

         SY1 Critical Messages

         Tape Drives and Mounts

         Applications in Error

         Batch Jobs

         Tso Users


                                                          05/17/95 14:35:30
===>
F1=HELP  2=DETAIL  3=END  6=ROLL  7=UP  8=DN                    12=TOP
```

Figure 32. OPC Automation Monitor Panel

The following fields and functions are added with OPC Automation:

SY1 Critical Messages Critical messages. Place the cursor on this field and press PF8 to display the screen described in “SY1 Critical Messages Panel” on page 48.

Tape Drives and Mounts Tape mount request and tape units varied online. Place the cursor on this field and press PF8 to display the screen described in “SY1 Tapes Panel” on page 51.

Applications in Error OPC operations ended-in-error alerts. Place the cursor on this field and press PF8 to display the screen described in “OPC Automation: Applications in Error Panel” on page 47.

Batch Jobs Batch jobs executing. Place the cursor on this field and press PF8 to display the screen described in “Batch Display Panel” on page 55.

TSO Users TSO users logged on. Place the cursor on this field and press PF8 to display the screen described in “TSO Users Status Panel” on page 53.

OPC Automation: Applications in Error Panel

If you select OPCERR from the SY1 System Status—Single-System Summary panel shown in Figure 31 on page 45, OPC Automation displays the OPC Automation: Applications in Error panel, as shown in Figure 33.

```
SY1OPCA         OPC Automation: Applications in Error


   RMFDLY_RMFMAINT

   APPL2_25

                                                          05/17/95   14:36:30
 ===>
PF1= HELP  2= DETAIL  3=END         6=ROLL  7=UP  8=DN    10=LF  11=RT  12=TOP
```

Figure 33. OPC Automation: Applications in Error Panel

This panel presents you with all the OPC applications that have ended in error, regardless of whether they have operations which use OPC Automation. If the operation that ended in error is a batch job, the name of the job is provided after the underscore. In the first example above, the operations ended in error is a job with a name of RMFMAINT in an application with the name of RMFDLY. In the second example, operation 25 of the application APPL2 has ended in error. Because a numeric name is displayed, this operation is not a batch job.

SY1 Critical Messages Panel

If you select CRITMSG from the SY1 System Status—Single-System Summary panel shown in Figure 31 on page 45, OPC Automation displays the SY1 Critical Messages panel, as shown in Figure 34.

Note: These message must have been selected by your systems programmer for display.

```
SY1MSGS               SY1  CRITICAL MESSAGES

 NAME     MESSAGE TEXT
 IEF450I  JOB CICS06CB - IEF450I CUCS06CB CICS06CB - ABEND=S22 U0000 REASON=0
 IEF450I  JOB VERA - IEF4501 VERA STEPISPF USERPROC - ABEND=S622 U000 REASON=0

                                                          05/17/95   14:37:00
 ===>

PF1= HELP  2= DETAIL  3=END         6=ROLL  7=UP          10=LF  11=RT  12=TOP
```

Figure 34. SY1 Critical Messages Panel

Press PF2 to list the critical messages in the system. OPC Automation displays the Critical Message—Detail Status Display, as shown in Figure 35 on page 49.

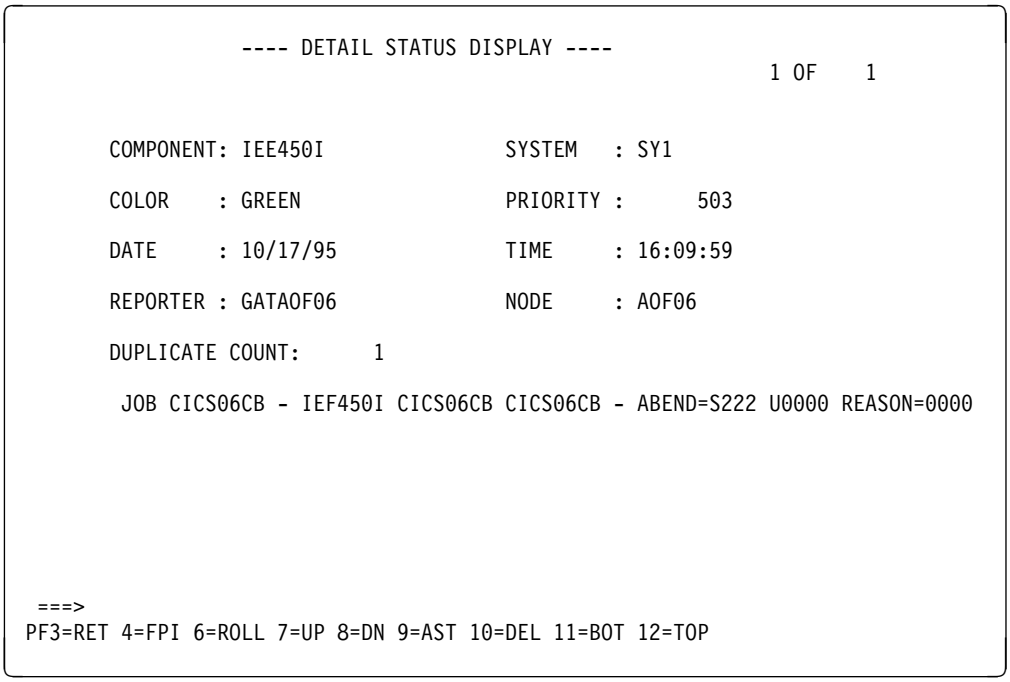

Figure 35. Critical Message—Detail Status Display

Your system administrator selects the messages on the Critical Message—Detail Status Display and determines whether they are critical. They remain in the system until the automation NetView is shut down, or you manually delete them using PF10.

SY1 Subsystem Status—Subsystem Messages Panel

If you press PF10 from the SY1 System Status—Single-System Summary panel shown in Figure 31 on page 45, OPC Automation displays the OPC Automation Subsystem Messages panel. Figure 36 shows this panel, which displays the current status message for each subsystem.

```
SY1SYS2              SY1  SYSTEM STATUS

 NAME     MESSAGE TEXT
 RMF      AOF571I 09:47 : RMF SUBSYSTEM STATUS FOR JOB RMF IS CTLDOWN
 OPCA     AOF571I 09:47 : OPCA SUBSYSTEM STATUS FOR JOB OPCS IS UP
 CICS06CA AOF571I 09:47 : CICS06CA SUBSYSTEM STATUS FOR JOB CICS06CA IS UP
 CICS06BA AOF571I 09:48 : CICS06BA SUBSYSTEM STATUS FOR JOB CICS06BA IS UP
 CICS06AA AOF571I 09:48 : CICS06AA SUBSYSTEM STATUS FOR JOB CICS06AA IS UP
 NVACCESS AOF571I 09:48 : NVACCESS SUBSYSTEM STATUS FOR JOB EMS IS UP
 TSO      AOF571I 09:49 : TSO SUBSYSTEM STATUS FOR JOB TSO IS UP
 ANOAPPL  AOF571I 09:49 : ANOAPPL SUBSYSTEM STATUS FOR JOB ANOAPPL IS UP
 VTAM     AOF571I 09:49 : VTAM SUBSYSTEM STATUS FOR JOB NET IS UP









                                                          05/17/95  14:38:00
 ===>
PF1= HELP  2= DETAIL  3=END         6=ROLL  7=UP          10=LF  11=RT  12=TOP
```

Figure 36. SY1 System Status—Subsystem Messages Panel

This panel is designed to give you an overview of your system. The color codes indicate the status of your system. For example, if all systems are up and running, the messages are all green. If a subsystem abends or is shut down, then that subsystem appears at the top of the screen in red or the color appropriate for its status.

SY1 Tapes Panel

If you select Tape Drives and Mounts from the OPC Automation Monitor panel shown in Figure 32 on page 46, then OPC Automation displays the SY1 Tapes panel, as shown in Figure 37.

```
SY1TAPE               SY1  TAPES
TAPE UNITS ONLINE
    350_3480    351_3400    352_3480    253_3480


TAPE MOUNTS
    350_KMX994        351_EXP723









                                                          05/17/95   14:38:30
 ===>
PF1= HELP  2= DETAIL  3=END         6=ROLL  7=UP          10=LF  11=RT  12=TOP
```

Figure 37. SY1 Tapes Panel

TAPE UNITS ONLINE lists the online tape units and their model numbers. TAPE MOUNTS displays outstanding tape mount requests. To use this panel, move the cursor to the specific tape mount and press PF2 to display detail information.

For example, if you request 350_KMX994, the Tape User—Detail Status Display is displayed as shown in Figure 38.

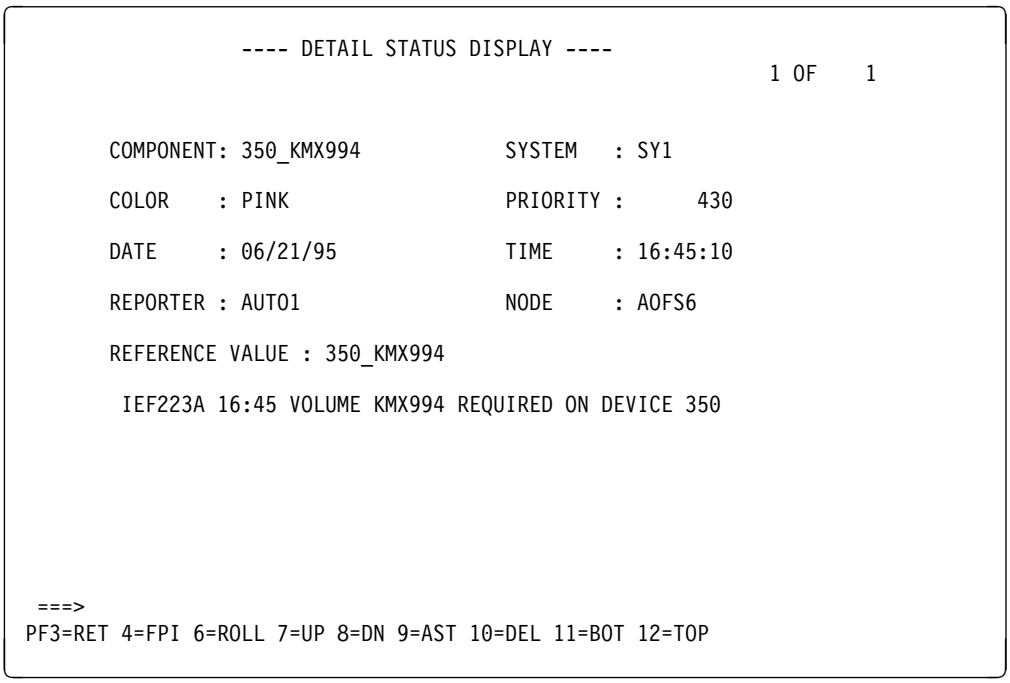

Figure 38. Tape User—Detail Status Display

Note: Mount requests that are not satisfied in two minutes turn to reverse video red.

TSO Users Status Panel

If you select TSO Users from the OPC Automation Monitor panel shown in Figure 32 on page 46, then OPC Automation displays the OPC Automation: TSO Users Status panel shown in Figure 39.

```
SY1TSOU            OPC Automation:   TSO USERS STATUS

       OPER1              NISTMPL

       NISTMPS            JILL

       DAVE               JAMES

       NISTMPK            SDL









                                                          05/17/95   14:39:00
 ===>
PF3=RET  4=FPI  6=ROLL  7=UP  8=DN  9=AST  10=DEL  11=BOT  12=TOP
```

Figure 39. OPC Automation: TSO Users Status Panel

The panel in Figure 39 shows all currently logged-on TSO users. As users log off, they are deleted from the screen. To use this panel, move the cursor to the specific user and press PF2 to display detail information.

For example, if you request NISTMPL, OPC Automation displays the TSO User—Detail Status Display, as shown in Figure 40.

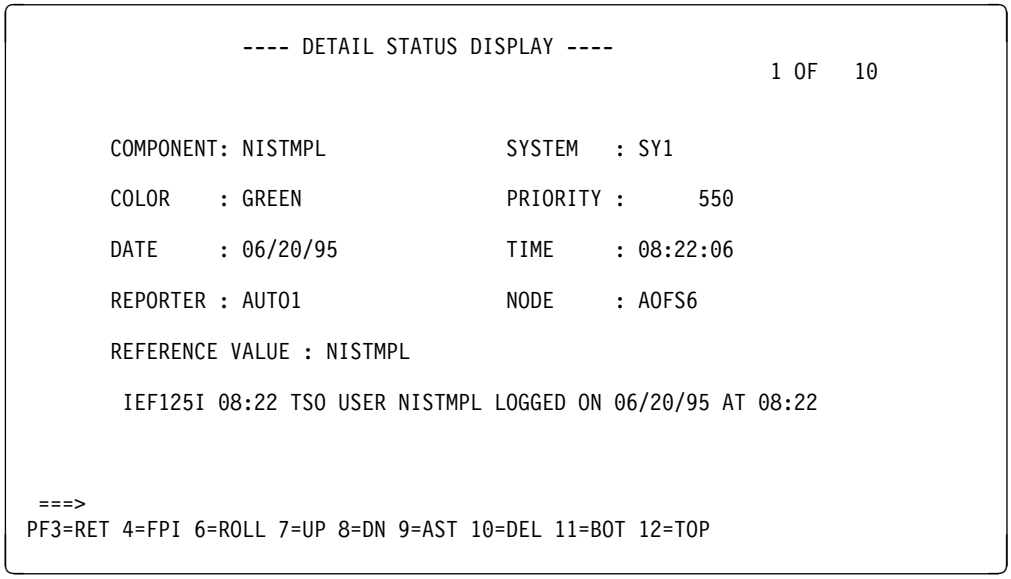

Figure 40. TSO User—Detail Status Display

Batch Display Panel

If you select Batch Jobs from the OPC Automation Monitor panel shown in Figure 32 on page 46, then OPC Automation displays the OPC Automation: Batch Jobs Executing on SY1 as shown in Figure 41.

```
 SY1BTCH         OPC Automation: Batch Jobs Executing on SY1

     PLI            SLRCOLL

     BLDFPI

     COMPRESS

     COLEY7

     CICSXA1

     APPLYPTF

     TAPEDMP

     EREP08

     NSC08B

     REMMAINT
                                                          05/17/95   14:40:00
 ===>
PF3=RET  4=FPI  6=ROLL  7=UP  8=DN  9=AST  10=DEL  11=BOT  12=TOP
OT  12=TO
```

Figure 41. OPC Automation: Batch Jobs Executing on SY1 Panel

The panel in Figure 41 shows all batch jobs active in the selected system. As jobs complete they are deleted from the screen. To use this panel, move the cursor to the specific job and press PF2 to display detail information. For example, requesting PLI displays the Batch Jobs—Detail Status Display shown in Figure 42.

```
                ---- DETAIL STATUS DISPLAY ----
                                                        1 OF   11

       COMPONENT: PLI                  SYSTEM   : SY1

       COLOR    : GREEN                PRIORITY :      550

       DATE     : 06/21/95             TIME     : 17:11:34

       REPORTER : AUTJES               NODE     : AOFS6

       REFERENCE VALUE : PLI

        IEF403I 17:11 : SUBSYSTEM ENTRY JOB PLI STARTED ON 06/21/95 AT 17:11.


 ===>
PF3=RET 4=FPI 6=ROLL 7=UP 8=DN 9=AST 10=DEL 11=BOT 12=TOP
```

Figure 42. Batch Jobs—Detail Status Display

OPCACMD—Interacting Dynamically with OPC

The primary use of the OPC SA OS/390 interface is to recover operations which could not proceed because of communications disruptions and to recover OPC-detected errors which require physical intervention. A NetView operator can use this interface for checking the status and progress of an application without logging on to OPC.

To access the OPCACMD panels, type the following on any NetView command line:

OPCACMD

After you press ENTER, OPC Automation displays the OPC Automation: Display or Modify OPC Data panel (EVJKAC01), as shown in Figure 43.

```
EVJKAC01      OPC Automation: Display or Modify OPC Data
                                                          Date: 05/17/95
                                                          Time: 14:41:30

Specify search criteria and press ENTER

  Subsystem    :  OPCC                From ACF-file

  Application  :  day*_____________   Can be generic
  Opno         :  ____                Numeric
  Jobname      :  ________            Can be generic
  Wsname       :  ____                Can be generic
  Group        :  ________            Can be generic
  Owner        :  ________            Can be generic
  Priority     :  _                   1-9 (1=low, 9=high)
  Errcode      :  ____                Can be generic
  Status       :  _                   A/W/S/R/C/I/E/U



Action===>
F1= Help     F2= End      F3= Return                          F6=Roll
```

Figure 43. Main OPC Command Dialog Panel

For a detailed example of using OPCACMD, see “Problems in an OPC-Defined Application” on page 35.

After you press ENTER, OPC Automation displays the Display or Modify OPC Data panel (EVJKAC03), as shown in Figure 44. This panel shows the operation summary.

```
EVJKAC03            Display or Modify OPC Data              Page:   1  of   1
                                                            Date: 05/17/95
CMD: C change  B browse                                     Time: 14:41:42
     H hold  R release  N no-op  U unno-op
CMD  Application      Jobname   Ws  Opno St   Inp. Arr   Description  H    N
---  ---------------- -------- ---- ---- -   ---------- -----------  -    -
 _   DAYRMF           RMF      NV11 0001 C    9505162200 STOP         N    N
 _   DAYRMF           RMFMAINT CPU1 0002 E    9505162200              N    N
 B   DAYRMF           RMF      NV11 0003 W    9505162200 START        N    N









Action===>
F1= Help     F2= End      F3= Return                 F5= Refresh   F6= Roll
```

Figure 44. Operation Summary Panel

After you type **B** (browse) in the CMD field, as shown in Figure 44, OPC Automation displays the OPC Occurrence Data panel (EVJKAC02), as shown in Figure 45. This panel shows operation details.

```
EVJKAC02             OPC Occurrence Data
                                                          Date: 05/17/95
                                                          Time: 14:42:00

Subsystem    : OPCC               Deadline  : 95/05/16 23:30
Application  : DAYRMF             OPIA date :   /  /     OPIA time :   :
Opno         : 0003               Edur      : 0005       PS reqd   : 0000
Inp arrival  : 95/05/16 22:30     Job Class : _          R1 reqd   : 0000
Jobname      : RMF                Auto Sub  : Y          R2 reqd   : 0000
Wsname       : NV11               AJR       : Y          A E C     : Y    Y/N
                                  Clate     : N          Timedep   : N
                                  Form no   : ____       Hi RC     : 0000
Priority     : 5                  Reroute   : N          Restart   : Y
Error Code   : ____               DeadWTO   : N          Cat mgt   : N
Status       : W                Manual hold : N          NOP       : N
                                  Description : START_____________________
                                  User data   : ________________




Action===>
F1= Help     F2= End      F3= Return                              F6=Roll
```

Figure 45. Operation Detail Panels

OPCAQRY—Display Status of Operations

The OPCAQRY command displays the status of OPC Automation operations.

To use this command, type the following on any NetView command line:

OPCAQRY

After you press ENTER, OPC Automation displays the OPC Automation Operation Status Display panel, as shown in Figure 46.

```
EVJKCGAA      OPC Automation Operation Status Display          Page:   1 of   1
                                                              Date: 05/17/95
Valid Actions: B Browse  D Delete  R Reset                    Time: 14:43:00

Act  Job     Application    Request     Date     Time     Status
 _   CX06AA                           03/24/95   08:44    No request
 _   DBSYS1  TEST2          STOP       03/29/95   14:16    Complete
 _   RMF     RMFMAINT       START      03/30/95   19:27    Complete
 _   SUBSYS1                          ********   *****    No request

Command ==>
F1= Help      F2= End      F3= Return               F5= Refresh  F6=Roll
```

Figure 46. OPC Automation Operation Status Display Panel

The panel in Figure 46 shows the status of requests from OPC Automation in NetView. It also provides a convenient place to delete unused records or to reset an operation in the event of problems.

The fields shown on the panel in Figure 46 on page 58 are defined as follows:

Act Action field, used for browsing, deleting, or resetting the status file record. See “Selecting Actions” for more discussion on this topic.

Job Name Job name from OPC, typically used to represent a subsystem.

Application OPC application requesting the operation.

Last requested action Action specified in the OPC operation description text.

Date Date the request was received.

Time Time the request was received.

Status Status of the request in NetView, either complete, incomplete, timeout, or no request. A status of timeout indicates that the operation is marked in error because it did not complete within the time limit set by the system programmer in the opca code entry. A status of incomplete indicates that the operation did not achieve the expected status set by the system programmer in the same entry. Complete and no request are considered normal statuses.

Selecting Actions

Select one of the following valid actions for any operation listed on the OPC Automation Operation Status Display panel, as shown in Figure 46 on page 58.

B Browse. Selecting the browse action allows the user to examine a record on OPC Automation Operation Status panel as shown in Figure 47 on page 60.

D Delete. This action deletes the record from OPC Automation status file.

A confirmation pop-up will appear, as follows:

```
..... DELETE CONFIRMATION .....
:                                :
:                                :
:        RECORD ID... TESTSYS    :
:                                :
:      _ ENTER 1 TO DELETE RECORD, :
:        PRESS F3 OR F12 TO KEEP RECORD. :
:................................:
```

R Reset. This clears flags in the record for reuse by performing an EVJESPIN CMD=RESET. This is useful for recovering operations that did not process normally. See “EVJESPIN—Initialization” on page 66 for more details.

After you type **B** (browse) in the Act field of the OPC Automation Operation Status Display panel shown in Figure 46 on page 58 and press ENTER, OPC Automation displays the OPC Automation Operation Status Detail panel, as shown in Figure 47.

```
EVJKCGA1      OPC Automation Operation Status Detail          Page:   1 of   1
                                                             Date: 05/17/95
 Status file record display for CX06AA                       Time: 14:44:00


 EHK170I OPCA RECORD DISPLAY FOR: CX06AA
 EHK171I ID= CX06AA          , TYPE= OPCA, OPID= RICK
 EHK172I LAST COMPLETED STATUS=  , LAST SEQUENCE NUMBER= 0000
 EHK173I TIMER FLAG=  , COMPLETION FLAG=
 EHK174I CURRENT SEQUENCE NUMBER= 0000, CHECK MODULE=
 EHK175I EXPECTED STATUS=         , TIMER INTERVAL=         ,TIMER ID=
 EHK176I ADNAME=                 , WSNAME=
 EHK177I OPNUM=   , JOBNAME=          , DATE= 05/17/95 ,TIME= 14:43
 EHK178I REQUEST=         , PARM1=         , PARM2=
 EHK002I END




Command ==>
F1= Help      F2= End      F3= Return               F5= Refresh  F6=Roll
```

Figure 47. OPC Automation Operation Status Detail Panel

OPCAPOST—Posting an OPC Operation from SA OS/390

This command is used by SA OS/390 to inform OPC of status changes. This is accomplished by the OPCAPOST command processor, which is normally used internally in OPC Automation. Although you can issue OPCAPOST as an operator command, operators should use OPCACMD, if possible. OPCACMD provides a full-screen interface to OPC and dynamically acknowledges the action, rather than OPCAPOST.

If you determine that you must use the OPCAPOST command, refer to the *AOC/MVS OPC Automation Programmer's Reference and Installation Guide*.

SRSTAT—Determining OPC Special Resource Status

This command lets you manipulate the status of OPC special resources. Status is returned via messages. The format of this command is:

SRSTAT *srname*,**SUBSYS=***subsys*,**AVAIL=Y|N**

srname
Special resource name — up to 44 characters.

subsys
Subsystem ID — 4 characters.

AVAIL=Y/N
Availability indicator.

Example:

```
SRSTAT EOD.CICSPRD1.TRANS,SUBSYS=OPCT,AVAIL=Y
```

In the above example, end-of-day transactions are required to complete before production work can begin. SRSTAT is executed when the transactions are complete. The special resource name EOD.CICSPRD1.TRANS is used to trigger OPC/ESA applications that are able to run when the transactions are finished. A number of applications are added to the current plan.

The variable used for *subsys*, OPCT, is the name of the tracker subsystem. Only in this command is the tracker subsystem name required.

DOWN—Terminating a Subsystem

This command terminates a subsystem or a subsystem group. The format of this command is:

DOWN *subsystem***{,ALL|ONLY|CHILDREN}**

subsystem The subsystem to be stopped.

{,ALL|ONLY|CHILDREN} The scope of the request.

ALL The subsystem and all children subsystems are stopped.

ONLY Only the specified subsystem is stopped (default).

CHILDREN Only the children subsystems are stopped.

Do not use this command on subsystems that are already running.

This example shows stopping RMF.

```
DOWN RMF
```

This example shows stopping only the subsystems with PARENT=VTAM.

```
DOWN VTAM,CHILDREN
```

$RESTART—Terminating and Restarting a Subsystem

This command terminates a subsystem or a subsystem group (the subsystem and its children) and then restarts it. This is sometimes called *recycling* a subsystem or *bouncing* it. **BOUNCE** is provided as a synonym for **$RESTART**; your system programmer may have added others.

The format of the command is:

| **$RESTART** *subsystem* **{,ALL|ONLY|CHILDREN}**

subsystem The subsystem to be restarted.

{,ALL|ONLY|CHILDREN} The scope of the request.

ALL The subsystem and all its children are restarted.

ONLY Only the specified subsystem is restarted (default).

CHILDREN Only the children subsystems are restarted.

Use this command only on subsystems that are already up.

This example shows restarting TSO:

```
$RESTART TSO
```

This example shows restarting TSO again with the **BOUNCE** synonym for this command:

```
BOUNCE TSO
```

This example shows restarting only the subsystems with PARENT=VTAM:

|
```
$RESTART VTAM,CHILDREN
```

UP—Starting a Subsystem

This command is used to start a subsystem or a subsystem group. The format of this command is:

UP *subsystem***{,ALL|ONLY|CHILDREN}**

subsystem The subsystem to be started.

{,ALL|ONLY|CHILDREN} The range of the request.

ALL The subsystem and all children subsystems are started.

ONLY Only the specified subsystem is started (default).

CHILDREN Only the children subsystems are started.

Do not use this command on subsystems that are already up.

This example shows starting RMF.

```
UP RMF
```

This example shows starting only the subsystems with PARENT=VTAM.

```
UP VTAM,CHILDREN
```

EVJESPIN—Initialization

OPC Automation uses this command during initialization when SA OS/390 is started. An operator can also use it to create a new OPC Automation status file record or to resynchronize an existing one.

Four command actions are available. The first, INIT, acts on all OPC Automation controlled subsystems, while the other three are available for the specified subsystem only.

EVJESPIN CMD=*action***,SUBSYSTEM=***subsystem*

CMD=*action* The command to execute where *action* is one of the following:

INIT Forces an equivalent action to that taken during normal initialization. Do not specify a subsystem parameter for this parameter. To properly understand the action of this command parameter and for a complete overview of the initialization process, refer to the initialization topic in the *AOC/MVS OPC Automation Programmer's Reference and Installation Guide*.

RESET Resets the timer and comp flags to a null value, and unlocks a specific subsystem after a user error is detected and corrected. By resetting the timer and comp flags, OPC Automation again accepts requests from OPC.

SYNC Checks the last completed status field in OPC Automation and ensures that the specified subsystem status agrees whether the last completed status is UP or CTLDOWN. OPC Automation resets the timer and comp flags to null. This marks the action as completed. Use SYNC when manual action is necessary to stop or start a subsystem.

CREATE Creates a new OPC Automation status file record from the control file definition for a specified subsystem. Use this function to refresh the control file dynamically with new OPC Automation definitions, when NetView is not recycled.

subsystem The subsystem initialized. Do not specify a subsystem with INIT.

EVJTRACE—Tracing

This command is used to trace the activity of OPC Automation.

EVJTRACE ON|OFF|STATUS

ON Tracing is started. Messages are sent to the NetView log on entry and exit of OPC Automation modules.

OFF Tracing is stopped.

STATUS The status of tracing is displayed.

Usage note:

Use this command for problem diagnosis, since it results in considerable output to the NetView log.

Chapter 4. Resynchronization and Recovery Considerations

OPC Automation combines the capabilities of two different subsystems, OPC and SA OS/390, which may reside over several systems. As a result, a failure or a scheduled interruption of services with one of the subsystems, processors, or telecommunications facilities may occur and prevent OPC Automation from processing operations. Further complications arise by shifting work load across multiple system images, either for scheduled work-load balancing or as part of a recovery situation.

In such a case, a loss of synchronization can occur between the OPC schedule and the OPC Automation components in NetView. When this happens, you may need a manual process to examine the OPC schedule, to ensure that OPC Automation in NetView is performing actions as required and, in some cases, to resynchronize OPC and OPC Automation.

During a loss of contact or a failure with either OPC or NetView, OPC Automation's facilities invoke and restore the environment as it was before the failure so that event scheduling can pick up where it left off. This results in a satisfactory resolution and manual resynchronization is not required. See "Automated Recovery Functions" on page 73 for additional information.

Generally, the longer the outage, the more likely that resynchronization is required. This depends on the number of scheduled events that are not processed. A long outage during the day may have a smaller synchronization impact than a shorter outage during a period when many online facilities are started or shut down.

Examples and Scenarios

This section describes possible scenarios for resynchronization and recovery.

Loss of Contact Between OPC and OPC Automation

Under most situations, once OPC Automation re-establishes connectivity, its automatic recovery schedules requests for execution that it could not execute prior to connectivity. In some situations you may desire to intervene manually, such as when the request is no longer valid. The following sections discuss several reasons for manual intervention.

Taking Action Manually on the Target System

The scheduler acts on the ended-in-error notification, requesting that an operator with access to the target system perform the required operation manually. The operation requested by OPC completes, and the scheduler manually updates it to a completed status, enabling OPC to continue processing.

You should issue EVJESPIN CMD=SYNC.

Taking Action Too Late

The remaining processing window is too small to allow an operation to occur. For example, an online system may require a certain amount of time to initialize. If this amount of time is close to the scheduled shutdown time, you probably should override the request and complete the operation manually.

You should issue EVJESPIN CMD=RESET.

Queuing Several Actions for a Specific Target Subsystem

Rather than not having enough time for a system initialization as in the previous example, the outage may have lasted long enough for a specific subsystem to receive several queued operations that frequently conflict. For example, an online task may have a start request with an ended-in-error status because of connectivity problems. This same task may also have a stop request already due for scheduling. If you allow automatic recovery for this application, the subsystem would start, but an immediate shutdown would follow.

Complete these operations manually. You should issue EVJESPIN CMD=RESET.

Backup on a Different Processor

If you perform a backup using a different processor, pay special attention to ensure that you properly restore the work load on the new system. Depending on the backup structure, you need to follow one of several different procedures discussed in the next sections.

Full Takeover onto a Standby System at the Same Site

This is the simplest type of backup. It becomes the same as a single-system recovery if the data is also available. OPC Automation uses the information in the status file to restore the various subsystems to the prebackup status.

Full Takeover onto an Standby System at a Different Site

If the status file is not available, restructure the environment manually. Examine pertinent applications that control this specific NV*nn* workstation. The last completed NV*nn* operation requires manual triggering. You can achieve this with the OPCACMD function, allowing the NetView operator direct access to the relevant applications. Although, at times, you may find it necessary to perform specific operations manually, in most cases, resetting an operation with EVJESPIN or restarting an application produces the desired effect.

Takeover onto a Working System

In most situations, the takeover is onto a working system and is restricted to certain critical applications. The previously discussed considerations as to whether the system is at the same site or at another site apply to this situation. Since not all applications are restarted on the backup system, several new considerations become important. Certain applications need cancelling before you can achieve a restoration of services. Some situations can result in duplications, such as with subsystems like TSO, which are frequently found on every MVS system. Although normally this is controlled by SA OS/390, OPC Automation can control this. Here, duplicate applications need cancelling for the backup period.

During the backup period, use one of these two methods to run OPC Automation:

- Add the new work load to the resident NetView by changing the NV*nn* workstation entry in the platform control file to point to the same NetView domain ID.
- Start another copy of NetView with the NetView domain ID used in the failing system.

The consideration of which method to use becomes important once you restore the original configuration.

With the single NetView solution method, you need to resolve the subsystems manually since their original identity is lost. With the extra NetView solution method, you can stop the subsystems controlled by the extra NetView by shutting it down. This simplifies the restoration process, requiring almost no manual intervention.

In both cases, restoring the environment follows similar procedures used to backup a host onto a standby backup system.

Long Term Outage

You must manually intervene when the outage duration is more than a single scheduling cycle. This type of recovery is confusing since many applications are shown as late in the OPC plan. Carefully review these applications since some of the them still need scheduling, while others need to be cancelled. For applications that need scheduling, certain operations involving the online portion need cancelling or holding. To ensure success, this type of recovery needs precise planning and monitoring. Otherwise, you can use the scenarios previously outlined.

Example Using Doubly-Defined NetView Domain IDs

The following example shows the control file entry for a 4-processor environment:

```
OPCA      DOMAINID,CODE=(NV00,,,NVTOR),
                   CODE=(NV01,,,XBAOF),
                   CODE=(NV02,,,AOFT5),
                   CODE=(NV06,,,AOFS6)
```

Here, NV00 maps to the NVTOR NetView domain ID, and the NV01 workstation maps to the XBAOF NetView domain.

Under normal circumstances, each NetView domain ID represents a processor with its MVS operating system and a unique NV*nn* general automatic reporting workstation. For situations such as testing, backup, or work load management, this relationship needs no maintenance.

In the previous example, both NV00 and NV06 OPC-defined workstations represent their own specific NetView domain ID, NVTOR, and AOFS6 respectively.

Assume that the system represented by NVTOR has failed, and you make the decision to shift the work load to the AOFS6 system. You can accomplish this by changing the relationship defined by the first CODE statement from a NetView domain ID of NVTOR to AOFS6. The resulting definitions in the control file to support this backup and recovery scenario follow:

```
OPCA      DOMAINID,CODE=(NV00,,,AOFS6),
                   CODE=(NV01,,,XBAOF),
                   CODE=(NV02,,,AOFT5),
                   CODE=(NV06,,,AOFS6)
```

If you accomplish this change without altering the OPC definitions, you must reload the SA OS/390 control file. The scheduler or operator needs to ensure that the OPC-defined applications that are running in the failed system are restarted on the backup system. Because the SA OS/390 status records are on the failed system, the scheduler manually recovers the failed environment. Once resynchronization completes, any new scheduled event originally intended for the NetView domain ID NVTOR automatically is scheduled for AOFS6. After you resolve the problem on the NVTOR system, perform the previous scenario in reverse order to restore the system to its original configuration.

When double definitions of this type are used, exercise caution to avoid creating conflicting requests for specific subsystems. For example, if RMF exists in the AOFS6 domain, OPC can then schedule a shutdown request on NV00 and a start request on NV06.

Automated Recovery Functions

Only a small portion of OPC Automation resides in the OPC address space in OPC user exits. These exits communicate to the rest of OPC Automation which resides in the NetView address space. A loss of contact results if a NetView address space becomes unavailable or if OPC Automation code in NetView is unavailable. Also, a communication failure can prevent a request from reaching its ultimate destination.

OPC Automation automated recovery determines which operations are affected by a specific loss of communications. It also determines when the connectivity and availability of a given target NetView is corrected and the NV*nn* operation is reset to the ready state. This redrives the EQQUX007/DRKUX007 exit, allowing it to re-create the original request.

OPC Actions in a Loss of Contact Situation

The EQQUX007/DRKUX007 exit or an intermediary NetView with an ended-in-error status and a return code of Sxxx reports a connectivity loss to OPC. OPC does not schedule any dependent operations and shows an error on the operations ended-in-error Status Display Facility panel. OPC takes no further action until the connectivity is restored and OPC Automation automatic recovery is invoked.

The operator or scheduler can manually override the ended-in-error status and permit continually or cancelling the application.

OPC Automation Actions in a Loss of Contact Situation

If loss of connectivity to NetView is detected in the OPC Automation portion of the EQQUX007/DRKUX007 exit (using the EQQUSINT/DRKUSINT function directly), then OPC Automation posts an ended-in-error status with a UNTV return code. OPC Automation uses this mechanism because the EQQUX007/DRKUX007 exit cannot directly modify operation status.

If the request is received in NetView, but OPC Automation cannot propagate the request to the appropriate target system, OPC Automation uses the OPCAPOST function to post the operation as ended-in-error with an S999 return code.

Appendix. Messages and Codes

Messages

EVJ000I *date time module* **ENTRY/EXIT***parms***/RC=***rc*

Explanation: When the trace option is activated, OPC Automation will write this data to the log for entry and exit tracing. With APAR OW23552 the date is expanded and is shown in the format `mm/dd/yyyy`, for example, 05/22/1997.

Operator Response: None.

EVJ001I *command***: STARTED, PARMS=***parameters*

Explanation: Command processor has been entered with the parameters shown. This is an informational message issued for audit purposes.

Inserts: Command processor name, parameters.

System Action: None.

Operator Response: None.

System Programmer Response: None.

EVJ002I *command***: COMPLETED SUCCESSFULLY, PARMS=***parms*

Explanation: Command processor has completed successfully. This is an informational message issued for audit purposes.

Inserts: Command processor name, parameters.

System Action: None.

Operator Response: None.

System Programmer Response: None.

EVJ003E *command* **UNABLE TO PROCESS PARMS=(***parameters***), REASON=***reason*

Explanation: OPC has initiated a request, but it cannot be processed due to the reason shown. This message will normally be preceded by a message giving specific details of the failure.

Inserts: Request parameters, reason text.

System Action: The request is cancelled.

Operator Response: Review log for other messages.

System Programmer Response: Resolve any system or definition problems, then reset the request.

EVJ004E *module***: BLANK ENTRY IN LINE** *nn* **OF STATUS FILE: PARMS=** *parms*

Explanation: The contents of a field in an OPC Automation status file record was found blank. An error message is issued with the parameters.

Inserts: Command processor name, line number, parameters.

System Action: Command processor ends and logs an error message.

Operator Response: Review error message in log.

Reset the record using OPCAQRY and retry the operation.

EVJ005E *module* **UNABLE TO OBTAIN** *parameter*

Explanation: The module indicated tried to obtain the value of a parameter from product globals, but the value was null or not as expected.

Inserts: Module name and failing parameter

System Action: The module ceases processing and returns a non-zero return code to its calling routine.

Operator Response: Review log for other messages and notify the system programmer.

Determine the source of the failure, typically an incorrect control file value or failure of the product initialization routines (beginning with EVJEAAIC) to run, and correct it. Call IBM Service if you require assistance.

EVJ006E *cmdname* **COMMAND FAILED FOR** *module***: RETURN CODE=***rc* **PARMS=***parms*

Explanation: A command called within this command processor has received an invalid return code. The command that failed and an error message is issued with the parameters of the original caller.

Inserts: Command processor name, failed command, error return code, parameters.

System Action: Command processor ends and logs an error message.

Operator Response: Review error message in log.

Correct any definition problems, and retry the failing operation.

EVJ007E *cmdname* **- SUBSYSTEM** *subsys* **IN INVALID STATUS** *stat* **FOR** *request*

Explanation: OPC Automation has attempted to execute the request for the indicated subsystem, but was unable to complete because the subsystem was in an invalid status. For example, trying to start a subsystem that is already up, or stop a subsystem that is already down.

Inserts: Command processor name, subsystem, subsystem status, OPC Automation request type.

System Action: The operation in OPC is posted in error with a U001 status.

Operator Response: Review NetView log for prior activity for the subsystem.

Check for manual intervention with the subsystem, such as starts and stops performed outside of automation control. Also, for shutdowns, specify RESTART=CTL so that subsystems will not be restarted by NetView initializations.

EVJ008E OPERATION ALREADY IN PROGRESS FOR *subsys***, PREVIOUS REQUEST=(***request parms***)**

Explanation: OPC Automation has attempted to execute the request for the indicated subsystem, but was unable to process because a previous request is still outstanding for the subsystem, or the previous request may not have completed successfully.

Inserts: Subsystem name, request parms.

System Action: The operation in OPC is posted in error with a U001 status.

Operator Response: Review NetView log for prior activity for the subsystem.

Check for previous errors in the log, or review OPC and workstation definitions. When problems are resolved, reset the OPC status file record using OPCAQRY and rerun the operation.

EVJ009E NO RESPONSE FROM DOMAIN *domainid* **WHILE SENDING REQUEST=(***request parms***)**

Explanation: OPC Automation has attempted to forward the request to the remote domain, but did not receive an acknowledgment from that system.

Inserts: Domain ID, request parameters.

System Action: The operation in OPC is posted in error with a U001 status.

Operator Response: Check status and connectivity of the system in question. If the problem can be resolved, rerun the failing operation.

Your gateway to the remote NetView is blocked, or the remote system may be extremely busy. If the former, you should resolve the gateway problem. If the latter, then you should schedule your operations on this remote NetView farther apart.

EVJ010I *variable data*

Explanation: This message is used internally to transfer data between systems.

Inserts: Date, time, message text.

System Action: None, this is a normal message.

Operator Response: None.

EVJ011I *date time OPC message text*

Explanation: While attempting to communicate with OPC, an operator initiated request to display or modify data has failed. The message text is the OPC message received. This message is normally preceded by another message, indicating the OPC Automation module that was attempting to request data from OPC.

Inserts: Date, time, message text.

System Action: None.

Operator Response: Review message in the appropriate OPC/ESA or OPC/A manual.

Correct problems such as incorrect definitions of OPC subsystem name.

EVJ020I CANNOT SCROLL FORWARD - LAST PAGE

Explanation: You have attempted to scroll beyond the end of the displayable data.

System Action: None.

Operator Response: None.

EVJ021I CANNOT SCROLL BACKWARD - FIRST PAGE.

Explanation: You have attempted to scroll backward from the first page of data.

System Action: None.

Operator Response: None.

EVJ022I INVALID OPTION *opt***, VALID OPTIONS ARE** *options*

Explanation: The indicated parameter is not valid.

System Action: None.

Operator Response: Correct the entry, review help screens for valid options.

EVJ023I *key* **IS NOT USED HERE**

Explanation: The indicated function key is not used in the panel you are viewing.

System Action: None.

Operator Response: Continue.

EVJ024I NO DATA FOUND

Explanation: No data was found that met the search criteria.

System Action: None.

Operator Response: Try again using different or generic search arguments.

EVJ025I INVALID VALUE *value* **FOR** *keyword*

Explanation: The parameter you have supplied is not valid.

System Action: None.

Operator Response: Try again using the correct parameters.

EVJ026I MISSING PARAMETER *parameter*

Explanation: The parameter is required, but was omitted.

System Action: None.

Operator Response: Try again using the correct parameters.

EVJ027I EVJESPIN PARAMETERS ARE CMD=INIT OR CMD=CREATE|SYNC|RESET,SUBSYSTEM=SUBSYS

Explanation: This message is issued as a reminder when invalid or missing parameters are detected by EVJESPIN.

System Action: None.

EVJ028I OPC Automation Tracing is *on|off*

Explanation: This message is issued when OPC Automation tracing is changed or inquired with the EVJTRACE command.

System Action: None.

Operator Response: None.

EVJ029I *command* **Encountered Unexpected event** *event_name*

Explanation: The indicated command was issued, but the response received was not as desired. For example, the command may have timed out, or the procedure may have been cancelled while waiting for a message.

System Action: None.

Operator Response: Investigate the cause of the failure.

EVJ030I Issued *command***, Received** *message text*

Explanation: The indicated command was issued, but the message response received was not as desired.

System Action: None.

Operator Response: Investigate the cause of the failure.

EVJ100I UX007001: OPC FEATURE EXIT IS ACTIVE

Explanation: OPC has now called the UX007001 exit.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ101I UX007001: DATA SENT=*data*

Explanation: This message logs the data sent to NetView across the NetView PPI in the system log.

Inserts: Data passed.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ102I UX007001: PPI REQUEST *nn* **FAILED, RC=***rc*

Explanation: UX007001 was unable to complete a NetView PPI request. See NetView PPI manual for more information on this request type and return code.

Inserts: Request type and return code.

System Action: OPC operation will be set to E status.

Operator Response: Review error message in log.

System Programmer Response: Review error message in log.

EVJ103I UX007001: GETMAIN FOR WORKAREA FAILED

Explanation: UX007001 was unable to do a GETMAIN for work area storage.

Inserts: None.

System Action: UX007001 will put the OPC operation in E status.

Operator Response: Report problem to your systems programmer.

System Programmer Response: Review NetView storage allocation and usage.

EVJ104I UX007001: OPC BAD OPERDESC: *data*

Explanation: UX007001 has found the contents of the operation description field to be invalid. Message EVJ105I will list the actual error data.

Inserts: OPC application name.

System Action: UX007001 will put the OPC operation in E status.

Operator Response: Inform OPC Scheduler of possible definition error.

System Programmer Response: None.

EVJ105I UX007001: OPC DATA *data*

Explanation: UX007001 has found this data to be invalid.

Inserts: Data.

System Action: UX007001 will put the OPC operation in E status.

Operator Response: Inform OPC Scheduler of possible definition error.

System Programmer Response: None.

EVJ106I *modname1*: *modname2* **STATUS UPDATE FAILED, RC=***rc*

Explanation: The module listed was unable to update operations status using DRK/EQQUSINT.

Inserts: Issuing module and status update module.

System Action: WTO error message is issued.

Operator Response: Inform OPC schedulers and system programmers.

System Programmer Response: Review system and OPC message logs.

EVJ107I UX007001: OPC FEATURE EXIT ENDED

Explanation: UX007001 has successfully ended.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ108I *module-name* **UNABLE TO LOAD MODULE CNMCNETV**

Explanation: This message is issued from the OPC status change exit, (EQQUX007/DRKUX007), when it cannot load the NetView PPI interface.

Inserts: Issuing module name.

System Action: Processing for this request halts. The operation will be posted in error with a return code of UNTV.

Operator Response: Notify the systems programmer.

System Programmer Response: This message most likely indicates an installation error. You must make the NetView PPI module accessible to OPC, either by STEPLIB or putting the NetView load library, CNMLINK, in LINKLST.

EVJ109I UX007001: Leaving AOC OPC exit, DRK/EQQUSINT successful

Explanation: UX007001 has successfully ended.

Inserts: Module name and successful command name.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ110I *modulename* **UNABLE TO LOAD MODULE** *target-module-name*

Explanation: This message is issued from an OPC command processor, which is unable to find or load an OPC interface module.

Inserts: Issuing module name and target module name.

System Action: Processing for this request halts.

Operator Response: Notify the systems programmer.

System Programmer Response: This message most likely indicates an installation error. You must make the OPC interface module accessible to NetView, either by STEPLIB (preferred) or by putting the OPC load library in LINKLST.

EVJ120I *application job opnumber* **CHANGED FROM ERROR STATUS**

Explanation: An application operation has changed its status from ERROR. This causes a Status Display Facility alert to be removed from the OPC Automation: Application in Error panel.

Inserts: Application name, jobname, operation number.

System Action: None.

Operator Response: None.

System Programmer Response: None.

EVJ121I Application *jobname opno* **NOT RESET BECAUSE TIME LIMIT EXCEEDED**

Explanation: This operation was found to be in error status in OPC with error code, UNTV, indicating that the NetView interface was unavailable at the time it came READY. Now that the interface is available, the status was not reset because more time has elapsed since the failure than is permitted by the OPRESET value in the control file (see ENVIRON OPCAO).

Inserts: Application name, Job name, Operation number.

System Action: Operation remains in error status

Operator Response: Manually reset operation when appropriate. OPCACMD may be used to do this.

System Programmer Response: Determine if OPRESET value should be increased.

EHK170I OPCA RECORD DISPLAY FOR: *subsystem*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK171I ID=*subsystem* **TYPE=OPCA, OPID=***operator ID*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK172I LAST COMPLETED STATUS=s, LAST SEQUENCE NUMBER=*nn*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK173I TIMER FLAG=*n* COMPLETION FLAG=*n*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK174I CURRENT SEQUENCE NUMBER=*nn* CHECK MODULE=*module*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK175I EXPECTED STATUS=*status*, TIMER INTERVAL=*nn* TIMER ID= id

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK176I ADNAME=*appname*, WSNAME=*name*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK177I OPNUM= *nnnn*, JOBNAME=*name*, DATE=*mm/dd/yy*, TIME=*hh:mm*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK178I REQUEST=*request*, PARM1=*parm1value*, PARM2=*parm2value*

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EHK002I END

Explanation: This is the output display when an OPC status file is displayed.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ201I EVJTOPPI: TASK IS ACTIVE

Explanation: NetView optional subtask initialization has completed successfully.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ202I EVJTOPPI: PPI TO EVJTOPPI INITIALIZED

Explanation: NetView optional subtask has initialized its NetView PPI queue.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ203I EVJTOPPI: PPI TO EVJTOPPI DEACTIVATED

Explanation: NetView optional subtask has deactivated the NetView PPI queue.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ204I EVJTOPPI: TASK IS TERMINATED

Explanation: NetView optional subtask has terminated successfully.

Inserts: None.

System Action: None.

Operator Response: Information message.

System Programmer Response: Information message.

EVJ205E EVJTOPPI: PPI REQUEST *nn* **FAILED, RC=***rc*

Explanation: EVJTOPPI was unable to complete a NetView PPI request. Refer to the *NetView Application Programming Guide: PPI* for more information on the request type and return code.

Inserts: Request type and return code.

System Action: None.

Operator Response: Notify system programmer.

System Programmer Response: Review log for messages that may indicate the reason for the failure. A possible reason is that NetView is being shutdown.

Note: The OPC operation status will be unchanged. User intervention may be required.

EVJ206I EVJTOPPI: PPI DATA FROM EVJTOPPI DISCARDED

Explanation: EVJTOPPI received request data from a non-OPC subsystem. The data is logged in the NetView log, but is not passed to EVJESPVY.

Inserts: None.

System Action: None.

Operator Response: Informational. Review message EVJ214I for data.

System Programmer Response: Informational.

EVJ207I EVJTOPPI: DSIFRE FAILED FOR USER STORAGE

Explanation: EVJTOPPI was not able to FREE user storage before terminating.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Review error.

EVJ208I EVJTOPPI: DSIFRE FAILED FOR QUEUED STORAGE

Explanation: EVJTOPPI was not able to FREE queued storage before terminating.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Review error.

EVJ209I EVJTOPPI: DSIGET FAILED FOR USER STORAGE

Explanation: EVJTOPPI was not able to GET user storage at initialization.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Review NetView storage allocation and utilization.

EVJ210I EVJTOPPI: DSILCS FAILED TRYING TO GET A SWB

Explanation: EVJTOPPI was not able to GET a service work block.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Review NetView storage allocation and utilization.

EVJ211I EVJTOPPI: ENQ ERROR

Explanation: EVJTOPPI was not able to perform an ENQUEUE.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Check for multiple EVJTOPPI tasks or possible deadlock condition.

EVJ212I EVJTOPPI: DEQ ERROR

Explanation: EVJTOPPI was not able to perform a dequeue.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Check for multiple EVJTOPPI tasks started, or a deadlock condition.

EVJ213I EVJTOPPI: DSILCS FAILED TRYING TO FREE SWB

Explanation: EVJTOPPI was not able to FREE a service work block.

Inserts: None.

System Action: None.

Operator Response: Report problem to system programmer.

System Programmer Response: Contact support center.

EVJ214I EVJTOPPI: *xxxxxxxxxxxxxxxxxxxxxxxxx*

Explanation: This message is sent to the NetView log to document the discarded data.

Inserts: Data.

System Action: None.

Operator Response: Informational.

System Programmer Response: Informational.

EVJ300I EVJSAOPS: DRK/EQQUSINT post was successful

Explanation: Post to OPC using DRK/EQQUSINT was successful.

Inserts: None.

System Action: None.

Operator Response: Informational.

System Programmer Response: Informational.

EVJ302I *cmdname***: INVALID KEYWORD SUPPLIED**

Explanation: Invalid KEYWORD was issued when invoking this command. Message EVJ303I will follow listing the KEYWORD and DATA in error.

Inserts: Command name.

Operator Response: Correct parameters used to invoke the command.

System Programmer Response: Correct the command and retry.

EVJ303I *cmdname*: **KEYWORD ERROR, KEYWORD(***keyword***) DATA(***data***)**

Explanation: This message gives the keywords and data that are invalid when this command was invoked.

Inserts: Command name, keyword and data.

Operator Response: Review parameters used to invoke the command.

System Programmer Response: Modify parameters accordingly.

EVJ304I SRSTAT: Special resource *name* **updated to status** *status*

Explanation: An OPC special resource has come available or unavailable.

Inserts: Special resource name, status (available or unavailable)

Operator Response: None

System Programmer Response: None

EVJ305I SRSTAT: Special resource *name* **update rejected.**

Explanation: OPC has rejected an attempt to update the status of an OPC special resource.

Inserts: Special resource name.

Operator Response: Review parameters used to invoke the command.

System Programmer Response: Modify parameters accordingly.

EVJ306I *routine*: **Unexpected return code RC=***rc* **from** *command*

Explanation: An OPC routine issued a command and received an unexpected return code.

Inserts: Routine name, return code, command

System Action: Routine terminates without further processing.

Operator Response: Report problem to the system programmer.

System Programmer Response: Investigate the cause of the return code. If you require additional assistance, contact the IBM Support Center.

EVJ410I ADID JOBNAME WS OPNO S ERRC IA OPTEXT

Explanation: This is the first line of a multiple-line message. It displays the titles for message EVJ411I.

EVJ411I *applid jobname workstation opid status errcode IA_time text*.

Explanation: This is the second line of a multiple-line message. It displays the corresponding information from message EVJ410.

EVJ412I END OF REQUEST.

Explanation: This message indicated the end of the data.

EVJ413I Unable to find data for requested list

Explanation: This message is issued from OPCALIST or OPCACMD and indicates that no application or operation in the current plan meets the combination of input parameters specified.

Inserts: None.

System Action: Processing for this request halts.

Operator Response: Correct the input to the OPCALIST command or the OPCACMD

panel. If it is not obvious which parameter is error, then specify fewer parameters and issue the command again, so that OPC may return a list of valid application occurrences in the current plan. If there are none, then go to the OPC dialog under ISPF and add an occurrence.

System Programmer Response: None.

EVJ420I PARAMETER ERROR

Explanation: This message indicates that OPC received an invalid parameter, such as workstation, application, or subsystem name.

EVJ421I INITIAL REQUEST TO OPC FAILED

Explanation: OPC is not active or an invalid subsystem has been specified in OPC Automation. Normally, a more detailed message will be issued by OPC, and echoed to the log if available under OPC Automation message EVJ011I.

EVJ422I LIST REQUEST FAILED

Explanation: OPC could not respond with data. This may indicate an invalid search argument or subsystem name. Normally, a more detailed message will be issued by OPC, and echoed to the log if available under OPC Automation message EVJ011I.

EVJ423I MODIFY CPOC RC = *return_code*

Explanation: The OPC modify command failed. This may indicate invalid parameters. Normally, a more detailed message will be issued by OPC, and echoed to the log if available under OPC Automation message EVJ011I.

EVJ430I ADID JOBNAME WS OPNO S ERRC IA OPTEXT

Explanation: This is the first line of a multiple-line message. It displays the titles for message EVJ431E, as follows:

ADID OPC Application ID

JOBNAME MVS Job Name

WS OPC Workstation

OPNO OPC Operation Number

S OPC Status

ERRC OPC Error Return Code

IA OPC Input Arrival Time

OPTEXT OPC Operations Text

EVJ431E *adid jobname ws opno s errc ia optext*

Explanation: This is the continuation of a multiple-line message begun with EVJ430I.

Inserts: The inserts are as follows:

adid OPC Application ID

jobname MVS Job Name

ws OPC Workstation

opno OPC Operation Number

s OPC Status

errc OPC Error Return Code

ia OPC Input Arrival Time

optext OPC Operations Text

EVJ432E INVALID PARAMETERS FOUND

Explanation: This message indicates that OPC received a parameter in an invalid format.

EVJ433E AN ERROR OCCURRED RETRIEVING CALENDAR DATA FROM OPC

Explanation: OPC Automation could not retrieve calendar information. Normally, a more detailed message will be issued by OPC, and echoed to the log if available under OPC Automation message EVJ011I.

EVJ434E INITIAL REQUEST TO OPC FAILED

Explanation: OPC is not active or an invalid subsystem has been sent to OPC. Normally, a more detailed message will be issued by OPC, and echoed to the log if available under OPC Automation message EVJ011I.

EVJ435E List request to OPC failed

Explanation: For some reason, OPC could not respond with data. Normally, a more detailed message will be issued by OPC, and echoed to the log if available under OPC Automation message EVJ011I.

EVJ440I *date day* **WORK|FREE.**

Explanation: This message tells you whether today is a work day or a free day in the OPC calendar.

With APAR OW23552 the date is expanded and is shown in the format `mm/dd/yyyy`, for example, 05/22/1997.

Codes Posted to OPC by OPC Automation

S998

Explanation: Job not defined to any active system in sysplex. A Request to run an operation from OPC was received by OPC Automation. The control file mapped the workstation (NV*nn*) to SYSPLEX. All active SA OS/390 domains in the local sysplex were queried but no definition was found for the job name specified in the OPC request.

This may be the result of one or more systems (or their SA OS/390) being inactive or it may be a definition error. If all systems where the job is defined are offline then when SA OS/390 gateway connectivity is re-established the operation will be retried.

S999

Explanation: Gateway to remote NetView was unavailable. A request to run an operation from OPC was received by the automation platform (NetView/OPC) address space, but it was destined for another NetView domain. An attempt was made to forward this operation to that domain, designated by the workstation name in the request, but this attempt failed. Either the gateway from the local system to the remote domain was down, or no response was received from the remote system was received after two minutes of waiting.

UNTV

Explanation: NetView unavailable. The OPC exit in OPC was unable to transmit this request to the automation platform (NetView/OPC) address space. Either the NetView in the system running the OPC/ESA Controller was down at the time this request was initiated or the NetView SSI was down.

U001

Explanation: Invalid status for the subsystem. The subsystem failed to start or terminate in the interval, or has changed to an unexpected status.

U002

Explanation: Invalid flag combination detected. This may indicate autotask definition errors. Review the NetView log for additional information.

U003

Explanation: Missing definitions in the automation control file. Review the control file definitions for OPCA CODE, OPCAPARM, and OPCACMD.

U004

Explanation: Error in reading status record for the subsystem. Usually indicates that a record has been deleted in the status file. Use the OPCAQRY command to display the records. If it has been deleted, use EVJESPIN CMD=INIT to create the records.

U005

Explanation: Sequence Error. This is usually an indication of two or more requests in progress for the same subsystem. Review the NetView log for additional information.

U006

Explanation: Configuration Error. An Automation Configuration File statement of the type OPCA DOMAINID,CODE= specifies SYSPLEX. The OPC Controller is not running in a system where SA OS/390 Release 3 or later is installed. Use of the SYSPLEX keyword on this statement requires SA OS/390 Release 3 or later to be installed where the OPC Controller runs.

Glossary of Terms

The intent of this glossary is to define terms as TME 10 OPC uses them. However, where applicable, terms are taken from the *IBM Dictionary of Computing*, New York; McGraw-Hill, 1994. These terms are marked by an asterisk (*). Unless otherwise noted, the definitions below apply equally well to OPC/A, OPC/ESA and TME 10 OPC.

A

actual duration. At a workstation, the actual time in hours and minutes it takes to process an operation from start to finish.

APAR. Authorized program analysis report. A report of a problem caused by a suspected defect in a current unaltered release of a program.

all workstations closed. A user defined interval during which *all* OPC's workstations are not available for running applications under OPC's control.

Note: All the workstations could be either shut down *or* simply not available to OPC.

application. (1) A group of related operations performed together to satisfy a specific end user task. (2) A measurable and controllable unit of work that completes a specific user task such as the running of payroll or financial statements. The smallest entity that an application can be broken down into is an operation. Generally, several related operations make up an application.

application description. A database description of an application.

application ID. The name of an application. Examples: Y1976, Payroll.

arrival (A). Status of an operation that indicates it is waiting for the input to arrive before processing.

authority. The ability to access a protected resource.

authority group. A name used to generate a RACF resource name for authority checking.

automatic events. Events recognized by or triggered by an executing program. Automatic events are usually generated by OPC job tracking programs but may also be created by a user-defined program.

automatic reporting workstation. A workstation that reports events (the starting and stopping of operations) in real time to OPC, such as a processor or printer.

automatic job recovery. An OPC function which allows you to specify, in advance, alternative recovery strategies for applications or operations ended in error.

availability. * The degree to which a system (and in OPC, an application) or resource is ready when needed to process data.

B

batch loader. An OPC batch program you can use to create and update information in the application description and operator instruction databases.

bracketed DBCS. A MIXED format field consisting of a DBCS part only, that is, DBCS characters enclosed by a shift-out/shift-in control character pair.

browse. An ISPF/PDF dialog function that manages data for display only. This function lets the user view but not change data.

C

CP. Current plan.

calendar. The data that defines the operation department's processing schedule in days and periods.

capacity. The actual number of parallel servers and workstation resources available during a specified open time interval.

capacity ceiling. The maximum number of operations a workstation can handle simultaneously.

case code. A code in the automatic job recovery function that represents a group of abend codes or return codes. Any code in the JOBCODE and STEPCODE parameters is considered a potential case code if defined as such in the case code macro.

closed workstation. A workstation that is unavailable to process work for a specific time, day, or period.

command. * A request from a terminal for the performance of an operation or the execution of a particular program. A character string from a source external to a system that represents a request for system action.

complete. Status of an operation indicating that it has finished processing.

completion code. An OPC system code indicating how the processing of an operation ended at a workstation.

complex of processors. A JES2 multi-access spool system or a JES3 system with more than one processor.

computer workstation. A workstation that performs MVS processing and usually reports status to OPC/A automatically. A processor when used as a workstation. It can refer to single processors or multiprocessor complexes serving a single job queue (for example JES2 or JES3 systems).

controller. The portion of TME 10 OPC or OPC/ESA that runs on the controlling processor and contains the tasks that manage OPC databases and plans. Comparable to the OPC/A PCS.

controlling processor. The processor on which the Production Control System (PCS) executes. See host processor.

critical path. The route within a network with the least amount of slack time.

critical resource. The term used in OPC 3.1 for 'workstation resource'. See workstation resource.

current plan. A minute by minute schedule of each operation of an application. It reflects the current state of the operating environment showing the status of work completed and work still to be done.

current schedule. The database that contains the current plan information.

cyclic interval. The number of days in a cyclic period.

cyclic period. A period with a specific origin date and set frequency. A cyclic period can be broken down into two types:

- Those that include work and free days
- Those that include only work days.

Cyclic periods must always represent a fixed time period in days. For example, week (7 days).

D

daily plan. A set of plans that shows work that the operations department does on a particular day or shift. A list by day and application of all operations to be performed within the operations department.

default calendar. (1) A calendar that you have defined for OPC/A to use when you do not specify a calendar in an application description. (2) A calendar that OPC/A uses if you have neither specified a calendar in an application description, nor defined your own default calendar. (3) The name (DEFAULT) given to your OPC/A Release 1 calendar by the migration program if you migrate to OPC/A Release 2. (OPC/A Release 1 allows you only one calendar, Release 2 allows you multiple calendars.)

deadline. See deadline date and deadline time.

deadline date. The latest date by which an occurrence must be complete.

deadline time. The latest time by which an occurrence must be complete.

defined. An open day status which indicates that specific open time intervals exist for a workstation on a particular day.

dependency. A relationship between two operations where the first operation must successfully finish before the second operation can begin.

dialog. The user's online interface with OPC.

displacement. A number specifying 'Number of Days from Period Start' or 'Number of Days from Period End'. Sometimes called offset. See offset.

duration. The time an operation is active at a workstation.

E

edit. An ISPF/PDF dialog function that is used for editing text, collecting data, and modifying data.

end user. A person who uses the services of the data processing center.

ended in error (E). The OPC reporting status for an operation that has ended in error at a workstation.

error code. The system completion code or program return code for automatic reporting workstations. The code entered by the workstation operator for manually reporting workstations.

event. An action in the operations department that results in an operation's change of status and a change in the current schedule.

event handler. A separate load module that changes the status of operations.

event manager. The OPC/A function that processes all job tracking events and determines which of these are OPC/A related.

Event Management Subsystem (EMS). The Event Management Subsystem runs as an MVS subsystem, tracking and logging event records on DASD. It is required on every processor in an OPC/A configuration. Equivalent to the OPC/ESA Tracker.

event reader. An OPC/A task that reads event records from an event data set.

event tracking. See job tracking.

Event Triggered Tracking (ETT). A component of OPC/A that waits for specific events to occur; and when they occur, it adds a predefined application to the current plan. ETT recognizes two types of events: the reader event, which occurs when a job enters the JES reader, and the resource event, which occurs when the availability status of a special resource is set to 'yes'.

event writer. An OPC/A task that writes event records in an event data set.

exclusive. The state of a special resource indicating that it is fully used by one operation and cannot be used simultaneously by other operations.

exclusive resource. A workstation resource that is solely used by one operation and cannot be shared with other operations.

expected arrival time. The time when an operation is expected to arrive at a workstation. It may be calculated by daily planning or specified in the long-term plan.

extend current period. An OPC function that allows the user to extend the current plan up to a maximum of 504 hours (21 days) from the current end date.

external dependency. A relationship between two occurrences where an operation in the first occurrence must successfully finish before an operation in the second occurrence can begin processing. See dependency.

external predecessor. The name given to the operation in the first occurrence of an external dependency that must finish before its external successor can begin processing.

external successor. The name given to the operation, in the second occurrence of an external dependency, that cannot begin until its external predecessor completes.

F

feedback limit. A numeric value from 100 to 999 which defines the limits within which actual data collected in job tracking is fed back and used by OPC/A.

free day. A nonworking day.

free day rule. A rule that determines how OPC will treat free days when the application run day falls on a free day. The rule is as follows:

Excluded: Free days excluded; only work days are taken into account.

Included: Free days included; all days are taken into account, as follows:

(1) Run before the free day.
(2) Run after the free day.
(3) Run on the free day.
(4) Do not run on the free day.

G

general workstation. A workstation where activities, usually manual, and other than printing and processing, are carried out. Manual activities might be data entry or job setup. A general workstation reporting to OPC/A is usually manual, but can be automatic.

generic search argument. A portion of a key containing a generic search character which in OPC is an asterisk (*) or percent sign (%). The asterisk represents any string of characters and the percent sign any single character. Use with any portion of a key to search the database for items to be displayed as part of a listing. Examples: %ABC, A*C, A*.

H

host processor. * A processor that controls all or part of a user application network. * In a network, the processing unit in which the access method for the network resides.

highest return code. A numeric value from 0 to 4095. If this return code is exceeded during a job's processing, the job will be reported as ended in error.

I

incident log. An optional function available under the job completion checker.

input arrival. The user-defined date and time an operation or an application becomes ready for processing.

internal dependency. A relationship between two operations within an occurrence where the first operation must successfully finish before the second operation can begin.

internal predecessor. The name given to the operation of an internal dependency that must finish before its internal successor can begin processing.

internal successor. The name given to the operation of an internal dependency that cannot begin until its internal predecessor completes processing.

ISPF. Interactive System Productivity Facility.

interrupted (I). An OPC reporting status for an operation indicating that the operation has been interrupted while processing.

J

job. * A set of data that completely defines a unit of work for a computer. A job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. In OPC, an operation performed at a CPU workstation.

job completion checker (JCC). An optional function of OPC that provides an extended checking capability of the results from CPU operations.

job control language (JCL). * A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

JES. Job Entry Subsystem.

job entry subsystem (JES). * A system facility for spooling, job queuing, and managing I/O.

job tracking. A function of OPC/A PCS that follows events in the operations department in real time and records status changes in the current schedule.

job setup. The preparation of a set of JCL statements for a job at an OPC/A workstation you defined for this purpose.

job submission. An OPC/A process that presents jobs to MVS for running on an OPC/A defined workstation at a time specified in the daily plan.

JS. The JCL repository data set.

K

keyword. * A symbol that identifies a parameter. * A part of a command operand that consists of a specific character string (such as DSNAME=).

keyword parameter. * A parameter that consists of a keyword, followed by one or more values.

L

LTP. Long-term plan.

last operation. (1) An operation in an occurrence that has no internal successor. (2) The terminating node in a network.

latest start. The latest start day and time (calculated by OPC/A) for an operation that will allow all occurrences to meet their deadline.

layout ID. A unique name that identifies a specific ready list layout.

limit for feedback. See feedback limit.

local. * Synonym for channel-attached.

local processor. * In a complex of processors under JES3, a processor that executes users' jobs and that can assume global functions in the event of failure of the global processor. In OPC, a processor in the same installation that communicates with the controlling OPC processor through shared DASD communication.

long-term plan. A high-level schedule of processing activities for the forthcoming weeks and months. The scope of a long-term plan can be from one day to four years.

The long-term planning function produces a list of application occurrences identified by name, date, and run time for a specified planning period.

M

manual reporting workstation. A type of workstation reporting where events, once they have taken place, are manually reported to OPC. This type of reporting requires that some action be taken by a workstation operator. Manual reporting is usually performed from a list of ready operations.

mass updating. A function of the application description dialog where a large update to the application database can be requested.

modify current plan. An OPC dialog function used to dynamically change the contents of the current

schedule to respond to changes in the operation environment. Examples of special events that would cause alteration of the current schedule are: a rerun, a deadline change, or the arrival of an unplanned application.

most critical application occurrences. Those unfinished applications that have a latest start time that is less than or equal to the current time.

N

Network Event Communicator (NEC). The Network Event Communicator collects event information from event data sets and transmits it to the NEC executing on the OPC/A controlling processor. It is required on the controlling processor and on each remote processor where remote job tracking is used.

node. * In a network, a point where one or more functional units interconnect transmission lines.

noncyclic period. A period that has a varying frequency for which you must define each origin date. Examples: month, payroll period, and quarterly.

nonreporting. A reporting attribute of a workstation which indicates that information is not fed back to OPC.

O

OPC/A. Operation Planning and Control/Advanced

OPC/ESA. Operations Planning and Control/Enterprise Systems Architecture

occurrence. Each instance of an application in the long-term plan and current plan is called an occurrence.

An application occurrence is one attempt to process that application. Occurrences are distinguished from one another by run date, input arrival time, and application ID. For example, one application that runs four times a day is said to have four occurrences a day.

offset. A maximum of 12 positive and 12 negative values in the ranges 1 to 999 and -1 to -999 that indicate on which days of a calendar period an application shall run. See displacement.

OPC host. The processor where OPC updates the current plan database.

OPC local processor. A processor that connects to the OPC host or remote processor through shared event data sets.

OPC/A remote processor. A processor connected to the OPC/A host processor by a VTAM network. An OPC/A event writer (EMS) and an event transmitter (NEC) are installed on the remote processor and transmit events to the OPC/A host processor by VTAM. With OPC/ESA, the Tracker combines these functions.

open time interval. The time interval during which a workstation is active and can process work.

operation. An operation is a unit of work that is part of an occurrence and is processed at a workstation.

operation waiting for arrival. The status of an operation that indicates that the necessary input has not arrived at a workstation so that the operation can begin processing. This status is applicable only for operations without predecessors.

operation status. The status of an operation at a workstation.

An operation's status can be one of the following:

A Waiting for input to arrive.

R Ready for processing. All predecessors are complete.

***** Ready for processing. There is a nonreporting predecessor. All predecessors are complete but one or more predecessors were executed at a nonreporting workstation.

S Started.

I Interrupted operation.

C Complete.

E Operation ended in error.

W Waiting for predecessor to complete.

U Undecided. The status is not known.

operator. * (ISO) A symbol that represents the action to be performed in a mathematical operation. * In the description of a process, that which indicates the action to be performed on operands. * A person who operates a machine.

option. A selection item on a menu panel in the OPC dialog.

origin date. The date on which a period (cyclic or noncyclic) starts.

P

panel. * A particular arrangement of presentation windows used to show information to the user. OPC uses only fixed-format panels.

parallel operations. Operations at workstations that are not dependent on one another and therefore can be performed simultaneously.

parallel server. The function that processes operations at a workstation, especially when there is more than one such function. See server.

parameter. * (ISO) A variable that is given a constant value for a specified application and that may denote the application. * A name in a procedure that is used to refer to an argument passed to that procedure.

pending application description. An application description which is incomplete and not ready for use in planning or scheduling.

period. A business processing cycle. A time period defined in the OPC/A calendar. They are used to describe when, and how often, applications are to run.

period name. A name of a period. Examples are week, month, quarter and fiscal period end.

period type. Periods are of two types: cyclic or noncyclic.

PDF. program development facility.

predecessor. An operation of an internal or external dependency that must finish successfully before its successor operation can begin.

printout routing. The ddname of the daily planning printout data set.

print workstation. A workstation that prints output and usually reports status to OPC automatically.

priority. A digit from 1 to 9 (where 1 = low, 8 = high, and 9 = urgent) that determines how OPC schedules applications to run. A number from 1 (low priority) to 9 (high priority) which establishes the importance of an application relative to other applications.

processor. * (ISO) In a computer, a functional unit that interprets and executes instructions. * A functional unit or part of another unit (such as a terminal or a processing unit) that interprets and executes instructions.

Production Control System (PCS). The Production Control System contains the controlling functions, all dialogs, and OPC/A's own batch programs. The program controls the entire OPC/A installation, including remote sites. Equivalent to the OPC/ESA Controller.

program interface. An OPC/A interface that allows a user-written program to issue various types of requests to the OPC/A subsystem.

Q

QCP. Query current plan.

R

RACF. Resource Access Control Facility.

read authority. A type of access authority that allows a user to read the contents of a data set, file, or storage area, but not to change it.

ready (R). The status of an operation indicating that predecessor operations are complete and that the operation is ready for processing.

ready list. A display list of all the operations ready to be processed at a workstation. Ready lists are the means by which workstation operators manually report on the progress of work.

recovery. See automatic job recovery.

remote processor. A processor connected to the OPC host processor by a VTAM network.

remote job tracking. The function of tracking jobs on remote processors connected by VTAM links to an OPC controlling processor. This function enables a central site to control the submitting, scheduling, and tracking of jobs at remote sites.

replan current period. An OPC function that recalculates planned start times for all occurrences to reflect the actual situation.

reporting attribute. A code that specifies how a workstation will report events to OPC.

rerun. An OPC function where an application or part of an application that ended in error can be run again.

rescale factor. A value from 0 to 100 used to reduce the new duration value by a given percentage amount.

return code. An error code issued by OPC for automatic reporting workstations.

row command. A dialog command used to manipulate data in a table.

run cycle period. A time frame defining the effective period and run days of a calendar period.

run day. The date on which an application is to run. It is expressed as a number relative to the start or the end of a run cycle period.

S

SAF. System Authorization Facility.

search argument. A value that is used to search the database for an item that is to be part of a displayed listing.

selection criteria. Search arguments entered on a list criteria panel in the dialog that limit the contents of a listing.

server. A program or device set up for a workstation to perform a service for that particular type of workstation. For example, an initiator is a server for a computer workstation. A printer is a server for a print workstation.

service functions. Functions of OPC that let the user deal with exceptional conditions such as investigating problems, preparing APAR tapes, and testing OPC during implementation.

shared DASD. Direct access storage device that can be accessed from more than one processor.

shared resource. A special or workstation resource that can be used simultaneously by more than one operation while the operation is processed at a work station.

slack. Used to refer to 'spare' time. Can be calculated for the critical path by taking 'Deadline less the Input Arrival less the Sum of Operation Durations'.

smoothing factor. A value between 0 and 100 that controls the extent to which actual durations are fed back into the application description database.

SMP. System Modification Program.

special resource. Resources that are not associated with a particular workstation but are needed to process work there.

splittable. Refers to an operation that can be interrupted while processing at a workstation.

standard. User specified open time intervals for a typical day at a work station.

status. The current state of an operation or an occurrence.

started (S). An OPC reporting status of an operation or an application indicating that an operation or an occurrence is started.

submit/release data set. A data set shared between the OPC host and a local OPC processor that is used to send job stream data and job release commands from the host to the local processor.

subresources. A set of resource names and rules for the construction of resource names. OPC uses these names when checking a user's authority to access individual OPC records.

subsystem. * A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system.

successor. An operation in an internal or external dependency that cannot begin until its predecessor completes processing.

sysout class. * An indicator used in data definition statements to signify that a data set is to be written on a system output unit. It applies only to print workstations.

T

temporary operator instructions. Operator instructions that have a specific time limit during which they are valid. They will be displayed to the workstation operator only during that time period.

TME 10 OPC. TME 10 Operations Planning and Control

tracker. The portion of TME 10 OPC or OPC/ESA that runs on every system in your complex. It acts as the communication link between the MVS system that it runs on and the controller. Comparable to the OPC/A EMS and NEC components.

tracking event log. A log of job tracking events and updates to the current schedule.

transport time. The time allotted for transporting materials from the workstation where the preceding operation took place, to the workstation where the current operation is to occur.

TSO. Time Sharing Option.

time zone support. A feature of OPC that allows applications to be planned and run with respect to the local time of the processor that runs the application. Some networks may have processors in different time zones. The controlling processor will make allowance for differences in time during planning activities, for example the input arrival time of predecessor applications, to make sure that interacting activities are correctly coordinated.

turnover. A subfunction of job tracking that is activated when job tracking creates an updated version of the current schedule.

U

undecided (U). An OPC reporting status for an operation or an application indicating that the status is not known.

update authority. Access authority given to a user by RACF to use the ISPF/PDF edit functions of the OPC dialog. Access authority to modify a master file or data set with the current information.

V

validity period. The time interval defined by an origin date and an end date within which a run cycle or an application description is valid.

versions. Applications with the same ID but different validity dates.

VSAM. Virtual Sequential Access Method.

VTAM. Virtual Telecommunication Access Method.

W

waiting (W). An OPC reporting status (for an application) indicating that it is waiting for a predecessor operation to complete.

waiting list. A list of submitted jobs that are waiting to be processed.

work day end time. The time at which OPC will consider a work day to have ended when that work day immediately precedes a free day. For example, if you specify Saturday to be a free day, you could specify 08.00 hours. Saturday morning as the end of Friday's work day. OPC can then plan work to be done from 00.00 to 08.00 Saturday morning, as if that time was actually part of Friday.

workstation. A unit, place, or group that performs a specific data processing function. A logical place where work occurs in an operations department.

OPC requires that you define the following characteristics for each workstation: the type of work it does, the quantity of work it can handle at any particular time, and the times it is active. The activity that occurs at each workstation is called an operation.

workstation description database. An OPC database containing descriptions of the workstations in the operations department.

workstation resources. Limited resources defined for each workstation that an operation requires a certain amount of to process work.

workstation type. Each workstation can be one of three types: computer, print, or general.

work day. A day on which applications can normally be scheduled to start.

Index

Special Characters

A

B

C

D

P

R

S

T

U

W

Communicating Your Comments to IBM

System Automation for OS/390
AOC/MVS OPC Automation
Operator's Guide
and Scheduler Reference
Version 1 Release 4

Publication No. SC23-3821-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of the book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF form and either send it postage-paid in the United States, or directly to:

  IBM Deutschland Entwicklung GmbH
  Department 3248
  Schoenaicher Strasse 220
  D-71032 Boeblingen
  Federal Republic of Germany

- If you prefer to send comments by FAX, use this number:
  - (Germany): 07031-16-3456
  - (Other countries): (+49)+7031-16-3456

- If you prefer to send comments electronically, use this network ID:

  IBM Mail Exchange: DEIBMBM9 at IBMMAIL

  Internet: s390id@de.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Readers' Comments — We'd Like to Hear from You

System Automation for OS/390
AOC/MVS OPC Automation
Operator's Guide
and Scheduler Reference
Version 1 Release 4

Publication No. SC23-3821-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC23-3821-02

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Attn: Dept EHJ - BP/003D
6300 Diagonal Highway
Boulder, CO 80301-9151

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5685-151

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC23-3821-02