System Automation for OS/390

IBM

IMS Automation Programmer's Reference and Operator's Guide

Version 2 Release 2

System Automation for OS/390

IBM

IMS Automation Programmer's Reference and Operator's Guide

Version 2 Release 2

> **Note!**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page ix.

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.

Programming Interface Information

This book documents programming interfaces that allow the customer to write programs to obtain the services of System Automation for OS/390.

Trademarks

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

DB2	MVS/ESA
IBM	NetView
IMS/ESA	OS/390
MVS	VTAM

NetView is a trademark of Tivoli Systems Inc. in the United States, or other countries, or both.

ix

About This Book

This book describes how to customize and operate IMS Automation. IMS Automation is a feature of System Automation for OS/390 that provides a simple and consistent way to monitor and control all of the IMS regions, both local and remote, within your organization. This automation feature automates, simplifies, and standardizes console operations and the management of component, application, and production related tasks.

Who Should Use This Book

This book is intended for two kinds of users/user groups:

- System programmers, system designers, and application designers who will automate IMS using IMS Automation.

  For these users, all three parts of the book will be of interest.

  Installing and customizing IMS Automation requires a programmer's understanding of NetView, IMS, SA OS/390, and IMS Automation, because most of the definitions take place in these programs. Also, you will modify JCL, command lists, and programs for some of the automation functions

- Operators and administrators who manage and monitor IMS subsystems.

  These users will mainly need part 1 and part 3.

  For operators, a working knowledge of IMS will be assumed.

What's in This Book

This book contains the following:

Part 1, "Introducing IMS Automation"
> Explains some main concepts of SA OS/390 and describes the special functions of IMS Automation.

Part 2, "Customizing IMS Automation"
> Describes the customization of IMS Automation and contains reference sections for MESSAGES policy items and for the programming interface.

Part 3, "Using IMS Automation"
> Describes the operator interface of IMS Automation.

Notation for Format Descriptions

The reference sections of this manual contain format descriptions of commands and of entries in the SA OS/390 policy database. The notation used for these descriptions is as follows:

- Items shown in braces { } represent alternatives. You must choose one. For example,

  `{A|B|C}`

  indicates that you must specify one item only: A, B, or C.
- Items shown in brackets [ ] are optional. You may choose one. For example,

  `[A|B|C]`

  indicates that you may enter A, B, or C, or you may omit the operand.

- A series of three periods (...) indicates that a variable number of items may be included in the list.
- An underscored item shows the default that the system will choose if you do not specify an item. For example,

  [A|**B**|C]

  indicates that if no operand is specified, B is assumed.
- Lowercase italicized items are variables; substitute your own value for them.
- Uppercase items must be entered exactly as shown.
- Parentheses must be entered as shown.
- Where operands can be abbreviated, the abbreviations are shown in capital letters. For example, ALL can be entered as A or ALL.
- Commas are used as delimiters between parameters. The last parameter does not require a comma after it. Because of this, we place the comma in front of a parameter to show that if you add this parameter, you need a comma, as for example in

  XYZ   [A[,B[,C]]]

  However, the comma actually goes after the preceding parameter and needs to be on the same line as that parameter.

Related Publications

The System Automation for OS/390 Library

The following table shows the information units in the System Automation for OS/390 library:

Table 1. System Automation for OS/390 Library

Title	Order Number
System Automation for OS/390 Planning and Installation	SC33-7038
System Automation for OS/390 Customizing and Programming	SC33-7035
System Automation for OS/390 Defining Automation Policy	SC33-7039
System Automation for OS/390 User's Guide	SC33-7040
System Automation for OS/390 Messages and Codes	SC33-7041
System Automation for OS/390 Operator's Commands	SC33-7042
System Automation for OS/390 Programmer's Reference	SC33-7043
System Automation for OS/390 CICS Automation Programmer's Reference and Operator's Guide	SC33-7044
System Automation for OS/390 IMS Automation Programmer's Reference and Operator's Guide	SC33-7045
System Automation for OS/390 OPC Automation Programmer's Reference and Operator's Guide	SC23-7046
System Automation for OS/390 Licensed Program Specifications	SC33-7037

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

    IBM Online Library z/OS Software Products Collection (SK3T-4270)

> **SA OS/390 Homepage**
>
> For the latest news on SA OS/390, visit the SA OS/390 homepage at
> http://www.ibm.com/servers/eserver/zseries/software/sa

Related Product Information

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. Related Products Books

Title	Order Number
ISPF User's Guide	SC34-4484
ISPF Dialog Management Guide and Reference	SC34-4266
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
OS/390: MVS System Commands	GC28-1781
MVS/ESA SPL Application Development Macro Reference	GC28-1857
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 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
S/390 Managing Your Processors	GC38-0452
Tivoli/Enterprise Console User's Guide Volume I	GC31-8334
Tivoli/Enterprise Console User's Guide Volume II	GC31-8335
Tivoli/Enterprise Console Event Integration Facility Guide	GC31-8337
Tivoli NetView for OS/390 Administration Reference	SC31-8222
Tivoli NetView for OS/390 Application Programming Guide	SC31-8223

Table 2. Related Products Books (continued)

Title	Order Number
Tivoli NetView for OS/390 APPN Topology and Accounting Agent	SC31-8224
Tivoli NetView for OS/390 Automation Guide	SC31-8225
Tivoli NetView for OS/390 AON Customization Guide	SC31-8662
Tivoli NetView for OS/390 AON User's Guide	GC31-8661
Tivoli NetView for OS/390 Bridge Implementation	SC31-8238
Tivoli NetView for OS/390 Command Reference Vol. 1	SC31-8227
Tivoli NetView for OS/390 Command Reference Vol. 2	SC31-8735
Tivoli NetView for OS/390 Customization Guide	SC31-8228
Tivoli NetView for OS/390 Customization: Using Assembler	SC31-8229
Tivoli NetView for OS/390 Customization: Using Pipes	SC31-8248
Tivoli NetView for OS/390 Customization: Using PL/I and C	SC31-8230
Tivoli NetView for OS/390 Customization: Using REXX and CLIST Language	SC31-8231
Tivoli NetView for OS/390 Data Mode Reference	SC31-8232
Tivoli NetView for OS/390 Installation: Getting Started	SC31-8767
Tivoli NetView for OS/390 Installation: Migration Guide	SC31-8768
Tivoli NetView for OS/390 Installation: Configuring Graphical Components	SC31-8770
Tivoli NetView for OS/390 Installation: Configuring Additional Components	SC31-8769
Tivoli NetView for OS/390 Messages and Codes	SC31-8237
Tivoli NetView for OS/390 MultiSystem Manager User's Guide	SC31-8607
Tivoli NetView for OS/390 NetView Management Console User's Guide	GC31-8665
Tivoli NetView for OS/390 User's Guide	SC31-8241
Tivoli NetView for OS/390 RODM and GMFHS Programming Guide	SC31-8233
Tivoli NetView for OS/390 Security Reference	SC31-8606
Tivoli NetView for OS/390 SNA Topology Manager and APPN Accounting Manager Implementation Guide	SC31-8239
Tivoli Management Platform Reference Guide	GC31-8324
TSO/E REXX/MVS User's Guide	SC28-1882
TSO/E REXX/MVS Reference	SC28-1883
VM/XA SP GCS Command and Macro Reference	SC23-0433
VSE/SP Unattended Node Support	SC33-6412
VTAM Messages and Codes	SC31-6493
VTAM Network Implementation Guide	SC31-6404
VTAM Network Implementation Guide	SC31-6434

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most
messages you encounter, as well as for some system abends and codes. Using

LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

Accessing z/OS™ licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

http://www.ibm.com/servers/resourcelink

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code. [1]

To obtain your IBM Resource Link user ID and password, log on to:

http://www.ibm.com/servers/resourcelink

To register for access to the z/OS licensed documents:
1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

---

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

Part 1. Introducing IMS Automation

This part describes principal concepts of SA OS/390, including some NetView related information, and gives an overview of the additional facilities offered by IMS Automation.

Chapter 1. Principal Concepts of SA OS/390

This section sketches some fundamentals of SA OS/390®. For more detailed information see the SA OS/390 documentation.

Automation Policies

System automation primarily deals with starting and stopping applications in accordance with their interrelationships. These interrelationships include dependencies of applications on other applications as well as being a component application of an application complex. Also, system automation supports permanent availability of an application by moving the application to another system in case of an unrecoverable abend (see "Application Groups" on page 8).

All applications and systems that you want to include in automation must be defined to SA OS/390 in an automation *policy database*. This database contains the objects to be managed by SA OS/390, and the rules according to which automation of these objects proceeds. You access the policy database from the so-called *customization dialogs*. The customization dialogs are described in *System Automation for OS/390 Defining Automation Policy*.

The objects that are defined in the policy database are called *policy objects* or *entries*. Applications and systems, for example, are policy objects. Every policy object belongs to an *entry type* which is identified by a three letter code; thus, applications belong to the entry type APL.

Policy objects have automation-related properties and are associated with one another; these properties and connections are called *policy items*. For example, there is a policy item STARTUP for applications that specifies how SA OS/390 is to start the application.

What you enter in the policy database are policy objects. However, the objects that can be automated are not these policy objects, but so-called *resources*, which are automatically generated from the policy objects.

This is especially important in the case of applications, since the resources that correspond to an application always represent a *subsystem*, that is, a combination of the application with a system on which it is intended to run; thus, one application can correspond to several subsystems. These resources are generated when an application is linked to a system in the policy database. Note also that some properties and connections are defined on the application (policy object) level (see "Triggers" on page 7) and handed down to all corresponding resources, while others are specified at the resource level (see "Dependencies, Request Propagation, and Desired State" on page 4), and therefore only apply to that resource.

The names of the resources have the following format:

resource_name/entry_type[*/system_name*]

The most common entry types are APL (application), APG (application group), and SYS (system). The system name is omitted when the resource is associated with a sysplex, and not a single system.

The policy database must be converted into an *automation control file* (ACF) in order to be accessible to SA OS/390.

Goal-Driven Automation

A basic concept of SA OS/390 is to distinguish between the *desired* state of a resource and (broadly speaking) its *actual* state. Every resource has a desired state, which is either AVAILABLE or UNAVAILABLE; AVAILABLE is the default. This desired state, which is also called the automation *goal*, can be different from the actual state; a resource whose desired state is to be running (AVAILABLE), can actually be down. SA OS/390 always tries to keep the actual state in line with the desired state, but sometimes this is not possible.

SA OS/390 is called goal *driven* because all requests that can be made to it from the outside refer to the desired state of the target resource. When an operator passes a start request for a resource to SA OS/390, this is a request to set the desired state of the resource to AVAILABLE. It is up to SA OS/390 to decide whether (1) this is at all possible, and if so, whether (2) the actual state can be modified accordingly:

1. Making a request does not automatically lead to a change of the desired state of the target resource. Rather, SA OS/390 compares the *priority* of the new request with that of the last successful request. Only when the new request has a higher priority does SA OS/390 change the desired state of the resource. Note that this presupposes that the old request is still available. For more details on this topic, see "Persistency of Requests and Conflicting Requests" on page 6.
2. The latter decision mainly depends on the *dependencies* between the target resource and other resources, and on the *triggers* that may have been associated with it. Dependencies and triggers are defined in the policy database. For more information, see "Dependencies, Request Propagation, and Desired State", and "Triggers" on page 7.

Dependencies, Request Propagation, and Desired State

One of the main tasks of system automation when starting or stopping a resource is to consider the dependencies that exist between the resource to be started/stopped and other resources. Certain resources can only be started when certain other resources are already running (start dependencies), and certain resources can only be stopped when certain other resources are already down (stop dependencies). Note that start and stop dependencies are in principle independent of each other, although if A can only be started when B is running, then it will, as a rule, not be possible to stop B unless A has been stopped beforehand.

Such dependencies can be specified in the policy database. The only restriction is that the dependent and the supporting resource must belong to the same sysplex (they need *not* reside on the same system). SA OS/390 takes dependencies into account when it is requested to start or to stop a resource. By default, it will try to start/stop all resources on which the target resource of the request directly or indirectly depends. The mechanism by which this is accomplished is called *request propagation*. It is best explained by an example.

Example 1: Let A, B, and C be resources so that A can only be started when B is running, and B can only be started when C is running. C is supposed

to have no start dependencies. Suppose, furthermore, that A, B, and C are all actually down, and that this conforms to their desired state (which is UNAVAILABLE).

Finally, assume that A, B, and C are not associated with any trigger (for the significance of this, see "Triggers" on page 7), and that there are no requests pending for any of the three resources (see "Persistency of Requests and Conflicting Requests" on page 6).

This situation is displayed in Figure 1. The labels of the arrows specify the dependency type. **MakeAvailable/WhenAvailable** is the format in which SA OS/390 specifies that the dependent (lower) resource, which is referred to by **MakeAvailable**, can only be started when the supporting (upper) resource, referred to by **WhenAvailable**, is running.
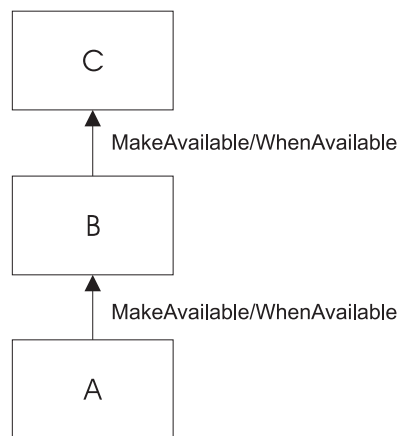
Figure 1. Example of Start Dependencies

When SA OS/390 receives a request to start A, the following chain of events will occur:
1. The request is propagated:
   a. Since A can only be started when B is running, a start request is put to B.
   b. Since B can only be started when C is running, a start request is put to C.
2. In response to these requests, the desired state of all three resources is changed to AVAILABLE.
3. SA OS/390 tries to change the actual state of the resources according to their desired state:
   a. At first, only C, which has no start dependencies, can be started. B and A cannot be started because C and B are not yet running.
   b. Then B will be started, because C is now available.
   c. Finally, A is started.

The propagated requests are usually called *votes* instead of requests.

In example 1, the request propagation is uniform; the desired state of all three resources is set to AVAILABLE because the condition of the dependency relationships is **WhenAvailable** in both cases. This is not always the case, as the following example shows.

Example 2: Modify example 1 to the effect that B can only be started when C is *unavailable*, and that C is running, in accordance with its desired state AVAILABLE, when the request comes in.

To reflect this modification, the upper arrow label of Figure 1 on page 5 would have to be changed to **MakeAvailable/WhenDown**. This expresses that the dependent (lower) resource can only be started when the supporting (upper) resource is unavailable (down).

In example 2, the request must be transformed when propagated from B to C, because in order to start B and then A, C must be down. Therefore, SA OS/390 would put a *stop* request to C in this case, and the desired state of C would be set to UNAVAILABLE.

By propagating requests, SA OS/390 actively supports the start or stop request. You can also switch off request propagation for a resource. If this were to be done for resource A in example 1, then A would not be started because B is not available, and SA OS/390 would do nothing to start B. In this case A would only be started after B had been started, directly or indirectly, through another request.

Persistency of Requests and Conflicting Requests

Requests (and the votes derived from them) are persistent. They are stored in SA OS/390 and continue to be taken into account until you explicitly remove them. This implies that there can be more than one request (vote) for the same resource at the same time, and these requests (votes) can be contradictory, as shown in the following example.

Example 3: Expand example 1 by a resource D, also depending on C, which can only be started if C is down. A, B, and C are as in Figure 1 on page 5; D is supposed to be down, and its desired state to be UNAVAILABLE.

Figure 2 contains a graphical presentation of example 3.
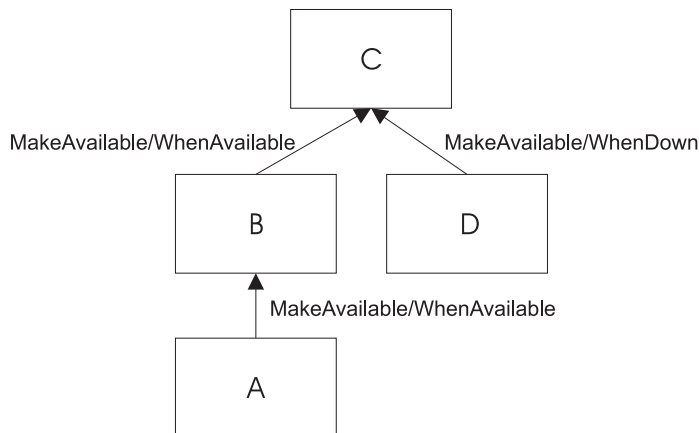
Figure 2. Example of Conflicting Requests

Now assume that first a request to start A and then a request to start D are passed to SA OS/390. The first request results in setting the desired state of C to AVAILABLE. Thereafter the propagation of the start request for D results in a vote to stop C. Since votes are persistent, the previous vote to start C is still existent,

and we have two contradictory votes for C. In such a situation, SA OS/390 uses the *priority* of the original requests to decide which one of the two votes wins.

When the priority of the old start vote for A is higher than that of the new vote to start D, then the desired state of D will be changed to AVAILABLE, but that of C will remain AVAILABLE; accordingly, SA OS/390 will not try to stop C, and thus D cannot be started. If, on the other hand, the vote to stop C has the higher priority, then the desired state of C is changed to UNAVAILABLE, and SA OS/390 will try to stop C in accordance with its desired state, and then to start D. When two contradictory votes have the same priority, a start vote wins over a stop vote.

The persistency concept implies that the losing vote is not automatically discarded. If, for instance, the start request for A wins, the start request for D and the propagated stop vote for C continue to be stored in SA OS/390, and can still be fulfilled after the request for A, and therefore also the start vote for C which was derived from it, have been removed by an operator. After the removal, SA OS/390 will determine the desired state of C again and will set it to UNAVAILABLE in response to the stop vote propagated from the start request for D, if no other vote is pending for C. After that, C will be stopped, and then D will be started.

Note that persistency of requests does not apply to successive requests of the same operator. In this case the second request will replace the earlier one.

Triggers

Triggers specify necessary conditions for starting or stopping an application; 'necessary' means that the application can only be started or stopped when the condition is satisfied. Triggers are defined independently of applications. In this way the same trigger can be associated with more than one application. Triggers are defined and linked to an application in the policy database.

The conditions contained in a trigger are either startup conditions or shutdown conditions; there can be more than one startup condition, and also more than one shutdown condition. When a trigger is associated with an application, the resources generated from this application can only be started if *at least one* of the startup conditions in this trigger is satisfied; analogously, they can only be stopped if at least one of the shutdown conditions is fulfilled.

A trigger condition consists of a set of *events*. An SA OS/390 event represents an external event that is not under control of SA OS/390, but is relevant to the state of the application associated with the trigger. The information that the external event has or has not occurred is passed to SA OS/390 by *setting* or *unsetting* the SA OS/390 event; this must be done by an operator or by an automation procedure. A trigger condition is only satisfied when *all* its events are set.

The following example illustrates the use of triggers and their interrelations with dependencies and request propagation.

Example 4: Expand example 1 to the effect that resource C is associated with a trigger that contains only one startup condition. This condition consists of two events, EVENT1 and EVENT2. EVENT1 is set, EVENT2 is unset.

When the request to start A arrives at SA OS/390, it will set off the same sequence of events as with example 1 up to step 2 on page 5. Since, however, the only startup condition of the trigger is not satisfied, C will not be started, and therefore

B and A will not be started either. In order to start A, EVENT2 must be set, for example, by an operator. This will lead to a re-evaluation of the startup condition. Since this condition is now satisfied, SA OS/390 will start C, and subsequently B and A.

Service Periods

So far we have always assumed that the start or stop requests are made by a human operator. However, SA OS/390 also provides the possibility to make start and stop requests at specified points in time independently of human intervention. The objects that are able to do this are called *service periods*. Service periods are defined in the policy database.

A service period is a set of time intervals, so-called *service windows*, during which an application should be available or unavailable. Service periods are defined independently of applications and can then be associated with one or more applications or application groups (see "Application Groups"). When an application is associated with a service period, the service period makes a start request for the application whenever the start time of a service window arrives; this request is canceled when the stop time of the service window arrives. You can also specify service windows during which the application should be unavailable; in this case, a stop request is made at the start, and canceled at the stop time of the service window. The following example is again an expansion of example 1.

Example 5: Resource A of example 1 is associated with a service period that contains at least one service window during which A should be available.

If the start time of this service window arrives, the same sequence of events will occur as with example 1.

An operator can temporarily modify a service period (this is called a *schedule override*). In case of a conflict between a request made by an operator and a request from a service period, the operator request wins when its priority is not lower than that of the service period request.

Application Groups

Modern applications often consist of more than one component, and these different components can be distributed among different systems. SA OS/390 provides the possibility to combine different components of an application on one or more systems within a sysplex into an *application group*. This allows you to start and stop a complex application by a single command, and to integrate it into automation processes as a whole.

Example 6: Suppose that resource B of example 1 is an application *group* with the members B1 and B2, and declare A dependent on group B (not on the individual group members), and B dependent on C. You can define B so that every request made to the group as a whole is automatically propagated to every group member.

Figure 3 on page 9 contains a graphical presentation of example 6.
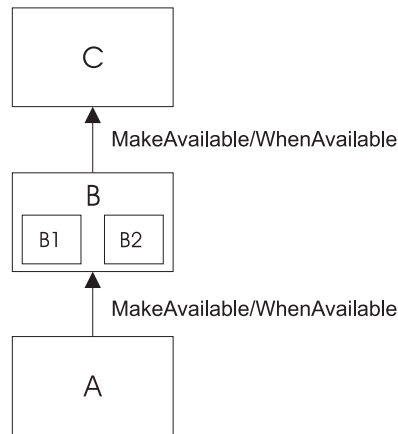
Figure 3. Example of a Request Involving a Group

Then, if you request A to be started, SA OS/390 will first, as before, propagate the request to group B and to application C. After C has been started and therefore group B can be started (step 3b on page 5 of example 1), a start vote will be propagated to every member of B. After the desired state of B1 and B2 has been set to AVAILABLE and both resources have been started, B will be considered available, and only then will SA OS/390 start A.

In this type of group (which is called BASIC) the group members form a complex entity, and therefore the group is only considered available when *all* its members are available.

The group concept is also used to move applications from their primary system to a backup system when the primary system has failed (group type MOVE). In this case the members of the group are instances of the same application on different systems. In accordance with their purpose, MOVE groups are declared available when *exactly one* of their members is available. You assign preferences to the elements in order to determine which group member is to be started when a start request is put to the group, and which group member takes over when the currently available member is not restartable any more.

SERVER groups are a third type of group. They are a variant of move groups and differ from these mainly in that you can specify how many of its members must be available before the group is considered available. As with move groups, you assign preferences to the members to determine which of them are to be started when a start request is put to the group, and which group members takes over when one of the currently available members is no longer restartable.

Groups can be nested. Suppose, for example, that you have a complex application that you want to be able to move from one system to another. Here you can first define two basic groups G1 and G2, each containing the application on a different system, and then define a move group that contains G1 and G2 as its members.

SA OS/390 and the NetView Message Automation Table

The implementation of SA OS/390 is based on NetView®. One important area, where SA OS/390 relies on NetView functionality, is the NetView Message Automation Table (MAT). This table serves to automate operator responses to messages that are sent to NetView. It contains instructions of the general form:

When message ABC arrives then issue command XYZ.

Whenever NetView receives a message, it scans the MAT. If it finds an entry for the message, it issues the command specified in that entry.

With applications controlled by SA OS/390, the command will typically be one of the generic routines that are shipped with SA OS/390 (see *System Automation for OS/390 Programmer's Reference*). Many of these routines retrieve information from the ACF and then act according to that information.

A typical example for such information is the MESSAGES/USER DATA policy item of the APPLICATION policy object. Within the MESSAGES/USER DATA policy item, you can associate a command with a message ID (see *System Automation for OS/390 Defining Automation Policy*). If you connect this message ID with the generic routine ISSUECMD in the MAT, then NetView will execute ISSUECMD when the application sends the message in question to NetView. ISSUECMD, in its turn, will search for the message ID in the ACF entry for this application, and if the message ID is associated there with a command, it will issue this command. For more information on ISSUECMD, see *System Automation for OS/390 Programmer's Reference*.

For example, you could associate the message ID `AHL031I`, which is the ID of the startup message sent by the application GTF, with the command `MVS $DMRO'GTF IS NOW UP'` in the MESSAGES/USER DATA policy item for GTF. Then the MAT would have to contain an entry like the following:

```
IF MSGID = 'AHL031I'
THEN EXEC(CMD('ISSUECMD AUTOTYP=START') ROUTE(ONE *));
```

Now, when NetView receives the AHL031I message it extracts the job name from the message and calls ISSUECMD. ISSUECMD knows where to find the job name and searches the ACF for the associated application. When it finds GTF, it will look for the AHL031I entry in the MESSAGES/USER DATA policy item and will issue the command that is associated with AHL031I for GTF,

```
MVS $DMRO'GTF IS NOW UP'.
```

For more information on the MAT, see *Tivoli NetView for OS/390 Automation Guide*. IMS Automation also has some special generic routines, see Chapter 6, "Common Routines" on page 125.

Chapter 2. Special Functions of IMS Automation

IMS Automation is integrated into SA OS/390. Thus, IMS regions must be defined in the policy database as subsystems by linking IMS applications to systems in order to be available to IMS Automation. Triggers and service periods for IMS regions are also defined as for any other application. But IMS Automation also offers some special facilities.

Recovery of Application Components

You can automate transaction, program, OLDS, and MSC link recovery globally and for individual components. This is achieved by combining basic functions of the product with IMS-specific policy items and several IMS-specific reserved MESSAGES/USER DATA keywords.

Program-to-Program Interface

NetView's program-to-program interface (PPI) provides the ability to communicate between a NetView application and other address spaces on the same host, such as IMS. The PPI enables NetView automation to include cooperative execution of commands in IMS and command processors in NetView.

IMS Automation uses the NetView PPI, among other things, to send IMS messages to NetView that drive resulting command lists or command processors.

AOI Exit

IMS Automation uses IMS's automation operator exit, DFSAOUE0, to trap messages destined for the IMS MTO operator. These messages can then be routed to NetView, suppressed from the MTO console, or sent to the MVS™ system console.

To accomplish this, IMS Automation supplies a program which is linked into the IMS nucleus. This program is supplied with an alias of DFSAOUE0, so that the IMS system gen can be run and then, by simply pointing at the right library, the IMS Automation version of DFSAOUE0 is linked into the nucleus.

Control over which messages to suppress, route, or send to the console is provided through an assembled table. The exit program loads the table during initialization. This table is contained in the IMS component of the PPI, which is named EVISPINM. The table can be modified by the user. IMS messages, that are exposed to automation using the EVISPINM parameter AUTO=YES, appear in the NetView log with a HDRMTYPE of #.

Many IMS systems already have a DFSAOUE0 exit. IMS Automation's exit has been written to call another exit. By default, the IMS Automation exit will call a program named EVIAOUE0. The EVIAOUE0 exit does not have to be linked into the nucleus, only into an IMS library accessible to the control region. The name of the program that the IMS Automation exit calls can be changed by altering EVISPINM (EXITNAME parameter of the INITIAL type).

For more information on EVISPINM, see "Step 1: Extend EVISPINM Member, If Required" on page 30.

FDR Environment

This section describes automation functions applicable to FDR-enabled IMS environments.

Note: FDR provides superior, sysplex aware automated recovery within IMS. It is the recommended solution for high availability of IMS applications.

The automation functions provide cross-system support for IMS and FDR startup, and cross-system automation of the messages associated with IMS failure and subsequent FDR recovery. This is achieved by invoking System Automation for OS/390 sysplex services.

Recovery Capability: FDR IMS Automation implementation will support automatic response to the IMS DFS4167A WTOR message when the IMS I/O prevention completed message AVM006I is received during recovery processing.

Functional Overview

The following sections provide an overview of the FDR region startup, shutdown and recovery functions provided by IMS Automation.

Startup Overview

It is recommended to automate startup by defining relationships as described in "Defining an FDR Environment" on page 40.

Otherwise, the following mechanism must be used: During IMS startup, when the control region has completed initialization, the IMS subsystem will be put to "UP" status. Message DFS4190I is trapped to indicate if FDR is enabled. If IMS is FDR enabled, automation will initiate a startup of the FDR region in the NetView domain as defined in the policy database.

Recovery Overview

IMS Automation provides an automated response to allow FDR recovery processing to proceed. Assume that IMS is running on system SYS1 and FDR on system SYS2. The sequence of events is as follows:

1. When IMS on system SYS1 abends, WTOR message DFS4167A is issued by the FDR region on SYS2 indicating that it is waiting for the failing IMS region to complete I/O Prevention.
2. When the failing IMS on system SYS1 completes I/O Prevention, message AVM006E is issued by the Availability Manager.
3. Automation then replies 'UNLOCK' to the WTOR message DSF4167A, to allow FDR on SYS2 to commence recovery processing.

Shutdown Overview

Automation then replies 'UNLOCK' to the WTOR message DSF4167A, to allow FDR on SYS2 to commence recovery processing.

XRF Environment

This section describes automation functions applicable to XRF IMS configurations.

Note: XRF is an old technology. It is recommended to utilize the sysplex capable FDR solution to maximize the availability of IMS applications.

Automation functions specific to the XRF IMS environment include a two-CPC approach and surveillance/takeover capability.

Two-CPC Approach: The XRF IMS Automation implementation goes beyond the intra-CPC, message-driven, passive approach to automation and focuses on a two-CPC approach, using active monitoring logic to communicate status information between the two NetViews in an XRF environment. This exchange of information between the two CPCs enables NetView-based CLISTs and REXX procedures to complement the automation functions of XRF during takeover. Furthermore, a key function included in the automated application is a flexible scheme for automatically synchronizing IMS regions between the *active* and the *alternate*, for startup, shutdown and takeover operations.

Surveillance/Takeover Capability: Facilities are provided in an XRF environment to forcibly terminate a failing *active* XRF partner, automatically respond to the IMS "I/O Prevention Completed" message, cancel DBRC on the old *active*, and automatically start the *alternate* (new *active*) XRF partner after takeover. IMS dependent message region synchronization is maintained between an active IMS subsystem and an alternate IMS subsystem in an XRF environment. IMS subsystem failures are detected. The IMS control region, DBRC address space, DL/I address space, and IRLM address space are automatically recovered.

Another key point is that the automation application supports a two-CPC implementation, using a NetView-to-NetView task (NNT) session to communicate between the *active* CPC and the *alternate* CPC.
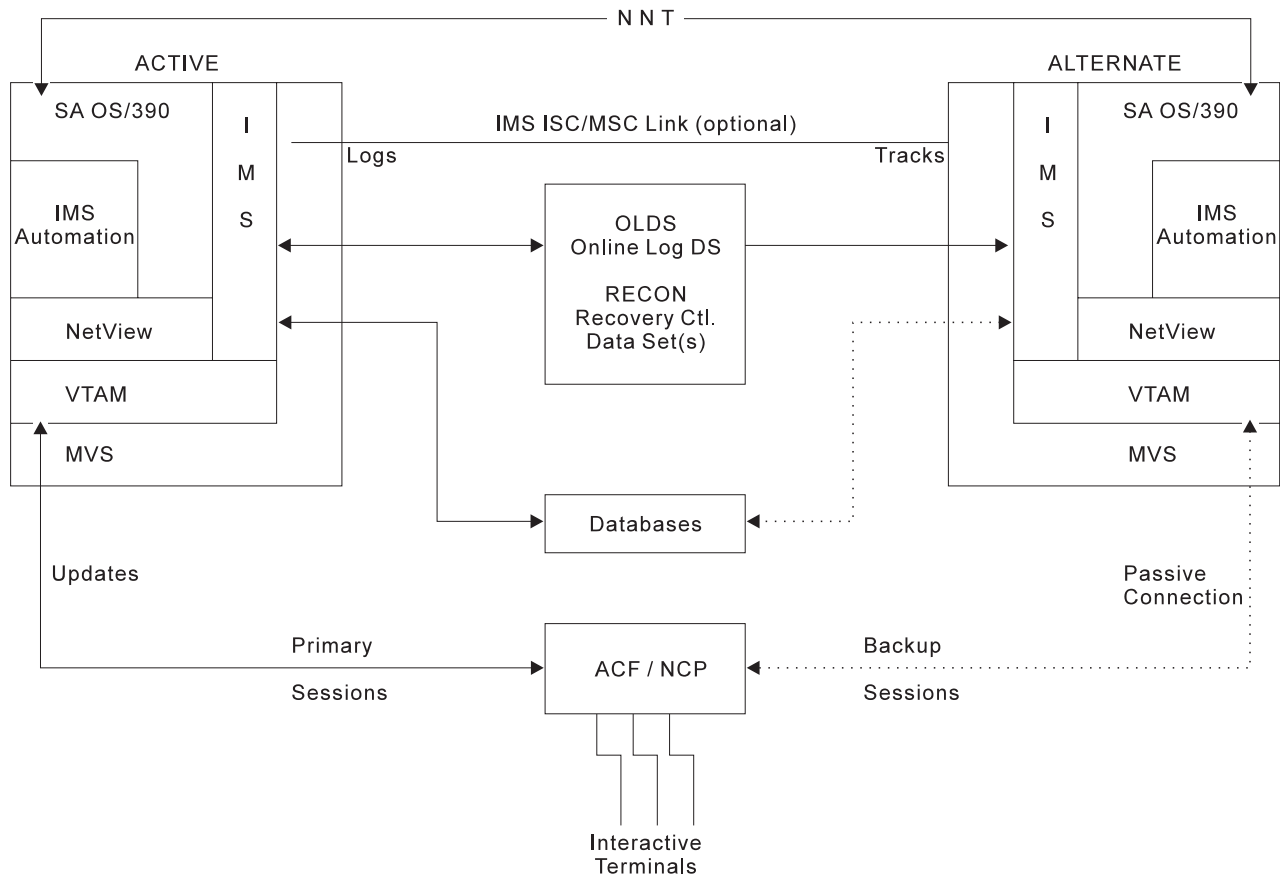
Figure 4. IMS Automation in a NetView Environment

Stopping and Recovering Regions in an XRF environment

Stopping a Region. The automation for shutting down the XRF complex focuses on allowing both the *active* and the *alternate* to be shut down in a synchronized manner.

The XRF shutdown function implemented in the sample provided with IMS Automation allows the operator to stop an active IMS subsystem and its alternate IMS subsystem. In an XRF environment, it is not possible to stop the active IMS subsystem without also stopping the alternate IMS subsystem. The alternate IMS subsystem could be stopped on its own.

During shutdown, the alternate IMS subsystem is terminated first. When the *alternate* shutdown is complete, its NetView will communicate to the *active's* NetView that the *alternate* has completed termination. Shutdown for the *active* will proceed.

Recovering a Region. In an XRF environment, recovery will be handled by XRF, if the active IMS subsystem fails and an alternate IMS subsystem has been started, with the automation application supplementing the XRF function. If an *alternate* has not been started when an IMS failure occurs on the *active*, the *active* will be restarted by the automation application for both the IMS control region and the dependent regions.

If the automation application does not know whether an *alternate* was started, it will send a message to the operator. The situation of not knowing about the *alternate* will occur only if the NetView-to-NetView communication link has been lost. The message will inform the operator about the loss of NetView communication. The message will also indicate, if an *alternate* started, whether XRF will handle the recovery automatically or the operator should manually recover IMS.

Takeover Overview

ARM and XRF can be used alone or together to provide system recovery. ARM provides services to support an XRF environment by preventing conflicting actions by ARM and XRF. XRF provides high availability for its users by providing an environment where an active production system has an alternate system that is ready to take over processing in the event of the active failing. If the alternate detects a failure of the active, then it initiates an XRF takeover.

IMS registers both the active and alternate XRF systems with ARM, but it only issues an associate for the alternate XRF system. For subsystem failures ARM restarts the alternate XRF system if it fails. It does not attempt, however, to restart the active XRF system if it fails. If the active XRF system fails, then the alternate XRF system takes over for it. Therefore, an ARM restart is not necessary.

XRF Takeover

A takeover condition is an event that causes an alternate IMS subsystem to request a takeover. In an installation that runs XRF, an IMS control region abend is always a takeover condition. When tailoring the IMS subsystem, the system programmer specifies whether any of the following failures are also takeover conditions:
- An MVS failure, loop, or wait state
- A CPC failure
- A VTAM® failure

- IRLM failure

Not all failures at an IMS installation can be takeover conditions. XRF does not address the outages caused by failures of service elements you do not duplicate. For instance, XRF does not respond to:

- A channel or link failure that causes a break in communication between the CPC and the communication controllers or DASD
- Failures in the telecommunication network, such as communication controllers, NCPs, lines, and terminals
- Inter-system failures, such as those caused by JES3 or CTCs
- Loss of or damage to the IMS databases
- A power failure that affects both CPCs in the XRF complex
- Failures of user catalogs that point to data sets, such as databases

One of the primary goals of IMS Automation is to eliminate, where possible, any human intervention during the takeover process for an XRF environment. Although XRF provides a very comprehensive approach to handling the workload transfer, a few exception conditions require operator interaction, and operational errors could cause severe problems. IMS Automation was designed to automate these actions and to eliminate the human element where possible.

With the subject of an XRF takeover, we are by definition talking about an "extra-system" automation environment and the need for communication between two CPCs. In the case of a takeover, the NetView-to-NetView session being used to pass information back and forth between the *active* and the *alternate* CPCs was judged to be inadequate, as it relies on having VTAM active. Because VTAM failure is one of the failure types that the redundant-resource design of XRF supports, IMS Automation was designed to provide support for a takeover without VTAM being operational.

Although the NetView-to-NetView session is used to check the dependent regions on a continuous basis, it is not used at all during the takeover.
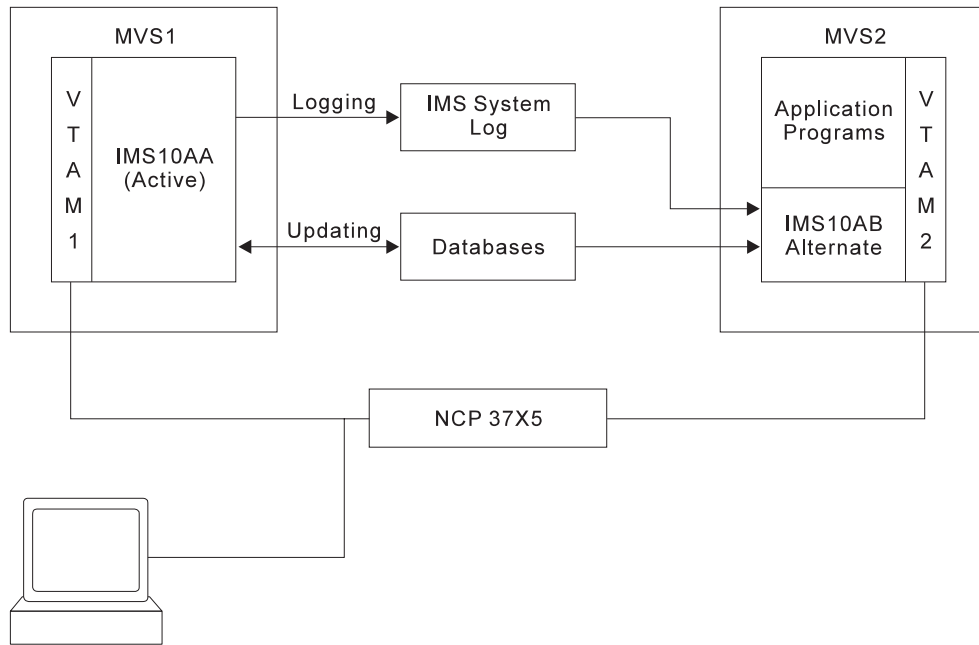
Figure 5. XRF Complex Before Takeover

In Figure 5, and Figures 6 and 7 on page 17, the XRF complex is shown before, during, and after takeover. Illustrated takeover activities are discussed below:

1. IMS10AA processes the high priority work that comes from the remote terminals. It updates the databases and also records its activity on the IMS system log.

2. IMS10AB tracks the *active* by monitoring the records on the IMS system log. It also opens backup sessions for Class 1 terminal users who log on to the *active*. To maintain an environment identical to that in the *active*, IMS10AB updates many control blocks and message queues in the *alternate* to reflect those in the *active*. Processor capacity and storage not used by this activity support the application programs.

3. When IMS10AA abends, the takeover begins. Depending on XRF's demand for real storage, MVS2 might swap out the application programs. IMS10AB shifts the production workload to itself and begins to serve Class 1 and Class 2 terminals. Problem determination activities can begin on the failing IMS10AA.

4. While IMS10AB recovers data and tells NCP to switch sessions on Class 1 terminals, IMS10AB and MVS1 prevent IMS10AA from writing to the IMS system log and the databases. IMS10AB isolates the log and proceeds with the takeover. At the same time, MVS1 performs I/O prevention, it ensures that all new I/O requests to the databases from IMS10AA return without being executed. When MVS has completed or cancelled all existing I/O requests to the databases, it notifies the operator.

5. The takeover is complete when all the users at Class 1 terminals can communicate with IMS10AB and can enter transactions and receive replies from their IMS applications. When IMS10AB learns that the failing IMS10AA cannot write to the databases, it stops protecting them.
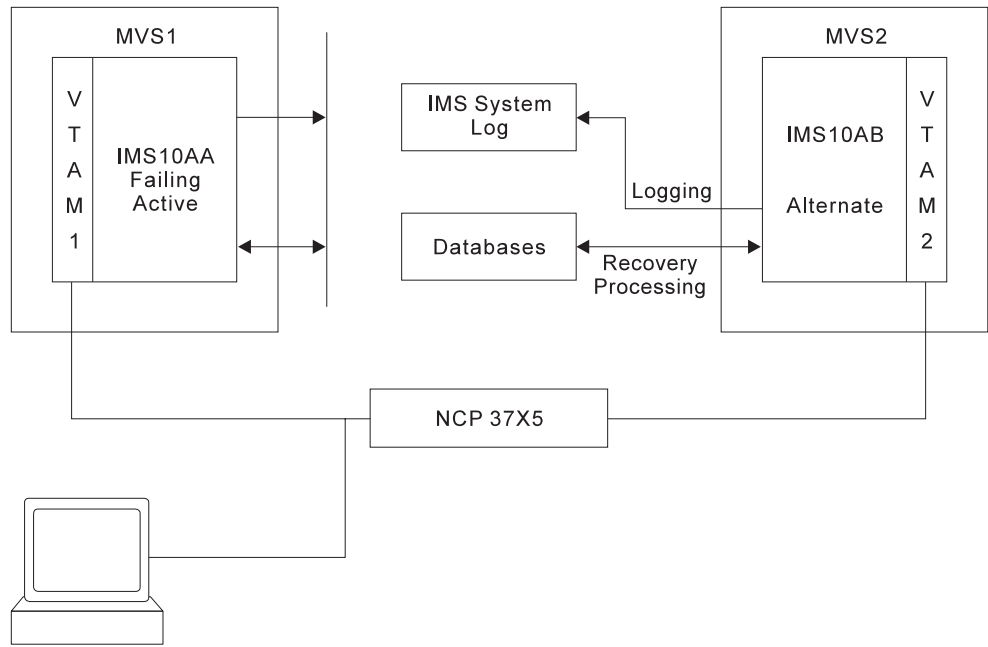
Figure 6. XRF Complex During Takeover
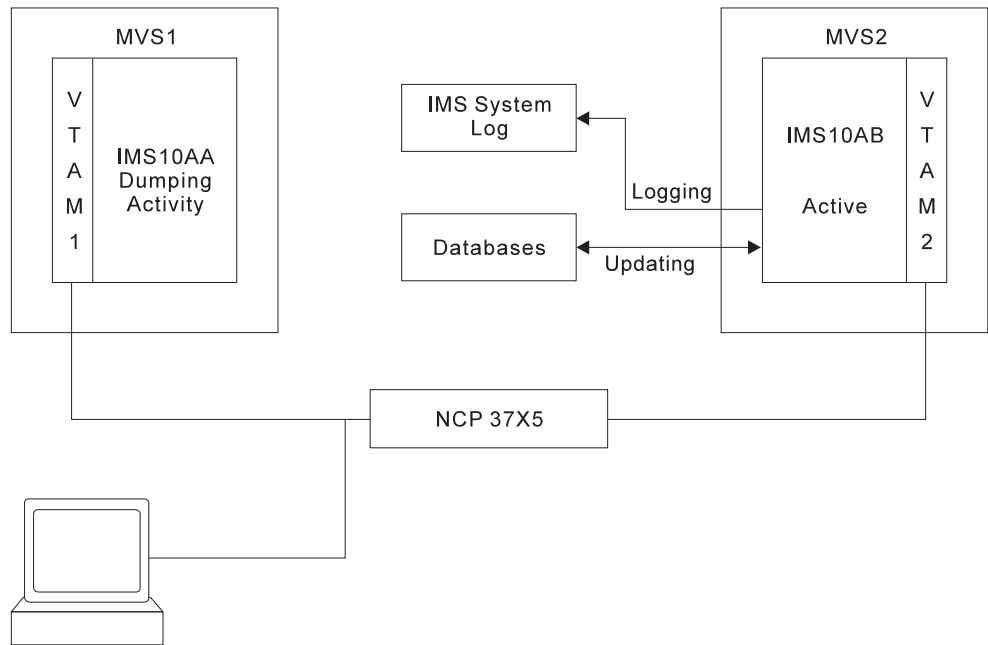
Figure 7. XRF Complex After Takeover

XRF and ARM Takeover

The following cases tell how XRF and ARM takeovers work.

XRF Active Fails While XRF Alternate Is Available: When an XRF pair are both running and the active subsystem (IMS) fails, an XRF takeover occurs. ARM does not attempt to restart the failing active subsystem.

ARM Recovers from Failure of XRF Alternate Subsystem: When an XRF alternate subsystem fails, ARM restarts that subsystem on the same system.

Entire System Fails: When an XRF pair are both running on a single system and the entire MVS system fails (both the IMS *active* and *alternate* fail), then ARM restarts an IMS on another system identified by the ARM element group. This IMS must be identified by the ARM element group defined as an XRF partner associated with the IMS that failed.

ARM Recovery of Failure of XRF Alternate When Entire System Fails: When the system where the XRF alternate is running fails, ARM restarts an IMS on another system identified by the ARM element group. This IMS must be identified by the ARM element group defined as an XRF partner associated with the IMS that failed.

Resuming Operations after a Takeover

After a takeover the control region that was previously the active system can be restarted. If ACTCODES (see "ABCODES/ACTCODES/ALTCODES—Restart control region after abend" on page 66) is not used to automatically restart the control region that is the subject of a takeover, it can be started manually through the operator interface panels.

Restarting a control region as the Alternate

The control region may be restarted via the operator interface panels in the usual way. Automation will detect that it must be started as the alternate system and responds to the DFS810A message with the correct response for the alternate control region.

Restoring the control region to Active status

To restore the control region to the active status, there are two methods:

- The control region can be restarted as the alternate as described above and then a planned takeover can be used to switch the active and alternate regions to their original states.
- A normal shutdown of the currently active system (the previous alternate) must be performed. Then the original active control region can be restarted using the BUILDQ start option. This will cause the message queues to be restored from the last /CHECKPOINT DUMPQ issued as part of the normal shutdown.

State/Action Tables

A major portion of automation involves driving actions based on *events* like error messages or operator actions. One way IMS Automation does event-driven automation is by associating a single message (an event) with a single automated response such as issuing a command, writing a message, or issuing an alert (the action). This is done in the MAT.

However, you may want different automated actions to occur in reaction to the same event under different conditions. For example, you may want different actions to be taken depending on whether the application is active or inactive when the event occurs. IMS Automation provides *state/action tables* for this type of automation. The state/action tables associate the event with the application's *state*, (active, stopping, down, and so on). Figure 8 on page 19 shows a sample state/action table.

```
 ┌─┐
 │1│ Event
 └─┘ ┐
     │
                 ┌─┐
                 │2│ State
                 └─┘ ┐
                     │
                         ┌─┐
                         │3│ Entry (action/new state)
                         └─┘ ┐
                             │
     /*********************▼************▼**********************************************/
     /* Status          →Init │    Down       Up          Stop        Discon  */
     /* State Value     →  0   │     1         2            3            4     */
     /********************▼**********************************************************/
     EVENT=DFS2160I       /2        NA        NA          /2           /2
  →  EVENT=DFS2161I     EVIECM01/3  NA    EVIECM01/3      NA           NA
     EVENT=DFS2168I     EVIECM01/2  NA    EVIECM01/2  EVIECM01/2   EVIECM01/2
     EVENT=DFS2169I     EVIECM01/4  NA    EVIECM01/4      /2           /2
     EVENT=DFS2142      EVIECM01/3  NA    EVIECM01/3      NA           NA
     EVENT=DFS2140        /3        NA        /3          NA           NA
     EVENT=DFS2236I       /3        NA        /3          NA           NA
```

Figure 8. Sample State/Action Table

The state/action table works like this:

1 **Event**

When an event occurs, IMS Automation finds the event in the state/action table. For example, when message DFS2161I is received, IMS Automation finds `EVENT=DFS2161I` in the state/action table.

2 **State**

IMS Automation checks the state of the application. For this example, the application is assumed to be in its initial state; the state value for the initial state is 0.

Notice that the other valid states in this sample table are down (state value 1), up (state value 2), stopping (state value 3), and disconnected (state value 4).

3 **Entry (action/new state)**

IMS Automation finds the associated entry. For this example, the entry is located in the same row as `EVENT=DFS216I` and the same column as the initial state (`INIT`). The entry IMS Automation finds is `EVIECM01/3`, which gives IMS Automation two pieces of information: the action to take (`EVIECM01`) and the new state value (3). Thus, IMS Automation calls the routine EVIECM01 and sets the state value for the application to 3, which is "stopping" in this example.

Note: `NA` signifies that this event/state combination cannot occur.

By changing the state of the application, IMS Automation records which events have occurred. By specifying an action to be performed along with the new state, IMS Automation compiles a complete package which:
- Maintains a history of events
- Identifies when to execute an action
- Identifies what action to execute.

Part 2. Customizing IMS Automation

This part describes the steps that are necessary to customize and set up IMS Automation. Furthermore, it contains reference sections for IMS-specific MESSAGES/USER DATA keywords and for common routines which request information or perform tasks associated with IMS Automation.

Important: IMS Automation 2.2 only supports IMS V6 and higher.

Chapter 3. Customizing IMS Automation

This section explains how to customize NetView, IMS and SA OS/390 for IMS Automation. The customization process mainly consists of defining the policy objects that are necessary for IMS Automation in the SA OS/390 policy database.

IMS Automation Definitions in NetView

You customize IMS Automation for your specific installation by modifying the policy database in the customization dialogs. For information on the customization dialogs, see *System Automation for OS/390 Defining Automation Policy*.

Step 1: Add the Automation Operators

Define the IMS Automation automation operators in the AUTO OPERATORS policy object (AOP entry type) according to the following table:

Automated Function	Operator ID	Message Classes
IMSMSTR	AUTIMS	EVI*
IMSWATCH	AUTSURV	
IMSPPI	AUTIPPI	

Note: Make sure that these operator IDs are defined in the DSIOPF member in the DSIPARM data set of NetView.

Step 2: Define IMS Regions

All IMS regions must be defined to SA OS/390 as APPLICATION objects in the customization dialogs; for these objects, the **Application Type** field must be set to IMS.

Important Note

Note that the subsystem names as specified in the **Subsystem Name** field of the **Define New Entry** panel must not exceed *eight* characters for applications of type IMS. This is in contrast with standard applications where 11 characters are allowed.

Applications of type IMS have the following IMS-specific policy items:
* IMS CONTROL

    This item must be defined for all region types. For details see *System Automation for OS/390 Defining Automation Policy*.
* RESOURCE THRESHOLDS

    This item specifies the thresholds for recovery of application components. For details see "Automating Recovery for Application Components" on page 48.
* STATE ACTION TABLE

    This item serves to link an application to a set of state/action tables. The state/action tables are used for component recovery. For more details see "How to Set Up the State/Action Tables" on page 54.

The following subsections contain special customization steps for different region types.

Substep 1: Code the Entries for Control Regions

When defining control regions, observe the following points:

- You must code thresholds in the standard THRESHOLDS policy item.
- If you will use service periods or triggers, link these to the control region under the SERVICE PERIOD or TRIGGER policy item.

In addition, specify the MESSAGES/USER DATA keywords shown in Table 3. For more information on IMS-specific MESSAGES/USER DATA keywords, see Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation" on page 57.

Table 3. Applicable MESSAGES/USER DATA Keywords for Control Regions

Required	Keyword	Comments	See Page
✔	ABCODES	Non-XRF abend codes.	66
	ACT CODES	XRF only.	66
	ALT CODES	XRF only.	66
	BRO	IMS control regions only. Code to issue the appropriate broadcast message prior to shutdown.	70
	CHE	XRF only.	71
	DFS3869A DFS3869I	XRF only. Specify exactly as indicated.	80
✔	DFS810A	IMS control regions only. Code for response to message DFS810A.	84
✔	DFS989I	DB control regions only. Code for response to message DFS989I.	86
✔	DFS994I	Code for actions after message DFS994I.	87
	FORCE	XRF only. Code to inform the old *active* that it is no longer the *active* when a predatory takeover occurs (the *alternate* assumes control).	90
	HOLDQ	Code to hold BMP initiators.	91
	IPS	XRF only. Code this command to adjust MVS processing if necessary during an XRF takeover.	93
	POSTCHKP	Code to reflect your installation's procedure to issue commands after a shutdown checkpoint has been issued.	97
	PRECHKP	Code to reflect your installation's procedure to issue commands prior to a shutdown checkpoint being issued.	99
	RELEASEQ	Code to release BMP initiators.	103

Required	Keyword	Comments	See Page
	RESTARTABORT	Code to reflect desired response to the receipt of messages DFS0618 and DFS166I.	104
✔	SHUTTYPES	None.	106
	SNAPQ	XRF only. Code to reflect your installation's procedure to issue the /SNAPQ command.	108
✔	STOPBMPREGION	IMS control regions only.	110
✔	STOPFPREGION	IMS control regions only.	112
✔	STOPREGION	IMS control regions only.	113
	TCO	None.	114
	TCOMEMBERS	None.	115
✔	TPABEND	Code exactly as shown with the appropriate subsystem identifier.	117
	UNLKAVM	XRF only. Code exactly as shown.	118
	UNLOCK	XRF and FDR only. Code to reflect your installation's procedure to issue the /UNLOCK command.	119
	VTAMTERMS	IMS control regions only. This entry is used in both an XRF IMS and a non-XRF IMS environment.	121
	$PI	XRF only. Code to reflect the draining of the initiator scheme for the IMS subsystem in an XRF environment.	122
	$SI	XRF only. Code to reflect the startup of the initiators after an XRF takeover.	123
	$TI	XRF only. Code to reflect the setup of the initiator scheme for the IMS subsystem in an XRF environment.	124

Substep 2: Code the Entries for DBRC/DLISAS Regions

When defining DBRC/DLISAS regions, observe the following points:

- Set the **External Startup** and **External Shutdown** fields in the AUTOMATION INFO policy item to ALWAYS.

Substep 3: Code the Entries for FDR Regions

When defining FDR regions, observe the following points:

- Code shutdown commands for the NORM and IMMED phases of the SHUTDOWN policy item.

- For recommendations concerning the dependency relationships for FDR regions, see "Defining an FDR Environment" on page 40.

Substep 4: Code the Entries for CQS Regions

When defining CQS regions, observe the following points:

- Code shutdown commands for the NORM and IMMED phases of the SHUTDOWN policy item.
- For recommendations concerning the dependency relationships between CQS and control region, see "Defining a CQS Complex" on page 42.

Substep 5: Code the Entries for Message Regions

When defining message regions, observe the following points:

- If you have set the **External Shutdown** field in the AUTOMATION INFO policy item to NEVER or INITIAL, you must code shutdown commands for all three phases of the SHUTDOWN policy item.
- If you have set the **External Startup** field in the AUTOMATION INFO policy item to NEVER or INITIAL, you must code a startup command in the STARTUP policy item.
- Code thresholds in the standard THRESHOLDS policy item.

Step 3: Code the Entries for OLDS

When configuring OLDS recovery, observe the following points:

- Define an entry with the name OLDS in the IMS-specific RESOURCE THRESHOLDS policy item. For details, see *System Automation for OS/390 Defining Automation Policy*.

In addition, specify the MESSAGES/USER DATA keywords shown in Table 4. For more information on IMS-specific MESSAGES/USER DATA keywords, see Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation" on page 57.

Table 4. Applicable MESSAGES/USER DATA Keywords for Online Data Sets (OLDS)

Required	Keyword	Comments	See Page
✔	OLDS	Code the names of OLDS to be kept as spares. IMS Automation starts the spares only when the number of available OLDS drops below the minimum needed.	94
✔	DFS3258A	Action to take on last OLDS.	79

Step 4: Code the Entries for MSC Links

When configuring MSC link recovery, observe the following points:

- Define entries with the name MSC (for all links) or MSC.*link_id* (for a single link) in the IMS-specific RESOURCE THRESHOLDS policy item.

In addition, specify the MESSAGES/USER DATA keywords shown in Table 5 on page 27. For more information on IMS-specific MESSAGES/USER DATA keywords, see Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation" on page 57.

Table 5. Applicable MESSAGES/USER DATA Keywords for MSC Links

Required	Keyword	Comments	See Page
✔	DFS2142	Code this entry to restart a logical link path.	76
✔	DFS2161I	Code this entry to restart a link after it has been stopped by an IMS system.	77
✔	DFS2169I	Code this entry to restart a link after disconnection of a Multiple Systems Coupling (MSC) link between two IMS systems.	78

Step 5: Code the Entries for RECONs Recovery

For RECON recovery, specify the MESSAGES/USER DATA keywords shown in Table 6. For more information on IMS-specific MESSAGES/USER DATA keywords, see Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation" on page 57.

Table 6. Applicable MESSAGES/USER DATA Keywords for RECONs

Required	Keyword	Comments	See Page
	RECONS	To turn on active monitoring for RECONs, code this entry. IMS Automation checks for spare RECONs at the interval you specify on the RECONS entry.	101

Step 6: Code the Entries for Transaction and Program Recovery

When configuring transaction or program recovery, observe the following points:

- Define entries with the name TRAN (for all transactions) or TRAN.*trans_id* (for a single transaction), and respectively PROG (for all programs) or PROG.*prog_id* (for a single program) in the IMS-specific RESOURCE THRESHOLDS policy item.

In addition, specify the MESSAGES/USER DATA keywords shown in Table 7. For more information on IMS-specific MESSAGES/USER DATA keywords, see Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation" on page 57.

Table 7. Applicable MESSAGES/USER DATA Keywords for Transaction and Program Recovery

Required	Keyword	Comments	See Page
✔	ABCODEPROG	Code this entry to specify system action in response to program abend codes.	64
✔	ABCODETRAN	Code this entry to specify system action in response to transaction abend codes.	68
✔	DFS554A	Code this entry to restart a transaction and program after an abend.	81

Step 7: Extended IMS Automation Definitions

If you will customize state/action tables, do the following:

1. Define a set of state/action tables under the STATE/ACTION TABLES policy object for IMS (ISA entry type).
2. Link the set to the subsystem under the STATE ACTION TABLE policy item of the APPLICATION object.

Step 8: Preparing IMS Automation to Manage an IMS XRF System in a Dual-CPC Environment

Perform the following step only if you are running an IMS subsystem that is XRF and using shared DASD within a dual-CPC environment. If your IMS configuration is XRF and the ACTIVE and ALTERNATE subsystems execute on separate CPCs then you must perform the following steps. An example for each of these steps follows:

1. Define additional TASK statements in the EVIDMN1 member, which is located in DSIPARM.
2. Define DSTINIT members, one for each DOMAIN, in the NetView DSIPARM data set. Use EVISTSM example in the ING.SINGSAMP data set.
3. Create IMS XRF STATUS FILE objects in the policy database.
4. Verify that SA OS/390 status file placement is in VSAM user catalogs that are accessible from the master catalogs of both CPCs.

Coding example

The following is an example for two NetView domains managing an IMS XRF system in a dual-CPC environment. The first domain is DOM01. The second domain is DOM02.

1. Define additional TASK statements in the EVIDMN1 member, located in DSIPARM. In domain DOM01 define the task that reads the status file for DOM02.

   ```
   TASK   MOD=DSIZDST,TSKID=AOFDOM02,MEM=STSDOM02,PRI=6,INIT=N
   ```

   In domain DOM02, define the task that will read the status file for DOM01.

   ```
   TASK   MOD=DSIZDST,TSKID=AOFDOM01,MEM=STSDOM01,PRI=6,INIT=N
   ```

2. Define DSTINIT members, one for each DOMAIN, in the NetView DSIPARM data set. When this is done, change the DSTINIT PDDNM value to match the unique DD name formed by concatenating "AOF" with the domain name. Use EVISTSM example in the ING.SINGSAMP data set.

   In domain DOM01, define the DSTINIT member called STSDOM02. The member name would be STSDOM02.

   ```
   **************************************************
   * DSTINIT MEMBER FOR IMS Automation XRF STATUS *
   * FILE DEFINES STATUS FILE IN OTHER DOMAIN     *
   **************************************************
   * COMMENT LINE
     DSTINIT PDDNM=AOFdom02
     DSTINIT XITVN=AOFSISTS
     DSTINIT FUNCT=VSAM
     DSTINIT DSRBO=1
     * END OF MEMBER
   ```

   In domain DOM02, define the DSTINIT member called STSDOM01.

   ```
   **************************************************
   * DSTINIT MEMBER FOR IMS Automation XRF STATUS *
   * FILE DEFINES STATUS FILE IN OTHER DOMAIN     *
   **************************************************
   * COMMENT LINE
   ```

```
 DSTINIT PDDNM=AOFdom01
 DSTINIT XITVN=AOFSISTS
 DSTINIT FUNCT=VSAM
 DSTINIT DSRBO=1
* END OF MEMBER
```

Ensure that the DSTINIT parameters start in the second column; this is a NetView requirement.

3. Create IMS XRF STATUS FILE objects in the policy database for both partners and link them to the respective systems in the IMS STATUS FILES policy items of the systems.

Specify the fully qualified name for the SA OS/390 status file in the **Partner status file** field.

4. The SA OS/390 sample INGALLC2 specifies the DEFINE CLUSTER control statement used to establish the SA OS/390 status file. The SHR(2) operand needs to be changed to SHR(3 4) and the status file reallocated in order to permit the sharing described in this step. Also, the new allocation must be on a shared volume accessible by both system images in the dual-CPC environment.

5. Use the VSAM ALIAS command to ensure that both status files are in USER catalogs that are accessible from each system's MASTER catalog.

Step 9: Defining NetView PPI Receiver task

A PPI receiver is required to allow IMS subsystems to communicate with NetView. This is a required step and must be done for correct automation of an IMS subsystem.

1. If you are using a policy database that has been converted from a previous release, use the migration function to migrate EVICFPPI into the policy database. This creates the subsystem classes for the PPI interface. Member EVICFPPI can be found in the SINGSAMP data set.

2. Define an application to represent the PPI receiver in a NetView. This application must be defined as NON-MVS and the job name MUST be EVINTASK. Specify the monitoring routine to be AOFATMON.

3. Link the application to the CLASS_IMS_NV_PPI class.

4. Do not specify any relationships, so that the application will be started as soon as possible.

5. Specify a Where Used APG that is connected to every system that runs an SA OS/390 NetView Agent. This ensures that this resource will be created for every SA OS/390 Agent in the sysplex. It is suggested that a SYSTEM APG that is linked to all the systems is used rather than a SYSPLEX APG. This allows the APG to be shutdown on a system by system basis.

Step 10: Defining IMS PPI Receiver task

A PPI receiver is required to allow NetView to communicate with IMS Control Region subsystems. This is a required step and must be done for correct automation of a IMS Control Region subsystem.

Note: An IMS PPI subsystem must be defined for every IMS Control Region subsystem to be automated.

1. If you are using a policy database that has been converted from a previous release, use the migration function to migrate EVICFPPI into the policy database. This creates the subsystem classes for the PPI interface. Member EVICFPPI can be found in the SINGSAMP data set.

2. Define an application to represent the PPI receiver in an IMS subsystem. This application must be defined as NON-MVS and the job name MUST be APPLID of the IMS subsystem. Specify the monitoring routine to be AOFAPMON.
3. Link the application to the CLASS_IMS_PPI class.
4. Specify the following relationships: hasParent(StartsMeAndStopsMe) → IMS subsystem. Relationship Type of hasParent and Condition of StartsMeAndStopsMe.
5. Specify a Where Used APG that the parent IMS subsystem is a member of.

Optional Additions to the PPI

NetView's program-to-program interface (PPI) provides the ability to communicate between a NetView application and other address spaces on the same host, such as IMS. For a detailed description of the NetView PPI, refer to the NetView documentation. The PPI makes it possible for NetViewautomation to have cooperative execution of commands in IMS and command processors in NetView.

IMS Automation uses the NetView PPI to:
- Initiate the execution of an IMS command from NetView
- Process a response from this IMS command
- Send IMS messages from IMS to NetView that drive resulting command lists or command processors

Both IMS and NetView have IMS Automation program-to-program interface components. The IMS Automation initialization member for IMS is EVISPINM. On the NetView side, EVINTASK is the initialization member. You can fully customize EVISPINM and, in some cases, you can change EVINTASK. Both PPI components use the ENQ/DEQ mechanism for conversation. This can be observed by an exclusive enqueue on the relevant resource involved. This is a normal situation for an SA OS/390-IMS PPI conversation and can be ignored.

This section describes the PPI initialization members, EVISPINM and EVINTASK, and explains how to change them to meet special system needs. The steps in this section are optional.

Step 1: Extend EVISPINM Member, If Required

EVISPINM is used on the IMS side of the program-to program interface to:
- Describe the search criteria to be used when processing messages in the automation operator exit and to determine whether the messages are to be sent to NetView, sent to the system console, or suppressed.

   Note: Messages specified in EVISPINM affect only MTO messages or the MTO copy of the message if it is also issued as a WTO.

   Thus, if IMS puts a message out as a WTO, the parameters specified in EVISPINM have no bearing on this. For further information, see "Automated Operator Exit Routine (DFSAOUE0)", in the *IMS/ESA Customization Guide*.
- Set the program-to-program interface buffer queue limit. This is the number of outstanding buffers that can be stored in the receiver buffer queue.
- Define the program-to-program interface receiver identifier for the EVISNPPI NetView subtask program.
- Define a unique job name for the IMS Automation BMP job if you do not want to use the default, PPI*imsid*.

- Define the default return code from the automated operator exit.
- Use the IMS ID to define an EVII*ims_id* member to generate a unique table for an IMS. You use this entry only if you want to define a separate table for each IMS instead of using one table for all.
- Define the transaction ID that the routine AOIEXIT uses to send messages to the BMP.
- If you do not want to use the default name, EVIAOUE0, define the name of a user-written AOI exit that IMS Automation should call instead.
- Describe the relationship between function names and IMS transaction names.

If any changes are required to the default EVISPINM, do the following:

1. Copy EVISPINM (located in ING.SINGSAMP) into a new member named EVII*ims_id*.
2. Edit EVII*ims_id* and:
   a. Update the IMSID.
   b. Change the RECEIVERID= name, if required. This value must be the same as the value defined in "Step 2: Extend EVINTASK, If Required" on page 34.
   c. Change the PPIPROC name, if required. For example, if you changed the name of the procedure to start the BMP, you must also change the name in the corresponding IMS program-to-program interface initialization member (EVISPINM or EVII*ims_id*).
   d. Add your own entries, if required.
3. Use the EVISJ020 sample JCL to assemble the program-to-program interface initialization member EVII*ims_id*.
4. Place the assembled member into a library accessible by both the control region and the PPI proc, typically IMS RESLIB.

Figure 9 shows a sample of the information contained in the EVISPINM member.

```
EVIMPINM TYPE=INITIAL,        INITIAL ENTRY
      IMSID=ims_id            MUST MATCH IMSID SPECIFIED
      BUFFQL=15,              BUFFER QUEUE LIMIT
      RECEIVERID=NETVIPPI,    NETVIEW RECEIVER IDENTIFICATION
      DEFAULTRC=CANCEL,       CANCEL (12) OR IGNORE (04)
      TRANID=EVITPPI1         TRANSACTION NAME
      SENDERID=applid         IMS SENDER IDENTIFICATION
      PPIPROC=name            MESSAGE-DRIVEN BMP PROC NAME
      EXITNAME=AIOEXIT        USER-DEFINED AOI EXIT MODULE

EVIMPINM TYPE=SEARCH,MSGID=DFS551I,SUP=NO,AUTO=YES,
EVIMPINM TYPE=SEARCH,MSGID=DFS552I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS0414I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS2161I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS2168I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS2169I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS554A,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS994I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=D,SUP=YES,AUTO=NO

EVIMPINM TYPE=ENTRY,       DEFINE A FUNCTION
      FUNCTION=IMSCMD,      FUNCTION NAME
      TRANSID=IMS          TRANSACTION NAME

EVIMPINM TYPE=FINAL        REQUIRED END
```

Figure 9. EVISPINM PPI Initialization Member

Each TYPE= keyword has a set of parameters you can customize. The parameters in EVISPINM are as follows:

TYPE=INITIAL
> Must be the first EVIMPINM specification.
>
> Optionally, the IMS buffer queue limit and the NetView receiver identifier can be specified. The valid parameters for TYPE=INITIAL are:
>
> **IMSID=**
> > Specifies the IMS ID of the procedure to be controlled by the processing table.
> >
> > This option will allow unique processing tables for different IMS control regions in which the control regions do not have unique data sets for every IMS procedure.
> >
> > If omitted, the name of this load module will have the default name of EVISPINM.
> >
> > If specified, the name of this load module will be EVII*ims_id*
>
> **BUFFQL=**
> > Specifies the IMS buffer queue limit for commands sent from NetView to IMS using the PPI.
> >
> > If omitted, a default of 3 is assumed.
> >
> > Minimum value is 1, maximum value is 999.
>
> **RECEIVERID=**
> > Specifies the NetView receiver identifier. If omitted, the NETVIPPI is assumed.
>
> **DEFAULTRC=**
> > Specifies the return code that the AOI exit should use when a message is not found in the TYPE=SEARCH entries.
> >
> > If CANCEL is specified, the AOI exit routine will stop IMS from sending additional message segments.
> >
> > Specifying CANCEL provides the best performance.
> >
> > If IGNORE is specified, the IMS Automation exit will receive additional segments, but will not act on the messages.
> >
> > IGNORE is provided for situations where other AOI exits need to see the messages even when the IMS Automation exit does not need to process the message.
> >
> > If omitted, DEFAULTRC=CANCEL is assumed.
>
> **TRANID=**
> > Specifies the IMS transaction identifier to queue messages from the automated operator exit.
> >
> > If omitted, EVITPPI1 is the default.
>
> **SENDERID=**
> > Specifies the IMS sender identifier.
> >
> > **Attention:** The sender ID must be the IMS applid.
> >
> > If omitted, the applid is assumed.

PPIPROC=
Specifies the name of the procedure to be started as a message-driven BMP.

This task will handle the transaction identified in TRANID=*keyword*.

If omitted, PPI*ims_id* is the default used.

EXITNAME=
Specifies the name of the user-defined load module that the IMS Automation AOI exit should call.

The exit must follow the IMS DFSAOUE0 coding requirements.

EXITNAME is not required.

If omitted, EXITNAME will default to EVIAOUE0.

TYPE=SEARCH
Describes the search criteria and action to be performed on a message routed to the master terminal operator. The valid parameters for TYPE=SEARCH are:

MSGID=
Specifies the message identifier to search for.

SUP=
Specifies whether to suppress the message if it is matched (YES), or not to suppress the message if it is matched (NO). If omitted, a default value is assumed (NO).

AUTO=
Specifies whether to automate the message by sending it to NetView (YES) or not to automate the message (NO). If omitted, a default value is assumed (YES).

IMS messages, that are exposed to automation using the EVISPINM parameter AUTO=YES, appear in the NetView log with a HDRMTYPE of #.

TEXT=
Specifies the text string to look for in the incoming message.

WTO=
Specifies whether to write the message to the system console (YES) or not to write the message (NO). If omitted, a default value is assumed, (NO).

TYPE=ENTRY
Relates function name to an IMS command name.

The TYPE default is ENTRY.

FUNCTION=
Specifies the name of a function

A request for this function causes the related IMS transaction to be executed.

TRANSID=
Specifies the IMS transaction name related to the function specification.

TYPE=FINAL
Generates the initialization table. The TYPE=FINAL keyword is required and must be the last TYPE= keyword in the EVISPINM member.

Step 2: Extend EVINTASK, If Required

EVINTASK is the IMS Automation PPI initialization member on NetView.
EVINTASK is used on the NetView side of the PPI to:

1. Set the program-to-program interface buffer queue limit. This is the number of outstanding buffers that can be stored in the receiver buffer queue.

2. Define receiver programs and automation operator tasks to be used for specific functions.

3. Define the PPI receiver ID for the EVISNPPI NetView subtask program.

Figure 10 shows an example of the information contained in EVINTASK :

```
BUFFQL=20
*
RECEIVERID=NETVIPPI
*
SERVER=SEND,MESSAGE,AUTIPPI,EVISNMSG
SERVER=RESPONSE,IMSCMD,AUTIPPI,EVISNRSP
SERVER=RESPONSE,NACK,AUTIPPI,EVISNACK
SERVER=REQUEST,NACK,AUTIPPI,EVISNACK
*
```

Figure 10. EVINTASK PPI Initialization Member

At least one valid SERVER= entry must be specified in EVINTASK. There must also be a SERVER= entry for each function that uses the IMS Automation PPI.

Usually, you will not change the EVINTASK PPI initialization member. However, you must change EVINTASK under these circumstances:

- If you must use a naming convention other than the default names for the automated operator tasks. (The default operator ID IMS Automation uses is AUTIPPI). It is recommended that you use the default names if possible.

- If your MVS system has both a test NetView and a production NetView running on the same host. Each NetView requires its own receiver ID. The receiver IDs are defined in both EVINTASK and in the corresponding EVISPINM or EVII*ims_id*.

Attention: Take care if you must change the operator IDs for the automated operator tasks or the receiver IDs. Errors when changing these can stop IMS Automation from working.

The keywords in the EVINTASK member are as follows:

BUFFQL=
PPI buffer queue limit for EVINTASK. Unless you have received a buffer error, do not change this value. To change the limit, specify a 2- or 3-digit number between 10 and 999. If no BUFFQL parameter is coded, the default used is 15.

For further information, refer to the description of buffer queue limit in the *NetView Programming: Program-to-Program Interface* documentation.

SERVER=
Do not change the SERVER= parameters (with the possible exception of the operator ID, AUTIPPI). The SERVER= parameters define:

1. The first parameter defines whether this function is a REQUEST, RESPONSE, or SEND. A REQUEST is used to identify a receiver program

to be invoked if NetView gets a CONVERSE or SEND from IMS. A RESPONSE is used to identify a sender program to be invoked if IMS sends a RESPONSE. The function name cannot begin with the prefix EVI. Do not change this parameter.

2. The second parameter defines the operator ID under which the program runs. The default name is AUTIPPI. Unless the naming convention at your site will not allow it, use the default name. To change the name of the default autotask, change the name of the IMSPPI automated function under the AUTO OPERATORS policy object (AOP entry type). Change AUTIPPI in the SERVER= entry to match the operator ID you define for the IMSPPI automated function.

   Note: If you change the operator ID for any automation operator, make sure that the new ID is correctly defined in DSIOPF member. Use the AUTPPI operator ID as an example.

3. The command list or command processor used for this function, such as EVISNRSP (the common response handler). Do not change this parameter.

RECEIVERID=
The PPI receiver ID that NetView uses to sign on to the PPI. If omitted, NETVIPPI is assumed. Do not change the receiver ID unless necessary. If you run both a test NetView and a production NetView within the same MVS system, you must give each NetView its own receiver ID and change this field to match. In addition, you must change the RECEIVERID= field in the corresponding EVISPINM or EVII*ims_id* member. Figure 11 on page 36 shows an MVS system with both a test and production NetView (if you run both in the same MVS system).
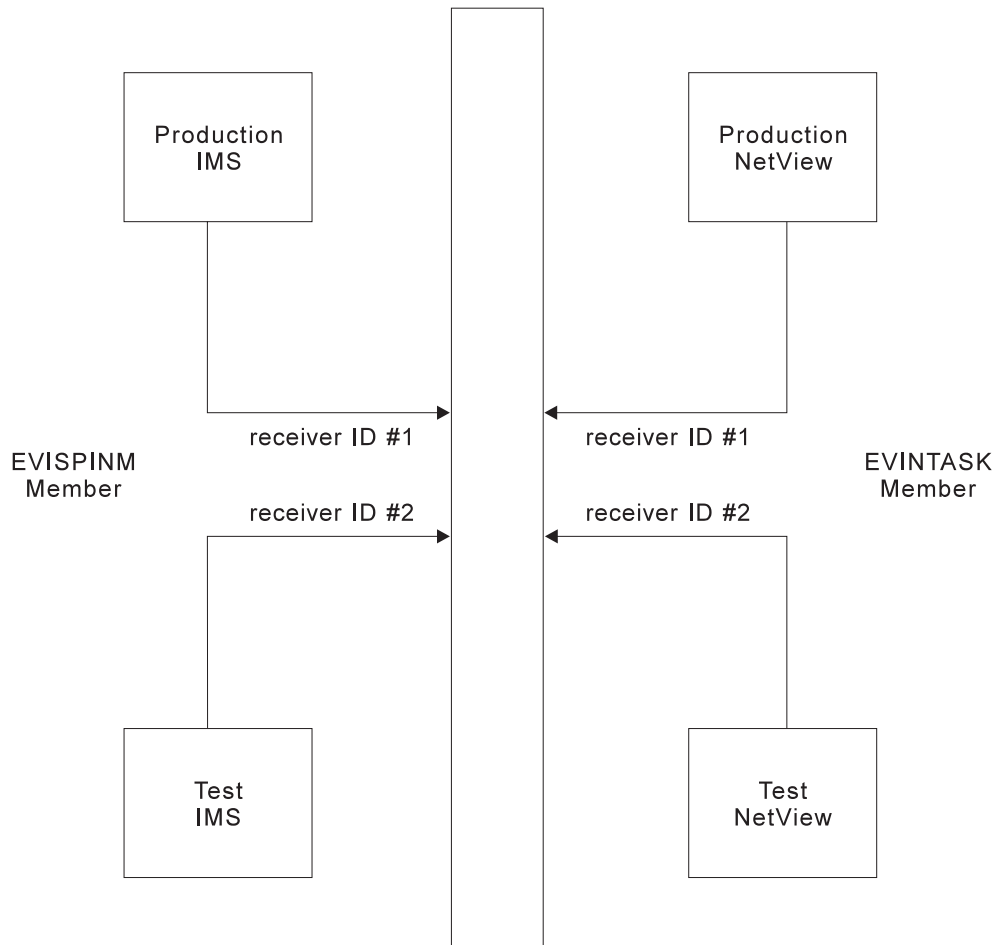
Figure 11. MVS System with Test and Production NetView. Define different receiver IDs for the test and production NetViews.

Migration and Coexistence

This section contains information on migration from IMS Automation Version 1 Release 4 to IMS Automation Version 2 Release 2, and on compatibility between Version 1 Release 4 and Version 2 Release 2.

If you migrate from IMS Automation 2.1 to 2.2, use the migration function as described in *System Automation for OS/390 Defining Automation Policy*.

Migration

The general migration process is described in *System Automation for OS/390 Defining Automation Policy*.

Most of the ACF keywords contained in Chapter 3 of the *AOC/MVS IMS Automation Programmer's Reference and Installation Guide* have been converted to keywords for the MESSAGES/USER DATA policy item of the APPLICATION objects; these are described in Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation" on page 57. For those of the remaining ACF keywords that are IMS specific or have an IMS-specific application, the following table provides the corresponding policy item of V2R2. Comments are added as required.

Table 8. IMS-Specific Correspondences between ACF Keywords and Policy Objects/Items

ACF keyword	Policy object/item	Comments
AREA	STATE/ACTIONTABLES object (ISA entry type)	These objects must be linked to applications through the STATE ACTION TABLE policy item of IMS applications.
AUTOOPS	AUTO OPERATORS policy object (AOP entry type)	The IMSOP*nn* auto operators are obsolete.
IMSCNTL	IMS CONTROL policy item for APPLICATION objects of type IMS	
IMSGROUP	—	Obsolete. Use standard application groups.
EXTCOND	—	Obsolete. Use standard triggers.
ENVIRON SETUP	—	The entry and the EVIEE001 exit are obsolete.
INITSTART	AUTOMATION FLAGS item of APPLICATION object	The EVIEIEXT exit is obsolete.
PRODUCT	STATE/ACTIONTABLES object (ISA entry type)	These objects must be linked to applications through the STATE ACTION TABLE policy item of IMS applications.
RECOVERY	MINOR RESOURCE FLAGS item of the APPLICATION object	For the names of the minor resources, see "Automating Recovery for Application Components" on page 48.
RESTART	AUTOMATION FLAGS item of APPLICATION object	The EVIEIEXT exit is obsolete.
SERVICE	—	Obsolete. Use standard service periods.
STSFILE	IMS XRF STATUS FILES policy object (ISF entry type)	
THRESHOLDS (IMS-specific thresholds in minor resource format)	IMS-specific RESOURCE THRESHOLDS policy item for application of type IMS	For the minor resource names, see "Automating Recovery for Application Components" on page 48.
TRIGGER	—	Obsolete. Use standard triggers.

Note that additional customization is necessary for the migrated policy database, as there are policy items that are required in V2R2, but have no counterpart in V1R4.

In particular, you must perform the following steps:
1. Define the startup commands for the individual IMS startup types in the STARTUP policy item of the IMS applications. For the valid startup types, see "Start" on page 163; for specifying startup commands in the STARTUP policy item, see *System Automation for OS/390 Defining Automation Policy*.
2. For an XRF environment, you must define application groups as described in "Defining an XRF Complex" on page 41, and events and triggers as in the XRFTRIG sample.

Coexistence between V1R4 and V2R2

Generally, you can control IMS regions (subsystems) running on an IMS Automation V1R4 target system from IMS Automation V2R2 (downward compatibility), and conversely IMS regions (subsystems) running on an IMS Automation V2R2 target system from IMS Automation V1R4 (upward compatibility). There are, however, some restrictions to this:

- Service periods and triggers.

  You can access the triggers and service periods of an IMS region running on an IMS Automation V1R4 target system from IMS Automation V2R2; in this case, the IMS-specific panels of IMS Automation V1R4 will be displayed. However, you cannot access an IMS region running on an IMS Automation V2R2 target system from the **Service Periods Functions** or the **Triggers List** panels of IMS Automation V1R4. Thus service periods and triggers are only downward compatible.

- XRF partners must have the same level, that is, they must either be both V2R2 or both V1R4. This also applies for FDR regions.

- You cannot manage message regions running on an IMS Automation V2R2 target system from the operator interface of IMS Automation V1R4 (option 6.3, **Perform Master Terminal Operator functions — Message Regions** no longer exists).

Chapter 4. How to Set Up the Special Functions of IMS Automation

This chapter explains how to set up the special functions of IMS Automation for your specific needs. For the setup of base functions, like starting and stopping subsystems, see the SA OS/390 documentation.

Defining the SDF States for IMS Automation

The Status Display Facility uses color to represent the various subsystem resource statuses, such as error, warning, action, or informational states. Typically, a subsystem shown in green on a Status Display Facility status panel indicates that it is up, whereas red indicates a stopped or problem state. For more information, see *System Automation for OS/390 Programmer's Reference*.

Priority, highlight level, and color definitions of the states are defined in the customization dialogs under a STATUS DETAILS policy object (SCR entry type). For IMS Automation, the following states must be present:

IMSTRAN	IMS transactions
IMSTIMR	IMS timers
IMSARCH	IMS archive problems
IMSMSCL	IMS MSC link recovery
IMSOLDS	IMS OLDS probems
IMSRECN	IMS RECON problems
CRITMSG	The default critical messages definition (IMS Critical Message)
CRITMSG	The default critical messages definition (IMS Critical Message)
CRITMSGA	Messages ending in A (IMS Critical Message)
CRITMSGE	Messages ending in E (IMS Critical Message)
CRITMSGW	Messages ending in W (IMS Critical Message)
CRITMSGI	Messages ending in I (IMS Critical Message)

Each of these categories except the suffixed **CRITMSG***x* keywords corresponds to an item on the **IMS Monitor Panel** (see Chapter 10, "Displaying Critical Messages" on page 175). The color definitions indicate which color to use when a message is logged against a specific category. **CRITMSGA** through **CRITMSGA** are subcategories of **CRITMSG**. These subcategories are associated with different priorities, and the color of the **Critical Messages** panel item, which corresponds to **CRITMSG**, is determined by the message that belongs to the subcategory with the highest priority.

The default specifications assign messages ending in I as having the lowest priority. Messages ending in W have the next highest, messages ending in E have the next highest, and messages ending in A have the highest priority. Thus, if a message ending in A is logged, the **Critical Messages** item will turn to the color defined for those messages (probably red), overriding any other message color.

Special Start and Stop Dependencies

This section discusses general restrictions for the definition of dependency relationships between IMS regions and how an XRF, CQS, and FDR complex must be defined in the SA OS/390 policy database.

General Restrictions

IMS Automation supports dependency relationships as provided with the SA OS/390 product with the following restrictions:

- An IMS control region in a DB/DC environment must be a child, grandchild or greatgrandchild of VTAM by the **HasParent** relationship. This does not hold for DB control regions, since these do not use VTAM.

- Certain IMS dependent regions (DBRC and DLISAS) must be direct children of the control region by the **HasParent** relationship and must have their **External Startup** and **External Shutdown** fields set to ALWAYS in their AUTOMATION INFO policy item.

- Any region, for which the **External Startup** or the **External Shutdown** field is set to ALWAYS in its AUTOMATION INFO policy item, must be a direct child of the control region by the **HasParent/StartsMeAndStopsMe** relationship and must not have multiple parents defined.

- When a dependent region is defined as a child of more than one subsystem, then the owning control region must be specified in the **Control region name** field of the IMS ENVIRON policy item to enable IMS Automation to determine which control region owns the dependent region.

- IMS Automation does not support one-shot, transient subsystems as dependent regions. Any dependent region with a status of ENDED will be reset to a status of DOWN when the control region starts.

- CQS and FDR regions cannot be defined as dependent regions.

Defining an FDR Environment

Figure 12 illustrates the relationships that need to be set up to handle IMS FDR regions. These relationships prevent the FDR from being put into STOPPED state (outside of automation) when the FDR terminates.
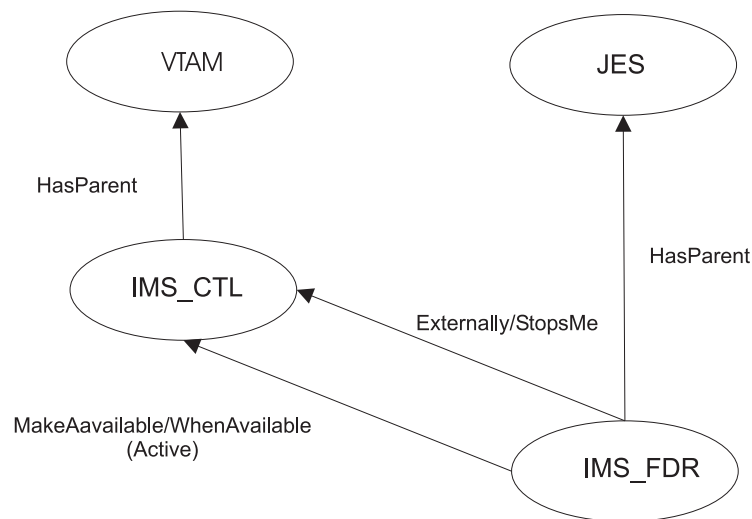
Figure 12. Relationships for IMS FDR

In order to simplify the start/stop of IMS and its associated FDR you can proceed as follows:

1. Create an application group (with **Application Group Type**=SYSTEM, **Nature**=BASIC) with the control region and its dependent regions as member resources.

2. Create another application group (with **Application Group Type**=SYSPLEX, **Nature**=BASIC) which contains the group defined in step 1 and the FDR region as its member resources.

Then, start and stop the application group created in step 2.

Defining an XRF Complex

An XRF complex has special definition requirements to allow the control regions to be started in the correct sequence and for synchronization of the dependent message regions. The following steps need to be taken.

1. For each XRF control region, create an application group (with **Application Group Type**=SYSTEM, **Nature**=SERVER) with the associated dependent message regions as member resources.

2. For each of the two XRF control regions, create an application group (with **Application Group Type**=SYSTEM, **Nature**=BASIC) with the control region and its associated DLI and DBRC regions as member resources.

3. Create another application group (with **Application Group Type**=SYSPLEX, **Nature**=SERVER) with member resources of both groups defined in step 2.

4. Create a **HasParent** relationship between the groups defined in step 1 and their respective control region.

5. Define a trigger as in the XRFTRIG sample trigger and link it to both the control regions.

Figure 13 on page 42 illustrates the group and relationship definitions:
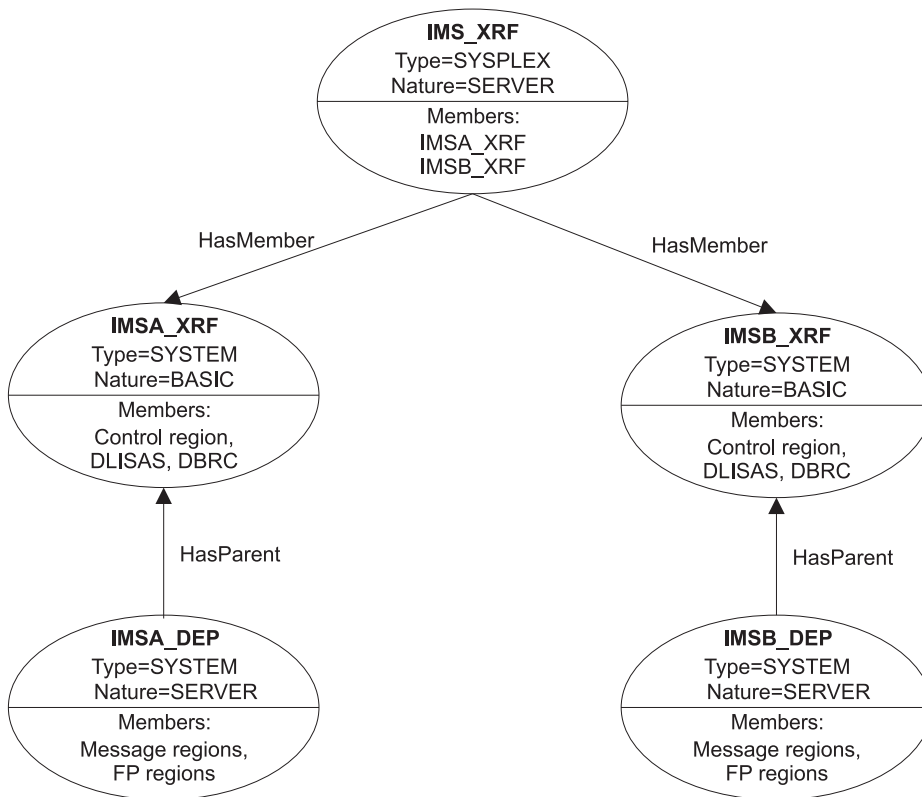
Figure 13. IMS XRF Groups, Resources and Relationships

Defining a CQS Complex

Figure 14 illustrates the relationships that should be defined to handle IMS CQS regions. The effect of this configuration is to prevent a shutdown of the IMS control region when the CQS region is not available.
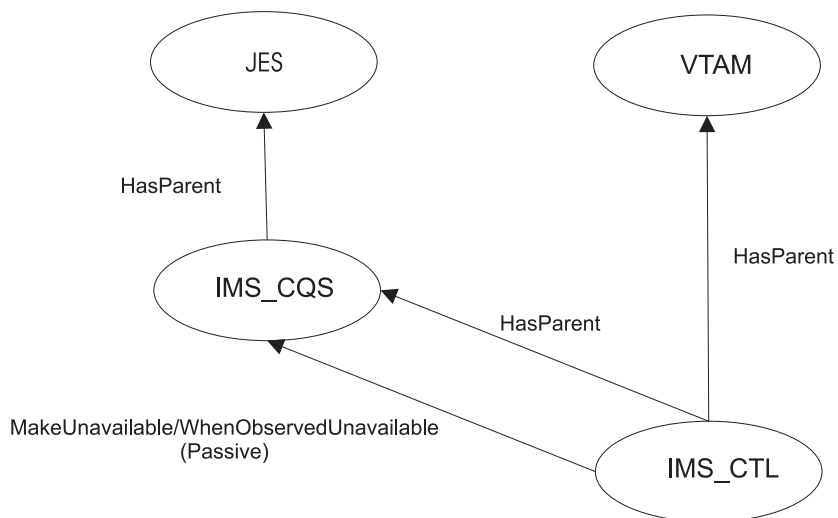
Figure 14. Relationships for IMS CQS

How to Automate Startup and Shutdown of IMS Regions

This section describes how to code startup and shutdown procedures for the different types of IMS regions.

Startup

For *control* regions, the start commands must be defined in the STARTUP item of the APPLICATION policy object. A possible format for these commands would be

```
MVS S job_name,PARM1='AUTO={Y|N}'
```

You must code a start command for every valid startup type except MANUAL (that is, for AUTO, BUILDQ, COLD, NORM, WARMSDBL). The **External Startup** field of the AUTOMATION INFO policy item must be set to NEVER for control regions.

For *any other* region type, you must specify start commands in the STARTUP policy item when the **External Startup** field of the AUTOMATION INFO policy item is set to INITIAL or NEVER. If it is set to ALWAYS such dependent regions are expected to be started by the associated control region.

You can pass user-defined parameters to the startup command by incorporating the &EHKVAR1 variable into the command. The user can specify a string in the **Appl Parms** field of the INGREQ input panel (for the INGREQ command see Chapter 8, "Starting and Stopping Resources" on page 163). The string is stored in the &EHKVAR1 variable before this variable is replaced by its value in the start command. The MANUAL startup type behaves differently in that the string specified in the **Appl Parms** is not used in the start command; rather it is used as the reply to the DFS810A WTOR from IMS (see "DFS810A—Define restart commands" on page 84).

XRF Dependent Regions

XRF dependent regions are a special case. If their **External Startup** field is set to NEVER or INITIAL, you must do two things:

- Code a startup command for them in the STARTUP item. The startup command *must* be EVIEI00T. EVIEI00T performs the synchronization between the *active* and the *alternate*.
- Specify the START keyword (see "START—Define start commands" on page 109) for them in the MESSAGES/USER DATA policy item as in the following example:

```
  COMMANDS  HELP
 -----------------------------------------------------------------------------
                                 CMD Processing              Row 1 to 2 of 20
  Command ===>                                              SCROLL===> PAGE

  Entry Type : Application           PolicyDB Name   : SCENARIO
  Entry Name : IMS722F1              Enterprise Name : TEST

  Subsystem  : IMS722F1
  Message ID : START

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text
  _____   _____
  MVS S IMS722F1_____
  _____

  _____   _____
  _____
  _____

   F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 15. Startup Command for Dependent XRF Regions

For specifying MESSAGES/USER DATA keywords, see "Translating Format Descriptions into MESSAGES/USER DATA Entries" on page 57.

Shutdown

Here, a distinction must be made between control regions and other region types.

Control Regions

For control regions, you must specify the shutdown commands in the SHUTDOWN item of the APPLICATION policy object. The commands must have the following format:

```
EVIET001 subsystem_name,{NORM|IMMED|FORCE},{DUMPQ|BACKUP|FREEZE|PURGE|DUMP|NODUMP}
```

The second parameter is called the shutdown *type*, the third is called the shutdown *option*.

A brief definition of the three shutdown types follows:

NORM   causes a checkpoint to be issued and an attempt to shut down IMS in an orderly, structured manner; cancellation of message regions and the control region occurs after a predetermined time delay.

IMMED   causes a checkpoint to be issued and immediate cancellation of message regions; cancellation of the control region occurs after a predetermined time delay.

FORCE   causes immediate flushing of the entire IMS complex, including message and control regions.

Generally, the NORM parameter is specified with the NORM phase of the SHUTDOWN policy item and the IMMED parameter is specified with the IMMED phase. It is possible, but not recommended, to intermix the NORM or IMMED parameters with either the NORM or IMMED phases (operator selection of the NORM or IMMED shutdown type in the INGREQ input panel relates to the NORM and IMMED phases respectively; however, the actual shutdown type is then dictated by either the NORM or IMMED parameter as specified in the command). The FORCE parameter *must* be specified with the FORCE phase.

The shutdown options (BACKUP, DUMPQ, FREEZE, PURGE, DUMP, NODUMP) serve to call additional commands during the shutdown process. Every shutdown option must be associated with a command through the reserved SHUTTYPES message ID (see "SHUTTYPES—Issue commands at operator shutdown" on page 106), which has to be defined in the MESSAGES/USER DATA policy item of the application. Note that you must specify DUMP or NODUMP as the shutdown option for shutdown type FORCE.

The option specified for the respective phase is the default; you can override it when you put a shutdown request to SA OS/390 with the INGREQ command. When a shutdown request contains no option override, EVIET001 issues the command that is associated with its shutoption parameter in the SHUTTYPES entry; otherwise it issues the command that is associated with the override option.

The following example illustrates the mechanism. Suppose that the shutdown command for the NORM phase is coded for the IMS control region IMSIMSZ as displayed in the following panel:

```
  COMMANDS  HELP
 -----------------------------------------------------------------------------
                         Shutdown Command Processing      Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application         PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ            Enterprise Name : TEST

 Subsystem     : IMSIMSZ
 Shutdown Phase: NORM

 Enter commands to be executed when the selected shutdown phase is invoked
 for this subsystem.

 Pass        Automated Function/'*'
 Command Text
 1_            _____
 EVIET00 IMSIMSZ,NORM,DUMPQ_____
 _____


 __            _____
 _____
  F1=HELP      F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP        F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Figure 16. Shutdown Command for IMS Control Region

Note that only one pass is coded. The logic in EVIET001 will handle all subsequent shutdown activities.

Suppose furthermore that the SHUTTYPES entry for IMSIMSZ looks as follows:

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                            Reply Processing            Row 1 to 7 of 20
 Command ===>                                            SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ             Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : SHUTTYPES

 Enter the replies to be issued when this resource issues the selected message.

 Pass/      Retry Reply Text
 Selection  Count
 DUMPQ_____   5_   /CHE DUMPQ_____
 BACKUP____   5_   /CHE BACKUP_____
 FREEZE____   5_   /CHE FREEZE_____
 PURGE_____   5_   /CHE PURGE_____
 _____   __   _____
 _____   __   _____
 _____   __   _____
  F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Figure 17. SHUTTYPES Entries

Now, if you require that IMSIMSZ be shut down with the INGREQ command, and
specify NORM as the shutdown type, then the command specified for the NORM
phase, namely `EVIET00 IMSIMSZ,NORM,DUMPQ` is issued (see Figure 16 on page 45).
This entails that the corresponding command of the SHUTTYPES entry, namely
`/CHE DUMPQ`, is invoked (see Figure 17), unless you specify another shutdown option
in the **Appl Parms** field of the INGREQ input panel. If you do this and enter, for
example, `OPTION=FREEZE` in this field, then `/CHE FREEZE` will be issued instead of
`/CHE DUMPQ`. See also "Shutdown" on page 165.

Other Region Types

For any other region type, two cases must be distinguished. If the **External
Shutdown** field of the AUTOMATION INFO policy item is set to ALWAYS, no
specifications are needed. Otherwise, proceed as follows.

A HASPARENT relationship must exist between the dependent region and the
owning IMS control region. This relationship is defined in the APPLICATION
policy item for the dependent region using the RELATIONSHIPS entry. The
HASPARENT definition, pointing to the control region, must have a sequence
number of 1.

Defining this HASPARENT relationship also removes the requirement to specify
the control region name on the 'Control Region Name' field of the IMS ENVIRON
entry in the dependent region's APPLICATION policy item even if multiple
parents are defined.

Specify shutdown commands in the SHUTDOWN item of the APPLICATION
policy object. The commands must have the following format:
`EVIET00J &SUBSAPPL,{NORM|IMMED|FORCE}`

The actual shutdown command is then determined by the shutdown type specified
for EVIET00J and by the region type. The BMP, FP, and dependent message region
types are each associated with a special reserved message ID which you must
define in the MESSAGES/USER DATA policy item of the associated *control* region.
These entries will specify for every shutdown type the command that is issued for
the region type in question. The following table specifies the message IDs for the

different types:

Region Type	Associated Message ID
BMP regions	STOPBMPREGION, see "STOPBMPREGION—Stop batch message regions" on page 110.
FP regions	STOPFPREGION, see "STOPFPREGION—Stop fast path regions" on page 112.
Dependent message regions	STOPREGION, see "STOPREGION—Stop IMS dependent message region" on page 113.

The following example illustrates the mechanism. Suppose that BMPIMSA is a BMP region, that the **External Shutdown** field of its AUTOMATION INFO policy item is not set to ALWAYS, and that the shutdown command for the NORM phase in its SHUTDOWN policy item is coded as follows:

```
  COMMANDS  HELP
 ----------------------------------------------------------------------------
                      Shutdown Command Processing        Row 1 to 2 of 20
 Command ===>                                            SCROLL===> PAGE

 Entry Type : Application        PolicyDB Name   : SCENARIO
 Entry Name : BMPIMSA            Enterprise Name : TEST

 Subsystem     : BMPIMSA
 Shutdown Phase: NORM

 Enter commands to be executed when the selected shutdown phase is invoked
 for this subsystem.

 Pass         Automated Function/'*'
 Command Text
 1_           _____
 EVIET00J &SUBSAPPL,NORM_____
 _____


 __           _____
 _____
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN   F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT    F12=RETRIEVE
```

Figure 18. Shutdown Command for a Dependent Region

Furthermore, suppose that BMPIMSA is associated with the control region IMSIMSZA. The message ID that determines the shutdown command for BMP regions is STOPBMPREGION. Thus, there must be an entry for this message ID in the MESSAGES/USER DATA item of IMSIMSZA, which could look as follows:

```
  COMMANDS  HELP
 --------------------------------------------------------------------------
                               Reply Processing            Row 1 to 7 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application           PolicyDB Name  : SCENARIO
 Entry Name : IMSIMSZA             Enterprise Name : TEST

 Subsystem  : IMSIMSZA
 Message ID : STOPBMPREGION

 Enter the replies to be issued when this resource issues the selected message.

 Pass/       Retry Reply Text
 Selection   Count
 NORMAL____  5_   /STOP REGION &EHKVAR1_____
 ABEND_____  5_   /STOP REGION &EHKVAR1 ABDUMP_____
 CANCEL____  5_   /STOP REGION &EHKVAR1 CANCEL_____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 _____    __   _____

  F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

Figure 19. STOPBMPREGION entry for Associated Control Region

The three **Pass/Selection** values correspond to the three shutdown types according
to the following table:

Shutdown type	Corresponding value
NORM	NORMAL
IMMED	ABEND
FORCE	CANCEL

Now, when a request to shut down BMPIMSA with shutdown type NORM is put
to SA OS/390, then EVIET00J is called as specified for the NORM phase of the
SHUTDOWN policy item (see Figure 18 on page 47). EVIET00J determines that
BMPIMS is a BMP region, consults the STOPBMPREGION entry of the associated
IMS control region IMSIMSZA (see Figure 19), and issues the command of that
entry whose first value corresponds to the second parameter of the call. In the
example, this is the command /STOP REGION &EHKVAR1.

Automating Recovery for Application Components

IMS Automation provides automated recovery for the following application
components:

- MSC links
- OLDS
- Transactions
- Programs.

You can control the automated recovery for these components through the
following three policy items of the APPLICATION object:

MINOR RESOURCE FLAGS

With these flags, you can switch automated recovery on and off for
application components. To do this, you must define a minor resource and
set its **Recovery** flag as required; for the definition of minor resources, see

System Automation for OS/390 Defining Automation Policy. The names of these minor resources must be as follows:

Table 9. Minor Resource Names for Application Components

Component	Minor resource name
MSC links	MSC[.*link_id*]
OLDSs	OLDS
Transactions	TRAN[.*trans_id*]TRXABEND
Programs	PROG[.*prog_id*]

For transactions, you can also define second-level minor resources by suffixing TRAN with the transaction name. The recovery flag of the TRAN minor resource applies to all transactions of the respective application, TRAN.*trans_id* only applies to the *trans_id* transaction. The transaction-specific recovery flag overrides the general TRAN flag. The same mechanism applies to the MSC and PROG minor resources.

When no minor resources are defined, IMS Automation acts according to the recovery setting of the application (AUTOMATION FLAGS policy item). When no second-level minor resource is defined for a transaction, the TRAN minor resource is applied. If that does not exist either, the application setting is applied. This also applies to the PROG and MSC resources. Thus, you only need to define minor resources when the recovery setting for a lower level is to be different from the next higher level.

RESOURCE THRESHOLDS

With this IMS-specific policy item (see *System Automation for OS/390 Defining Automation Policy*), you determine the threshold at which recovery should stop. This threshold is defined by the number of errors within a certain time interval. As with the recovery flags, you must associate the threshold definition with the transaction/problem area by giving it one of the names listed in Table 9; you can also specify thresholds for a single transaction, program, or MSC link.

MESSAGES/USER DATA

For every recovery type, there are one or more keywords that are used to specify how recovery is to proceed. These keywords are:

Table 10. MESSAGES/USER DATA Keywords for Component Recovery

Application components	Keywords
MSC links	DFS2142 (see page 76), DFS2161I (see page 77), DFS2169I (see page 78)
OLDSs	OLDS (see page 94), DFS3258A (see page 79)
Transactions	ABCODETRAN (see page 68), DFS554A (see page 81)
Programs	ABCODEPROG (see page 64), DFS554A (see page 81)

In the following sections, recovery configuration is described in more detail for transactions.

How to Define Transaction Recovery

Customization of transaction recovery consists of:
- Determining which application program (TP) transactions will have recovery automation
- Identifying the batch message region (BMP) transactions that will have recovery automation
- Specifying the error threshold level at which a recovery should stop
- Identifying specific abend codes for which you want recovery procedures to occur
- Specifying the recovery procedure, which usually consists of invoking a command, a routine, and/or sending notifications to an operator

The recovery itself is typically triggered from the MAT by calling the EVIEY00S routine when certain messages arrive at NetView. EVIEY00S then consults the TRAN state/action table in order to learn whether recovery is to be attempted. If so, the ACF is consulted in order to determine what has to be done.

The following sections illustrate the configuration process by an example.

Specifying the Transactions or Programs to be Recovered

Suppose that recovery is enabled for the IMS10AA application on the application level, and that you want it also enabled for transactions PAYR, DBTS, and BLNG, but not for any other transaction. Then you must define four minor resources for IMS10AA in the customization dialogs as follows:

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                        Minor Resource Selection        Row 1 to 10 of 24
 Command ===>                                               SCROLL===> PAGE

 Entry Type : Application           PolicyDB Name   : SCENARIO
 Entry Name : IMS10AA               Enterprise Name : TEST

 Major Resource: IMS10AA

 Select minor resource option to be altered.

  s   TRAN_____
  _   TRAN.BLNG_____
  _   TRAN.DBTS_____
  _   TRAN.PAYR_____
  _   _____
  _   _____
  _   _____
  _   _____
  _   _____
 _____
 F1=HELP     F2=SPLIT    F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN     F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

Figure 20. Defining Minor Resources for Transactions

Set the recovery automation flag to NO for TRAN and to YES for the three second-level minor resources. For example, to do this for TRAN, enter s in the **Select** column and press ENTER. The following panel is displayed:

```
   COMMANDS  ACTIONS  HELP
  ----------------------------------------------------------------------------
                          Flag Automation Specification
  Command ===>

  Entry Type : Application          PolicyDB Name   : SCENARIO
  Entry Name : IMS10AA             Enterprise Name : TEST
                                                              More:     +
  Resource:  IMS10AA.TRAN
  Enter level of automation desired.


  Automation Flags:  Y = Yes     N = No   E = Exits
  Assist Flags:      D = Display  L = Log  N = None


  Actions    Flag          Auto     Assist     Exits
             Automation .                        0
             Recovery . .  NO                    0
             Start. . . .                        0
             ShutDown . .                        0
             Initstart. .                        0
             Restart. . .                        0

   F1=HELP      F2=SPLIT     F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Figure 21. Automation Flag Panel

Here you specify which flags are set and which are not. For more information, see
System Automation for OS/390 Defining Automation Policy.

In the same way you can fine-tune recovery automation for programs. Just replace
the TRAN keyword with PROG.

Defining Recovery Thresholds

You can specify that recovery is to be stopped when the number of abends within
a certain time interval reaches a certain threshold. To do that, define thresholds
under the IMS-specific RESOURCE THRESHOLDS item of the APPLICATION
policy object. The thresholds must have the name TRAN or TRAN.*tranid*, where
the values of the TRAN thresholds will be used for all transactions *tranid* for which
no TRAN.*tranid* thresholds exist. The **Critical** value of the thresholds will be used.

If you want to stop recovery specifically for PAYR if two or more abends occur
within one hour, you must enter the values on the **Thresholds Definitions** panel
as follows:

```
   COMMANDS  HELP
  --------------------------------------------------------------------------
                           Thresholds Definition              Policy saved
  Command ===>

  Entry Type : Application           PolicyDB Name   : SCENARIO
  Entry Name : IMS10AA              Enterprise Name : TEST

  Resource:     IMS10AA.TRAN.PAYR
  Description:

  Specify the number of times an event must occur to define a particular level.

                    ---------------------- Levels -----------------------
                        Critical            Frequent            Infrequent
  Resource          Number  Interval   Number  Interval   Number  Interval
                            (hh:mm)            (hh:mm)            (hh:mm)

  IMS10AA.TRAN.PAYR    2      01:00       2      05:00       2      24:00




   F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
   F7=UP       F8=DOWN      F9=SWAP    F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

Figure 22. Thresholds Definitions Panel

For more details, see *System Automation for OS/390 Defining Automation Policy*.

For recovery thresholds of programs, you must use PROG instead of TRAN as the first
part of the thresholds' name.

Selecting the Abend Codes

The abend codes for which recovery is to take place are specified in the
MESSAGES/USER DATA policy item for IMS10AA through

* the ABCODETRAN keyword for transactions, and
* the ABCCODEPROG keyword for programs.

If you want to initiate recovery for transaction PAYR only when the abend code is
U3033 or U907, you must create the ABCODETRAN entry in the **Message
Processing** panel and associate codes with this entry as displayed in Figure 23 on
page 53:

```
   COMMANDS  HELP
 ----------------------------------------------------------------------------
                              Code Processing            Row 1 to 6 of 21
 Command ===>                                            SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name  : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : ABCODETRAN.PAYR

 Enter the value to be passed to the calling CLIST when this resource
 issues the selected message and the following codes are contained in
 the message.

     Code 1           Code 2           Code 3           Value Returned
 _____  U3033_____  _____  INCLUDE_____
 _____  U907_____  _____  INCLUDE_____
 _____  *_____  _____  EXCLUDE_____
 _____  _____  _____  _____
 _____  _____  _____  _____
 _____  _____  _____  _____

   F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 23. Code Processing Panel

For more details, see "ABCODETRAN—Transaction abend recovery" on page 68.

In a similar way, you can use the ABCODEPROG keyword to include certain
abend codes for certain programs in recovery automation.

Specifying Recovery Actions

You specify the commands to be issued for recovery in the **CMD Processing** panel
for the DFS554A message ID of IMS10AA. For example:

```
   COMMANDS  HELP
 ----------------------------------------------------------------------------
                              CMD Processing             Row 1 to 2 of 20
 Command ===>                                            SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name  : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : DFS554A

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 TRAN_____   _____
 IMSCMD &SUBSAPPL  /STA TRAN &EHKVAR1_____
 _____

 PROG_____   _____
 IMSCMD &SUBSAPPL  /STA PGM &EHKVAR2_____
 _____
   F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 24. Command Processing Panel

For more details, see "DFS554A—Respond to program abend" on page 81.

Note: The IMS/ESA V6 DFSNDMX0 Non-Discardable Message Exit Routine will
not alter the behaviour of this recovery action.

How to Set Up the State/Action Tables

In IMS Automation, state/action tables are used for recovery of the following application components:

- MSC links
- OLDs
- Transactions
- Programs.

State/action tables work independently of service periods and external triggers and are referenced when messages occur that are relevant to these entities. For an explanation on what state/action tables are and how they work, see "State/Action Tables" on page 18.

If you want to enable automated recovery for the application components listed above, proceed as follows:

- If the recovery flag of the application (AUTOMATION FLAGS policy item) is set to NO, define a minor resource for the respective component in the customization dialogs under the MINOR RESOURCE FLAGS policy item of the IMS application and set its recovery flag to YES. For details, see "Automating Recovery for Application Components" on page 48.
- Associate the respective IMS subsystem with a set of state/action tables. To do this, you must perform two steps in the customization dialogs:
  1. Define a set of state/action tables as an IMS STATE/ACTION policy object for IMS (ISA entry type).
  2. Link the set to the subsystem under the STATE ACTION TABLE policy item of the APPLICATION object.

The state/action tables are read and the actions respectively state changes performed by the common state handler routine. This routine is typically invoked from the MAT. It determines which set of state/action tables is associated with the subsystem that issued the message, and then consults the appropriate table. If the table cell determined by the state and the message ID contains a call of the recovery program , recovery is initiated. In the case of a transaction recovery, the recovery program makes use of the items described in "How to Define Transaction Recovery" on page 50.

Besides the state handler, the following components for support of state/action processing are shipped with IMS Automation:

- Default state/action tables for the different components. These are:

Component	Name of state/action table
MSC links	EVISS002
OLDS	EVISS003
Transactions/programs	EVISS005

- Common routines to be used by the action routines.
- Action routines for the different components.

Every state/action table is associated with an *area* and a *product*. The area tag specifies for which of the three problem areas the table is intended, the product tag

says whether the table is to be used by IMS Automation or by CICS Automation. These tags must be specified in the first two rows of the table. The format for IMS Automation is:

```
PRODUCT=IMS
AREA={MSC|OLDS|TRAN}
```

In the third row of the table, you must specify the initial state of the table, that is, the state that is assumed when the table is consulted for the first time.

Adding Local Applications to the IMS Automation Operator Interface

Option 99, Local Applications, from the IMS Automation main menu, enables you to add your local applications to the IMS Automation interface.

To do this, write a module named EVIEU000 using the usage notes described below. We do not provide this module. However, this module is called when option 99 is selected.

These usage notes assume that you understand how to write a NetView panel handler EXEC. These notes clarify unique functions or conventions used with IMS Automation. For your panel to be logically consistent with the IMS Automation interface, incorporate the following usage notes.

Usage notes:
1. To exit IMS Automation (PF2) or to return to the main menu (PF4) code the following after displaying your panel and accepting the input:

```
WHEN VIEWAID = 'PF2' | VIEWAID = 'PF14' THEN
  DO
    EVI_PF2 = 'YES'
    'GLOBALV PUTT EVI_PF2'
    EXIT 0
  END
    and
WHEN VIEWAID = 'PF4' | VIEWAID = 'PF16' THEN
  DO
    EVI_PF4 = 'YES'
    'GLOBALV PUTT EVI_PF4'
    EXIT 0
  END
```

When you call a module and you return from that module, you should exit if the called module displays a panel and PF2 or PF4 was pressed. To check for this, code the following after the call.

```
'GLOBALV GETT EVI_PF2'
IF EVI_PF2 = 'YES' THEN
  DO
    EXIT 0
  END
    and
'GLOBALV GETT EVI_PF4'
IF EVI_PF4 = 'YES' THEN
  DO
    EXIT 0
  END
```

2. To handle fastpath:
   - Add the following to the beginning of the program:
     ```
     'SIGNAL ON HALT'
     ```

     Add the following routine:

```
HALT:
    EVI_PF2 = 'YES'
    'GLOBALV PUTT EVI_PF2'
    EXIT 0
```

- Add the following code to support fastpath entered on your panel by the operator:

```
WHEN VIEWAID = 'ENTER' & CMD ¬= ''THEN
    DO
        IF SUBSTR(CMD,1,1) = '=' THEN
            DO
                PARSE VAR CMD '=' REST
                CMD = 'EVIE0000 ' || REST
            END
        'CMD HIGH 'CMD
    END
```

Note: In this code, CMD is the command line on the NetView panel.

- If you code a menu panel, add the following code to support fastpath entry:

```
'GLOBALV GETT EVI_SELECTION'
IF EVI_SELECTION ¬= ''
    DO
        PARSE EVI_SELECTION MYSELECTION '.' EVI_SELECTION
        'GLOBALV PUTT EVI_SELECTION'
    END
```

3. Support for the name of the IMS subsystem:

- On entry, or returning from a called routine, to get the IMS subsystem name (if the previous routine had a valid name and saved it) code the following:

```
'GLOBALV GETT EVISELNM'
MYNAME = EVISELNM
```

- Validate the name using the command processor described in "IMSQRY—Name lookup" on page 146. Following is an example of usage:

```
'IMSQRY REQ=VALIDATE,TYPE=IMS,NAME='MYNAME
IF RC ¬= 0
    DO
      write your error message
    END
ELSE
    EVISELNM = MYNAME
    'GLOBALV PUTT EVISELNM'
```

Chapter 5. MESSAGES/USER DATA Entries for IMS Automation

As IMS Automation is integrated into SA OS/390, you must enter any information for IMS Automation in the policy database via the customization dialogs. In most cases the customization dialogs precisely determine the format in which this information must be entered. There are, however, a number of IMS-specific automation parameters that must or can only be specified as entries in the MESSAGES/USER DATA item of the application policy object; for the MESSAGES/USER DATA policy item, see *System Automation for OS/390 Defining Automation Policy*. In these cases, the customization panels provide no information about the keywords and the format of their parameters.

The following chapter contains detailed descriptions of these automation entries. Note, however, that a general understanding of the MESSAGES/USER DATA policy item will be assumed.

Translating Format Descriptions into MESSAGES/USER DATA Entries

The following three examples show how to convert the formal descriptions of the keyword parameters into entries in the MESSAGES/USER DATA panels of the customization dialogs.

The first example is the RESTARTABORT keyword. With this entry, you specify the emergency command after an abort termination of a restart process (see "RESTARTABORT—Emergency restart commands" on page 104 for more details). The format description of RESTARTABORT is as follows:

```
Format
IMS Control Region:
RESTARTABORT REP=(OVERRIDE,n,command_text)
             REP=(BACKUP,n,command_text)
             REP=(ERE,n,command_text)

DB Control Region:
RESTARTABORT CMD=(OVERRIDE,,command_text)
             CMD=(ERE,,command_text)
```

The NAME=value pairs are called the *attributes* of the entry. RESTARTABORT can be used with the REP or the CMD attribute. You must specify three instances of the REP attribute, respectively two instances of the CMD attribute, to meet different abort situations.

For IMS control regions, you usually code the REP attribute since IMS control regions use a WTOR for accepting commands (REP stands for 'reply'); you can also use the CMD attribute, but as a rule this is not specified in the format descriptions. For DB Control regions, the CMD attribute must be used; the reason for this is that unlike IMS control regions, DB control regions do not use a WTOR.

For general information on the format descriptions, see "Notation for Format Descriptions" on page xi.

To specify a RESTARTABORT entry for an IMS control region IMSIMSZ in the customization dialogs, you must call the **Message Processing** panel for that subsystem:

```
   COMMANDS   ACTIONS   HELP
 -----------------------------------------------------------------------------
                          Message Processing          Row 1 to 4 of 20
 Command ===>                                          SCROLL===> PAGE

 Entry Type  : Application          PolicyDB Name   : SCENARIO
 Entry Name  : IMSIMSZ             Enterprise Name : TEST

 Subsystem   : IMSIMSZ

 Enter messages issued by this resource that will result in automated actions.
 Actions:  CMD = Command   REP = Reply  CODE = CODE   USER = User defined values

 Action    Message ID                        Cmd  Rep  Code  User
           Description
 REP____   RESTARTABORT_____
           Emergency commands for restart abort___
 _____    _____
           _____
 _____    _____
           _____
 _____    _____

  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

Figure 25. Message Processing Panel of the Customization Dialogs 1

In this panel you specify the keyword of the entry (RESTARTABORT in our example) in the **Message ID** field. The attributes are specified through the **Action** field. Here, two cases must be distinguished according to the following rule:

Rule

- The attribute names CMD, CODE, and REP must be entered in the **Action** field; the values for these attributes are specified in a follow-on panel.
- For all other attributes, you must enter USER in the **Action** field; in this case, both name and value are entered in a follow-on panel.

For the CODE attribute and attributes defined by USER actions, see below.

The fields on the right side of the panel specify how many actions of the respective type are associated with the message ID of the respective line.

To specify the value for the REP attributes enter REP in the **Action** field and press ENTER. This invokes the **Reply Processing** panel:

```
  COMMANDS  HELP
  -------------------------------------------------------------------------
                              Reply Processing            Row 1 to 7 of 20
  Command ===>                                            SCROLL===> PAGE

  Entry Type : Application          PolicyDB Name   : SCENARIO
  Entry Name : IMSIMSZ             Enterprise Name : TEST

  Subsystem  : IMSIMSZ
  Message ID : RESTARTABORT

  Enter the replies to be issued when this resource issues the selected message.

  Pass/      Retry Reply Text
  Selection  Count
  OVERRIDE__  5_   /ERE OVERRIDE._____
  BACKUP____  5_   /ERE BACKUP._____
  ERE_____  5_   /ERE._____
  _____   __    _____
  _____   __    _____
  _____   __    _____
  _____   __    _____

   F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
   F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Figure 26. Reply Processing Panel of the Customization Dialogs

Every entry in this panel consists of three fields that correspond to the three items of the value list for the REP attribute. Thus, the general format for the REP attribute is

REP=([*Pass/Selection*],[*Retry_Count*],*Reply_Text*)

The format description of the REP attribute for a certain keyword specifies what type of information you must enter in the three fields. For the RESTARTABORT keyword, all three fields must be filled for all three replies.

For the CMD, REP, and CODE attributes, the following notational conventions apply:

┌─ **Notational Conventions for CMD, CODE, REP Attributes** ────────────┐
│   │
│ • The '=' sign, the parentheses enclosing the value list, and the commas │
│   separating the individual values must not be entered in the panels. │
│   │
│   They just serve to make the format description more readable and to │
│   identify uniquely the panel field with which a value specification is │
│   associated.   │
│   │
│ • When the format of any value is specified in more detail, and the │
│   specification itself contains a comma, the value is enclosed in single quotes; │
│   these quotes must not be entered in the respective panel field.   │
│   │
└──┘

For the other attributes, the conventions are different; see "Notational Conventions for USER Type Attributes" on page 63.

For more information on the panel fields, see *System Automation for OS/390 Defining Automation Policy*.

Now consider the subsystem IMS0C1, which is not an IMS control region, but a DB control region. In this case you specify the RESTARTABORT keyword as before;

but now you must enter CMD in the **Action** field, according to the Rule stated on page 58. Pressing ENTER invokes the following panel:

```
   COMMANDS  HELP
  -----------------------------------------------------------------------------
                                 CMD Processing               Row 1 to 2 of 20
  Command ===>                                                SCROLL===> PAGE

  Entry Type : Application          PolicyDB Name   : SCENARIO
  Entry Name : IMS0C1               Enterprise Name : TEST

  Subsystem  : IMS0C1
  Message ID : RESTARTABORT

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text
  OVERRIDE__        _____
  MVS &EHKVAR7ERE COLDSYS OVERRIDE._____
 _____

   ERE_____        _____
   MVS &EHKVAR7ERE._____
 _____
   F1=HELP       F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
   F7=UP         F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

Figure 27. CMD Processing Panel of the Customization Dialogs

The general format for the CMD attribute is

CMD=([*Pass/Selection*],[*Automated_Function*],*Command_Text*)

The format description of the CMD attribute for a certain keyword specifies what type of information you must enter in the three fields. For the RESTARTABORT keyword, the **Automated Function** field must be left blank.

For more information on the panel fields, see *System Automation for OS/390 Defining Automation Policy*.

The ABCODETRAN entry (see "ABCODETRAN—Transaction abend recovery" on page 68) supplies the second example. This entry serves to specify conditions for transaction recovery (for more details on transaction recovery, see "How to Define Transaction Recovery" on page 50). The format of the ABCODETRAN entry is as follows:

```
  Format
ABCODETRAN[.tran]
            CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})
            [CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})]
            .
            .
            .
            [CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})]
```

For ABCODETRAN, you must specify one or more instances of the CODE attribute.

If you want transaction recovery to be performed for transaction PAYR on IMS10AA only when the abend code is U3033 or U907, you must enter the ABCODETRAN entry for IMS10AA as follows.

First specify the keyword in the **Message Processing** panel:

```
   COMMANDS  ACTIONS  HELP
 ---------------------------------------------------------------------------
                            Message Processing              Policy saved
 Command ===>                                          SCROLL===> PAGE

 Entry Type  : Application        PolicyDB Name   : SCENARIO
 Entry Name  : IMS10AA            Enterprise Name : TEST

 Subsystem   : IMS10AA

 Enter messages issued by this resource that will result in automated actions.
 Actions:  CMD = Command  REP = Reply  CODE = CODE  USER = User defined values

 Action    Message ID                           Cmd  Rep  Code  User
           Description
 CODE___   ABCODETRAN.PAYR_____
           Abend codes for recovery_____
 _____    _____
           _____
 _____    _____
           _____
 _____    _____
           _____
 _____    _____

   F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 28. Message Processing Panel of the Customization Dialogs 2

Note that the keyword is expanded by the transaction name in order to restrict the application of the specified codes to transaction PAYR.

Now you must enter CODE in the **Action** column according to the rule stated on page 58. When you press ENTER, the **Code Processing** panel is called:

```
  COMMANDS  HELP
 ---------------------------------------------------------------------------
                              Code Processing              Row 1 to 6 of 21
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : ABCODETRAN.PAYR

 Enter the value to be passed to the calling CLIST when this resource
 issues the selected message and the following codes are contained in
 the message.

    Code 1           Code 2           Code 3           Value Returned
 _____  U3033_____  _____  INCLUDE_____
 _____  U907_____  _____  INCLUDE_____
 _____  *_____  _____  EXCLUDE_____
 _____  _____  _____  _____
 _____  _____  _____  _____
 _____  _____  _____  _____

  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 29. Code Processing Panel of the Customization Dialogs

Here you must specify the values of the CODE attribute as displayed in Figure 29 on page 62. The general format of the CODE attribute is:

CODE=([*Code_1*],[*Code_2*],[*Code_3*],*Value_Returned*)

For the ABCODETRAN keyword, **Code 1** must be a transaction name, **Code 2** an abend code, and **Code 3** the name of a program that abended. **Value Returned** must specify whether to include in or to exclude from recovery the combination specified by **Code 1** to **Code 3**. For other keywords, however, the fields can have a quite different function.

An asterisk (*) is admitted as a *trailing* wildcard character for the three **Code** fields; that is, you can specify simply * and ABC*, but not *ABC. For more information on the panel fields, see *System Automation for OS/390 Defining Automation Policy*.

The third example is the OLDS keyword. By this, the automation settings for online data sets are defined (see "OLDS—Define recovery criteria for OLDS" on page 94). The format description of OLDS is as follows:

```
┌─ Format ──────────────────────────────────────────────────────────────
│
│ OLDS[MINIMUM=nn]
│     [SPARES=(nn,nn...)]
│     [ARCHIVETIME=hh:mm:ss]
│     [RETRYCNT=n]
│     [BACKOUT=nn]
│
└───────────────────────────────────────────────────────────────────────
```

To define OLDS call the **Message Processing** panel for the respective application, enter the OLDS keyword in the **Message ID**, specify USER in the **Action** field on the OLDS line according to the rule stated on page 58, and press ENTER. The following panel is invoked:

```
  COMMANDS  HELP
 ----------------------------------------------------------------------------
                            User Defined Data              Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application           PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ              Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : OLDS

 To change keyword-data pair, specify the following:

 Keyword
 Data
 MINIMUM_____
 5_____
 _____

 SPARES_____
 (97,98,99)_____
 _____
  F1=HELP       F2=SPLIT      F3=END       F4=RETURN     F5=RFIND      F6=RCHANGE
  F7=UP         F8=DOWN       F9=SWAP      F10=LEFT      F11=RIGHT     F12=RETRIEVE
```

Figure 30. User Defined Data Panel of the Customization Dialogs

As you can see from Figure 30, you must enter the attribute names in the **Keyword**
field and the attribute value(s) in the **Data** field.

For attributes of the USER action type, the following conventions apply:

Notational Conventions for USER Type Attributes

- The '=' sign must not be entered in the panels.
- Everything to the right of the '=' sign, including parentheses, commas, and
  single quotes, must be entered in the **Data** field.

For more information on the panel fields, see *System Automation for OS/390 Defining
Automation Policy*.

IMS-Specific MESSAGES/USER DATA Keywords

Most of the following keywords must be specified in the MESSAGES/USER DATA
item of the respective *control* region and apply to DB control regions as well as to
IMS control regions. Any deviation from this is pointed out in the description of
the respective keyword. This also holds for keywords that only apply to a FDR,
CQS, or XRF environment.

ABCODEPROG—Respond to BMP region abends

Use this entry to define actions to be taken for program abends of program-driven batch message processing (BMP) regions. Only abends for program-driven BMPs use this entry. Transaction driven BMPs use ABCODETRAN to determine recovery actions.

```
┌─ Format ─────────────────────────────────────────────────────────────────┐
│ ABCODEPROG[.progid]                                                        │
│              CODE=(*,acode,progid,INCLUDE|EXCLUDE)                         │
│              CODE=(*,acode,progid,INCLUDE|EXCLUDE)                         │
└───────────────────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

ABCODEPROG[.*tran*]
> You can add the name of a program as a suffix to the keyword. In this case the specifications of the CODE attribute(s) will only apply to this program.

CODE=
> Use this keyword to define which abends should be included or excluded from recovery.

> ***** An asterisk as the first positional parameter is required for compatibility with the ABCODETRAN entry. Code an asterisk as shown.

> *acode*
> > The abend code. An asterisk (*) can be used for generic specifications. System abend codes should be prefixed with an S, such as S0C1.

> *progid*
> > The program name.

> **INCLUDE|EXCLUDE**
> > Indicates whether or not to initiate a recovery sequence for this program and abend code combination. Use INCLUDE to initiate a recovery and EXCLUDE if you do not want a recovery initiated. INCLUDE is the default.

Comments and Usage Notes

If the ABCODEPROG entry is omitted, no recovery takes place and a warning message is issued.

The program name can either be specified as **ABCODEPROG.**progid or as the first value of the CODE attribute. Use **ABCODEPROG.**progid when you want all of the specifications to apply to one specific program. Use the CODE attribute when you want to code several transactions.

Examples of Usage

```
   COMMANDS  HELP
 ---------------------------------------------------------------------------
                                 Code Processing            Row 1 to 6 of 21
 Command ===>                                               SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : ABCODEPROG

 Enter the value to be passed to the calling CLIST when this resource
 issues the selected message and the following codes are contained in
 the message.

     Code 1           Code 2          Code 3           Value Returned
 *_____ U0778_____ SAMPLE4_____ EXCLUDE_____
 *_____ *_____  *_____ INCLUDE_____
 _____  _____  _____  _____
 _____  _____  _____  _____
 _____  _____  _____  _____
 _____  _____  _____  _____

  F1=HELP       F2=SPLIT    F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP         F8=DOWN     F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 31. Code Processing Panel of the Customization Dialogs

In this example, recovery will not be attempted for program SAMPL4 when the abend code is U0778. In all other cases, recovery will take place.

ABCODES/ACTCODES/ALTCODES—Restart control region after abend

IMS Automation processing utilizes the SA OS/390 CODE specification for automated error code matching, to specify the abend codes for which SA OS/390 will automatically restart an IMS control region after the abend process is complete.

```
┌─ Format ──────────────────────────────────────────────────────────
  ABCODES CODE=({SYScode│IMScode},,,ABENDING│STOPPING)
          .
          .
          .
          CODE=({SYScode│IMScode},,,ABENDING│STOPPING)
  ACTCODES CODE=({SYScode│IMScode},,,ABENDING│STOPPING)
          .
          .
          .
          CODE=({SYScode│IMScode},,,ABENDING│STOPPING)
  ALTCODES CODE=({SYScode│IMScode},,,ABENDING│STOPPING)
          .
          .
          .
          CODE=({SYScode│IMScode},,,ABENDING│STOPPING)
```

Keyword and Parameter Definitions

ABCODES
An ABCODES entry lists codes for a non-XRF IMS control region.

ACTCODES
An ACTCODES entry lists codes for an *active* XRF IMS control region.

ALTCODES
An ALTCODES entry lists codes for an *alternate* XRF IMS control region.

CODE=
Use this keyword to define the data, as shown in the following list:

{SYS*code*│IMS*code*}
Specify the matching abend code here. All abend codes must start with either SYS for MVS system abend codes or IMS for IMS user abend codes. For system abends, *code* is a three-character alphanumeric. For IMS user abends, *code* is a four-digit number.

ABENDING│STOPPING
ABENDING defines a recoverable IMS abend. With this abend, IMS can be restarted. STOPPING defines a non-recoverable IMS abend.

Comments and Usage Notes

1. If an abend code is omitted, the abend is considered non-recoverable.
2. When an IMS control region abends with an abend code listed in the appropriate table, SA OS/390 will automatically restart the IMS control region.
3. One of the three entries is searched depending on the type of IMS control region that is abending. For an XRF IMS subsystem, it is recommended that both ACTCODES and ALTCODES entries be specified.

Examples of Usage

```
   COMMANDS  HELP
 ----------------------------------------------------------------------------
                             Code Processing            Row 1 to 6 of 21
 Command ===>                                            SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ             Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : ABCODES

 Enter the value to be passed to the calling CLIST when this resource
 issues the selected message and the following codes are contained in
 the message.
     Code 1          Code 2          Code 3           Value Returned
 SYS0C4_____ _____ _____ ABENDING_____
 IMS0113_____ _____ _____ ABENDING_____
 IMS0020_____ _____ _____ ABENDING_____
 IMS0707_____ _____ _____ ABENDING_____
 _____ _____ _____ _____
 _____ _____ _____ _____
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

In this example, IMSIMSZ is supposed to be a non-XRF control region. It will be restarted after an abend if the abend code is one of the four listed. Any other abend will be considered irrecoverable.

ABCODETRAN—Transaction abend recovery

Use this entry to include transaction abend codes in recovery or exclude them from recovery. This includes both transaction-driven message processing regions (MP) and transaction-driven batch message processing (BMP) regions.

```
  Format
ABCODETRAN[.tran]
              CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})
              [CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})]
              .
              .
              .
              [CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})]
```

Keyword and Parameter Definitions

ABCODETRAN[.*tran***]**

You can add the name of a transaction as a suffix to the keyword. In this case the specifications of the CODE attribute(s) will only apply to this transaction.

CODE

Defines which abends are recoverable, as shown in the following descriptions:

tran

The transaction ID.

abend1

The abend code.

pgm

The program that abended.

INCLUDE|EXCLUDE

Indicates whether or not to initiate a recovery for this transaction, abend code, and program. Use INCLUDE to initiate a recovery and EXCLUDE if you do not want a recovery initiated.

Comments and Usage Notes

1. The transaction name is either specified as **ABCODETRAN.***tran* or as the first value of the CODE attribute. Use **ABCODETRAN.***tran* when you want all of the specifications to apply to one specific transaction. Use the CODE attribute when you want to code several transactions.

Examples of Usage

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                              Code Processing              Row 1 to 6 of 21
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : ABCODETRAN.PAYR

 Enter the value to be passed to the calling CLIST when this resource
 issues the selected message and the following codes are contained in
 the message.

     Code 1            Code 2           Code 3          Value Returned
 _____  U3033_____  _____  INCLUDE_____
 _____  U907_____  _____  INCLUDE_____
 _____  *_____  _____  EXCLUDE_____
 _____  _____   _____  _____
 _____  _____   _____  _____
 _____  _____   _____  _____
  F1=HELP     F2=SPLIT     F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

In this example, recovery will take place for transaction PAYR on IMS10AA if the
abend code is U903 or U3033. No recovery will take place for any other transaction
or abend code.

BRO—Broadcast a message prior to shutdown

This entry defines the command and message text that may be issued prior to
normal shutdown to indicate to active users that shutdown of the associated IMS
subsystem is imminent.

Format

```
BRO REP=(,n,/BRO ACTIVE)
    [REP=(,n,message_text.)]
```

Keyword and Parameter Definitions

/BRO ACTIVE

This command may be issued prior to normal shutdown to indicate to active
users that the IMS subsystem (*subsys*) shutdown is imminent.

message text

The message text may be issued by the /BRO ACTIVE command as described
in the parameters above. The variable &EHKVAR1 can be used in the message
text; its value will be the number of minutes that IMS Automation will wait
until the IMS is shut down, as specified in the **Broadcast Delay** field of the
IMS ENVIRON policy item of the application.

Comments and Usage Notes

⇒ **Not required**. Not applicable for DB control regions.

You must end the *message text* input for this entry with a period.

This IMS command and associated message text can be issued just prior to the
normal IMS control region shutdown process.

One variable is available for use in the BRO entry. &EHKVAR1 is the Broadcast
value specified in the **Broadcast Delay** field of the IMS ENVIRON policy item.

Examples of Usage

```
   COMMANDS  HELP
 -------------------------------------------------------------------------
                            Reply Processing           Row 1 to 7 of 20
 Command ===>                                           SCROLL===> PAGE

 Entry Type : Application        PolicyDB Name  : SCENARIO
 Entry Name : IMSIMSZ            Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : BRO

 Enter the replies to be issued when this resource issues the selected message.

 Pass/      Retry Reply Text
 Selection  Count
 _____    5_   /BRO ACTIVE_____
 _____    5_   IMS SHUTTING DOWN IN &EHKVAR1 MINUTES._____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

CHE—Issue a checkpoint command

This entry specifies the IMS command to be issued when a normal checkpoint is required.

Format

```
IMS Control Region:
  CHE REP=(,n,/CHE)

DB Control Region:
  CHE CMD=(,,MVS &EHKVAR7CHE)
```

Keyword and Parameter Definitions

n   The number of retry attempts (*n*).

/CHE
    Code the /CHE command exactly as specified.

Comments and Usage Notes

This IMS command is issued when a normal checkpoint is required.

&EHKVAR7 is a variable containing the console command character.

CQS0031A—Confirm CQS restart for structure

This entry defines the reply required in response to the WTOR message CQS0031A during CQS startup.

Format

```
CQS0031A REP=(,,CONFIRM)
```

Keyword and Parameter Definitions

CONFIRM
Code this exactly as specified.

Comments and Usage Notes

Not applicable to non-shared queue systems.

This reply is issued in response to CQS0031A WTOR informing subsys that CQS could not read the system checkpoint log token from the checkpoint data set, but found a system checkpoint log token in the control entry in the shared queues structure. Replying CONFIRM allows CQS to use the log token in the CQS0031A message (that is, the system checkpoint log token in the control entry) for restart processing.

CQS0032A—Respond to CQS structure restart

This entry defines the reply required in response to the WTOR message CQS0032A during CQS startup.

Format
```
CQS0032A REP=(,,COLD)
```

Keyword and Parameter Definitions

COLD
    Code this exactly as specified.

Comments and Usage Notes

Not applicable to non-shared queue systems.

This reply is issued in response to CQS0032A WTOR informing subsys that CQS could not read the system checkpoint log token from either the checkpoint data set or from the control entry in the shared queues structure. CQS does not know where to start reading from the log without the log token. Replying COLD allows CQS to begin coldstart processing.

CQS0033A—Respond to client takeover restart

This entry defines the reply required in response to the WTOR message CQS0033A during CQS startup.

Format

```
CQS0033A REP=(,,COLD)
```

Keyword and Parameter Definitions

COLD
Code this exactly as specified.

Comments and Usage Notes

Not applicable to non-shared queue systems.

This reply is issued in response to CQS0033A WTOR informing subsys that CQS could not read the system checkpoint log token for the CQS that was connected to the failed client from the control entry in the shared queues structure or an error prevented CQS from accessing all required log records. Replying COLD allows CQS to begin client takeover coldstart processing.

CQSET—Issue Structure Checkpoint at CQS Termination

This entry defines the /CQSET command that will be issued during IMS startup to cause a Structure Checkpoint at CQS shutdown.

Format

```
CQSET CMD=(,,IMSCMD &SUBSAPPL /CQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL)
```

Keyword and Parameter Definitions

IMSCMD &SUBSAPPL /CQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL
Code this exactly as specified.

Comments and Usage Notes

Not applicable to non-shared queue systems.

The command is issued during IMS startup. The process is triggered by message EVI130I indicating that the PPI is active. When the /CQSET command is issued, IMS passes the request to CQS when IMS terminates normally with a /CHECKPOINT FREEZE|DUMPQ|PURGE command.

IMS Automation code will ensure that the &SUBSAPPL task global is set to the correct IMS.

IMS Automation code correlates CQS regions to IMS Control regions during IMS Control region startup.

DFS2142—Respond to stopped logical link path message

This entry defines the commands required for response to the DFS2142 message following notification that a logical link path was stopped.

```
┌─ Format ──────────────────────────────────────────────────┐
│ DFS2142 REP=(START,,/START MSNAME&EHKVAR2)                 │
│         [REP=(RESTART,,/PSTOP LINK   &EHKVAR1)]            │
│         [REP=(RESTART,,/START MSNAME &EHKVAR2)]            │
│         [REP=(RESTART,,/RSTART LINK  &EHKVAR1)]            │
└───────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

START
When message DFS2142 is issued, the START reply will be issued to start the logical link path.

RESTART
When message DFS2142 is issued, the RESTART reply may optionally be issued to stop the link, start the link path and then restart the link.

/START MSNAME &EHKVAR2
Code the /START MSNAME command exactly as specified.

/PSTOP LINK &EHKVAR1
Code the /PSTOP LINK command exactly as specified.

/RSTART LINK &EHKVAR1
Code the /RSTART LINK command exactly as specified.

Comments and Usage Notes
Not applicable for DB control regions.

This reply is issued in response to the DFS2142 message informing the respective subsystem that a logical link path has stopped. Issuing the /START MSNAME command will activate the link path back. For persistent occurrences of this message, stopping the link, starting the link path and then restarting the link is indicated. If this is done, it must be performed on both IMS systems.

Two variables are used to automate this response.
- &EHKVAR1 is a variable containing the logical link number.
- &EHKVAR2 is a variable containing the logical link path name.

DFS2161I—Link stopped by other system

This entry defines the command required (for response to the DFS2161I message) following the stopping of a link by the other system attached to it.

Format

```
DFS2161I REP=(RESTART,,/RSTART LINK &EHKVAR1)
```

Keyword and Parameter Definitions

RESTART
When message DFS2161I is issued, the RESTART reply will be issued to restart the link with the other system.

/RSTART LINK &EHKVAR1
Code the /RSTART LINK command exactly as shown.

Comments and Usage Notes

Not applicable for DB control regions.

This reply is issued in response to the DFS2161I message informing *subsys* that a link was stopped by the IMS system at its other end. The /RSTART LINK command must be issued on both IMS systems in order to restart the link.

&EHKVAR1 is a variable containing the logical link number.

DFS2169I—Respond to MSC link disconnection message

This entry defines the command required (for response to the DFS2169I message) following the completion of the disconnection of a Multiple Systems Coupling (MSC) link between two IMS systems.

Format

```
DFS2169I REP=(RESTART,,/RSTART LINK &EHKVAR1)
```

Keyword and Parameter Definitions

RESTART
When message DFS2169I is issued, the RESTART reply will be issued to restart the link with the other system.

/RSTART LINK &EHKVAR1
Code the /RSTART LINK command exactly as shown.

Comments and Usage Notes
Not applicable for DB control regions.

This reply is issued in response to the DFS2169I message informing *subsys* that disconnection of a link is complete. The /RSTART LINK command must be issued on both IMS systems in order to restart the link.

&EHKVAR1 is a variable containing the logical link number.

DFS3258A—No online data sets available

This entry defines the commands required (for response to the DFS3258A message) following the use of either the last OLDS or all OLDS.

Format

```
DFS3258A CMD=(SYSTEM,,command_text_1),
         CMD=(LAST,,command_text_2)
```

Keyword and Parameter Definitions

SYSTEM

When message DFS3258A is issued and there are no more available OLDS for this IMS subsystem, the SYSTEM command will be issued.

command_text_1

The command to be executed when there are no more available OLDS.

LAST

When message DFS3258A is issued and the IMS subsystem is using the last available OLDS, the LAST command will be issued.

command_text_2

The command to be executed when the last available OLDS is in use.

Comments and Usage Notes

This command is issued in response to the DFS3258A message.

Examples of Usage

```
  COMMANDS  HELP
 ------------------------------------------------------------------------------
                                CMD Processing            Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ              Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : DFS3258A

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 SYSTEM____     _____
 IMSCMD &SUBSAPPL /LOG WAITING_____

 _____

 LAST_____     _____
 IMSCMD &SUBSAPPL /STA OLDS 99_____

 _____
  F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

DFS3869A/DFS3869I—Alternate XRF IMS subsystem detects potential failure

This entry defines the commands required (for response to DFS3869A or DFS3869I messages) following the *alternate* XRF IMS subsystem's detection of a potential failure of the *active* XRF IMS subsystem.

```
┌─ Format ──────────────────────────────────────────────────────┐
│  DFS3869A REP=(,n,/SWI SYSTEM FORCE)                           │
│  DFS3869I REP=(,n,/SWI SYSTEM FORCE)                           │
└───────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

/SWI SYSTEM FORCE
    This parameter must be specified exactly as shown to enable takeover to proceed.

Comments and Usage Notes
Required only for an XRF IMS subsystem.

These replies are issued in response to messages DFS3869A and DFS3869I, indicating that the *alternate* IMS subsystem has detected a potential failure of the *active* IMS subsystem. REP is coded as shown in the format above.

The command will only be issued when the **Non automatic takeover** field of the IMS ENVIRON policy item is set to YES.

DFS554A—Respond to program abend

This entry defines the commands required for response to the DFS554A message, following notification of a program abend.

Format

```
DFS554A CMD=(PROG,,command_text)
        CMD=(PROG,,command_text)
or

DFS554A REP=(TRAN,,command)
        REP=(PROG,,command)
```

Keyword and Parameter Definitions

TRAN
> When message DFS554A is issued and the DFS554A message indicates the transaction is stopped, the TRAN reply is issued to restart the transaction.

PROG
> When message DFS554A is issued and the DFS554A message indicates the program is stopped, the PROG reply or command is issued to restart the program.

command
> Usually, you will code these commands as shown in the first example if you are using REP.

command_text
> Usually, you will code this command as shown in the second example if you want IMS Automation to automatically start the program and you are using CMD. Note that you *must* use IMSCMD for IMS control regions in this case.

Comments and Usage Notes

For DB control regions, use the **CMD=** syntax only. Note that transactions are not handled by DB control regions so that you need not code the **CMD=(TRAN,...)** attribute for a DB control region.

This reply is issued in response to the DFS554A message issued when an IMS program abends to restart the transaction and the program.

Two variables are used to automate this response:
* &EHKVAR1 is a variable containing the transaction name.
* &EHKVAR2 is a variable containing the program name.

Four additional variables are supplied for job- or region-unique recovery that may be necessary:
* &EHKVAR3 is a variable containing job identifier.
* &EHKVAR4 is a variable containing region identifier.
* &EHKVAR5 is a variable containing user abend code.
* &EHKVAR6 is a variable containing system abend code.

DFS554A

Examples of Usage

```
   COMMANDS  HELP
 -------------------------------------------------------------------------------
                             Reply Processing              Row 1 to 7 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ             Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : DFS554A

 Enter the replies to be issued when this resource issues the selected message.

 Pass/      Retry Reply Text
 Selection  Count
 TRAN_____  __   /STA TRAN &EHKVAR1_____
 PROG_____  __   /STA PGM &EHKVAR1_____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 _____    __   _____
 _____      __
 -------------------------------------------------------------------------------
  F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

In this example, the REP attribute is used for the IMS control region IMSIMSZ.

```
   COMMANDS  HELP
 -------------------------------------------------------------------------------
                             CMD Processing               Row 1 to 2 of 20
 Command ===>                                             SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS01C             Enterprise Name : TEST

 Subsystem  : IMS01C
 Message ID : DFS554A

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 PROG_____      _____
 IMSCMD &SUBSAPPL /STA PGM &EHKVAR2_____
 _____

 _____      _____
 _____
 _____
 -------------------------------------------------------------------------------
  F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

In this example, the CMD attribute is used for the DB control region IMS01C.

```
   COMMANDS  HELP
 -----------------------------------------------------------------------------
                             CMD Processing              Row 1 to 2 of 20
 Command ===>                                            SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSA             Enterprise Name : TEST

 Subsystem  : IMSIMSA
 Message ID : DFS554A

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 TRAN_____    _____
 USER REXX &EHKVAR1 &EHKVAR3_____
 _____

 _____      _____
 _____
 _____
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

In this example, a user-written recovery routine will use the job ID and transaction ID to determine the recovery actions.

DFS810A—Define restart commands

This entry defines the restart commands required (for response to the DFS810A message) following successful completion of IMS initialization.

```
┌─ Format ──────────────────────────────────────
│ DFS810A REP=(COLD,n,command)
│         REP=(BUILDQ,n,command)
│         REP=(WARMSDBL,n,command)
└───────────────────────────────────────────────
```

Keyword and Parameter Definitions

COLD
  When the operator selects a start type of COLD on the INGREQ panel, this parameter is required to define the restart command to be issued in response to message DFS810A, and the number of RETRY attempts before determining that a reply cannot be issued.

BUILDQ
  When the operator selects a start type of BUILDQ on the INGREQ panel, this parameter is required to define the restart command to be issued in response to message DFS810A, and the number of RETRY attempts before determining that a reply cannot be issued.

WARMSDBL
  When the operator selects a start type of WARMSDBL on the INGREQ panel (causing the MSDB to be loaded during WARM start), this parameter is required to define the restart command to be issued in response to message DFS810A, and the number of RETRY attempts before determining that a reply cannot be issued.

command
  The IMS command to be issued in response to the DFS810A message. Refer to the example below for sample *command* entries.

Comments and Usage Notes

Not applicable for DB control regions.

These replies are issued in response to the DFS810A message requesting that an /NRESTART or /RESTART be entered.

Examples of Usage

```
   COMMANDS  HELP
 ------------------------------------------------------------------------------
                                 Reply Processing              Row 1 to 7 of 20
 Command ===>                                                 SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ             Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : DFS810I

 Enter the replies to be issued when this resource issues the selected message.

 Pass/      Retry Reply Text
 Selection  Count
 COLD_____  5_   /NRE CHKPT 0 FORMAT ALL DETACH_____
 BUILDQ____  5_   /NRE FORMAT RS BUILDQ_____
 WARMSDBL__  9_   /NRE MSDBLOAD_____
 _____   __   _____
 _____   __   _____
 _____   __   _____
 _____   __   _____

  F1=HELP     F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

DFS989I—Define Restart Commands (DBCTL Only)

This entry defines the restart commands required (for response to the DFS989I message) following successful completion of IMS initialization.

Format

```
DFS989I CMD=(WARMSDBL,,MVS &EHKVAR7 command_text)
        CMD=(COLD,,MVS &EHKVAR7 command_text)
```

Keyword and Parameter Definitions

COLD

When the operator selects a start type of COLD on the INGREQ input panel, this attribute entry is required to define the restart command to be issued in response to message DFS989I.

WARMSDBL

When the operator selects a start type of WARMSDBL on the INGREQ input panel, this attribute entry is required to define the restart command to be issued in response to message DFS989I.

command_text

The IMS command to be issued in response to the DFS989I message. Refer to the example below for sample *command_text* entries.

Comments and Usage Notes

These commands are issued in response to the DFS989I message requesting that a restart command be entered. For a DBCTL region the entry for message DFS810A is not required since a DBCTL region issues message DFS989I instead of DFS810A.

The variable &EHKVAR7 contains the IMS command control character.

Examples of Usage

```
   COMMANDS  HELP
  --------------------------------------------------------------------------
                              CMD Processing              Row 1 to 2 of 20
  Command ===>                                            SCROLL===> PAGE

  Entry Type : Application        PolicyDB Name  : SCENARIO
  Entry Name : IMSIMSZ           Enterprise Name : TEST

  Subsystem  : IMSIMSZ
  Message ID : DFS989I

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text
  WARMSDBL__       _____
  MVS &EHKVAR7NRE MSDBLOAD_____
  _____

  COLD_____       _____
  MVS &EHKVAR7NRE CHKPT 0 FORMAT ALL_____
  _____
   F1=HELP     F2=SPLIT    F3=END     F4=RETURN    F5=RFIND    F6=RCHANGE
   F7=UP       F8=DOWN     F9=SWAP    F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

DFS994I—Respond to checkpoint written to the IMS log

This entry defines the commands required for response to the DFS994I "*xxxx* START COMPLETED" message which follows the writing of a checkpoint to the IMS system log.

```
  ┌─ Format ─────────────────────────────────────────────────────────────────┐
    IMS Control Region:
    DFS994I REP=(ERE,n,command)
            REP=(COLD,n,command)
            REP=(BUILDQ,n,command)
            REP=(WARMSDBL,n,/STA DC)
            REP=(WARMSDBL,n,/CHE SNAPQ)
            REP=(MANUAL,n,command)
            REP=(WARM,n,/STA DC)
            REP=(WARM,n,/CHE SNAPQ)
            CMD=(WARM,,command_text)

    DB Control Region:
    DFS994I CMD=(COLD,,command_text),
            CMD=(ERE,,command_text),
            CMD=(MANUAL,,command_text),
            CMD=(WARM,,command_text)
  └────────────────────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

ERE
> When the operator selects a start type of AUTO on the INGREQ panel and IMS Automation determines that an emergency restart is necessary, this parameter is required to define the command to be issued in response to message DFS994I, and the number of retry attempts before determining that a reply cannot be issued.

COLD
> When the operator selects a start type of COLD on the INGREQ panel, this parameter is required to define a command to be issued in response to message DFS994I, and the number of retry attempts before determining that a reply cannot be issued.

BUILDQ
> When the operator selects a start type of BUILDQ on the INGREQ panel, this parameter is required to define a command to be issued in response to message DFS994I, and the number of retry attempts before determining that a reply cannot be issued.

WARMSDBL
> When the operator selects a start type of WARMSDBL on the INGREQ panel (causing the MSDB to be loaded during WARM start), this parameter is required to define a command to be issued in response to message DFS994I, and the number of retry attempts before determining that a reply cannot be issued.

MANUAL
> When the operator selects a start type of MANUAL on the INGREQ panel, he can enter an IMS restart command. This parameter is required to define the command to be issued in response to message DFS994I and to define the number of retry attempts before determining that a reply cannot be issued.

command
> The IMS command to be issued in response to the DFS994I message. Refer to the example below for sample *command* entries.

WARM
> When the operator selects a start type of AUTO on the INGREQ panel and IMS Automation determines that a warm start is necessary, this parameter is required to define the command to be issued in response to message DFS994I, and the number of retry attempts before determining that a reply cannot be issued.

Comments and Usage Notes

These IMS commands and replies are issued after the DFS994I message is received. There may be more than one reply to be issued for each type.

Based on the type of restart (warm versus emergency) and the type of startup requested (AUTO, COLD, BUILDQ, and so on), one of the reply types will be selected by IMS Automation and the IMS commands will be issued upon receipt of the DFS994 message. WARMSDBL is used to load the MSDB during a warm start. The WARM and WARMSDBL parameters coded on the DFS994 entry must be the same.

The replies may be specified in any order, but individual parameter types must be coded in the order in which you wish the commands to be issued. For example, for parameter type WARMSDBL, the /STA DC entry must be coded prior to the /CHE SNAPQ entry.

Examples of Usage

```
   COMMANDS   HELP
 ----------------------------------------------------------------------------
                               Reply Processing              Row 1 to 7 of 20
 Command ===>                                                SCROLL===> PAGE

 Entry Type : Application           PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ              Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : DFS994I

 Enter the replies to be issued when this resource issues the selected message.

 Pass/       Retry Reply Text
 Selection   Count
 ERE_____    5_   /STA DC._____
 ERE_____    5_   /CHE SNAPQ._____
 ERE_____    5_   /STA TRAN ALL._____
 COLD_____    5_   /STA DC._____
 COLD_____    5_   /CHE SNAPQ._____
 COLD_____    5_   /STA NODE ALL._____
 BUILDQ____    5_   /STA DC._____
  F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

This is the start of a set of entries for an IMS control region.

```
   COMMANDS  HELP
 ------------------------------------------------------------------------------
                               CMD Processing              Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS0C1               Enterprise Name : TEST

 Subsystem  : IMS0C1
 Message ID : DFS994I

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 COLD_____  _____
 MVS &EHKVAR7CHE_____
 _____

 ERE_____  _____
 MVS &EHKVAR7CHE_____
 _____

 MANUAL____  _____
 MVS &EHKVAR7CHE_____
 _____

 WARM_____  _____
 MVS &EHKVAR7CHE_____
 _____
  F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

These are the entries for a DB control region.

FORCE—Takeover by alternate XRF IMS subsystem

This entry defines the command to be issued when predatory takeover has occurred (the *alternate* IMS subsystem has taken control from the *active* IMS subsystem).

Format

```
FORCE CMD=(,,command_text)
```

Keyword and Parameter Definitions

command_text
    This is the command to be issued to initiate a forced shutdown.

Comments and Usage Notes
Required only for each XRF IMS subsystem.

This IMS command is issued when a predatory takeover has occurred (the *alternate* has assumed control). The command is issued on the old *active* to inform it that it is no longer the *active*.

One variable is available for use in the FORCE entry, &EHKVAR1.

Examples of Usage

```
   COMMANDS  HELP
  ----------------------------------------------------------------------------
                              CMD Processing              Row 1 to 2 of 20
  Command ===>                                            SCROLL===> PAGE

  Entry Type : Application       PolicyDB Name   : SCENARIO
  Entry Name : IMSIAOCA          Enterprise Name : TEST

  Subsystem  : IMSIAOCA
  Message ID : FORCE

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text

  _____   _____
  MVS F &EHKVAR1,STOP_____
  _____


  _____    _____
  _____
  _____
   F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

&EHKVAR1 is the jobname. This is an MVS modify command that causes IMS to terminate immediately.

HOLDQ—Issue commands at shutdown

This entry defines the commands that may be issued at the initiation of the shutdown process and subsequent to PRECHKP.

Format

```
HOLDQ CMD=(,,command_text)
     [CMD=(,,command_text)]
```

Keyword and Parameter Definitions

command_text
> Any appropriate MVS/JES command may be coded.

Comments and Usage Notes
⇒ **Not required**.

These MVS commands are issued at the very beginning of the shutdown process. Any appropriate MVS/JES commands may be coded; however, the original intent was to specify JES commands that would **hold** certain job classes that BMPs run in. This would prevent other BMPs from starting once the shutdown process was initiated.

Examples of Usage

```
  COMMANDS  HELP
 -----------------------------------------------------------------------------
                                CMD Processing              Row 1 to 2 of 20
 Command ===>                                               SCROLL===> PAGE

 Entry Type : Application         PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ            Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : HOLDQ

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text

 _____   _____
 MVS $HQ,C=123_____
 _____


 _____   _____
 _____
 _____
 F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

IMSINFO - Display Information

These commands are issued when the INGIMS REQ=INFO command is used to display the state of the selected IMS Control or DBCTL region. The commands are issued via the IMS subsystem ID on an MVS EMCS console and the resulting messages are either displayed on the INGIMS panel or written to the users NetView console.

For further information about the INFO request, see the description of the INGIMS command in *System Automation for OS/390 Operator's Commands*.

```
┌─ Format ──────────────────────────────────────────────────────────────┐
│  IMSINFO , IMSCMD = (description, IMS command)                          │
└────────────────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

description
> The description is text that will be placed before the output of the IMS command. This can be used to identify the command output in the output stream. The description can be any string, but must be enclosed in quotes.

IMS command
> The IMS command is the command to be executed. This command will be appended to the IMS subsystem ID and issued as an MVS command to an EMCS console. The output will be collected and displayed. The command can be any valid IMS DBCTL or Control region command. The command must be enclosed in quotes.

Comments and Usage Notes
This policy is required for correct operation of the INGIMS command and also PF10 of the DISPINFO panel.

Examples of Usage

```
   COMMANDS  HELP
 ----------------------------------------------------------------------------
                             User Defined Data              Row 1 to 7 of 20
 Command ===>                                             SCROLL===> PAGE

 Entry Type : Application         PolicyDB Name   : SA22_KEYPLEX
 Entry Name : IMS711C4            Enterprise Name : KEY1FAMILY

 Subsystem  : IMS711C4
 Message ID : IMSINFO

 To change keyword-data pair, specify the following:

 Keyword
 Data
 IMSCMD
 ('ACTIVE STATE','DIS A')
 IMSCMD
 ('Comms','DIS DC')




  F1=HELP       F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP         F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

IPS—XRF IMS takeover commands

This entry defines the commands that may be issued during the XRF IMS takeover process.

Format

```
IPS CMD=(,,command_text)
    [CMD=(,,command_text)]
```

Keyword and Parameter Definitions

command_text
> Any appropriate MVS command may be coded.

Comments and Usage Notes

⇒ **Not required**. Applicable to XRF IMS environment only.

These MVS commands are issued during an XRF IMS takeover process on the system where the new *active* IMS subsystem resides. Due to the change in workload characteristics resulting from an XRF IMS takeover, it may be advantageous to code optional MVS commands to make adjustments to MVS processing.

Examples of Usage

```
   COMMANDS  HELP
  -----------------------------------------------------------------------------
                                 CMD Processing              Row 1 to 2 of 20
  Command ===>                                               SCROLL===> PAGE

  Entry Type : Application          PolicyDB Name   : SCENARIO
  Entry Name : IMSIAOCA            Enterprise Name : TEST

  Subsystem  : IMSIAOCA
  Message ID : IPS

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text
  _____ _____
  MVS SET IPS=02_____
  _____


  _____    _____
  _____
  _____
   F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

OLDS—Define recovery criteria for OLDS

The OLDS keyword defines automation settings for online data sets (OLDS). IMS Automation monitors the OLDS at regular intervals and notifies operators before the IMS fails. Using the OLDS entry, you specify the minimum number of OLDS that must be available at all times. If the number of available OLDS drops below that minimum, IMS Automation starts the OLDS you designate as spares. If the number of available OLDS exceeds the minimum, IMS Automation will stop unneeded OLDS. Please read the "Comments and Usage Notes" on page 95 carefully.

During normal IMS operations, the archive jobs copy data from the OLDS to the system log data sets (SLDS) to make the OLDS available for more data. When archive jobs fail, the OLDS fill up and are no longer available. IMS Automation monitors how long it takes the archive jobs to execute and notifies operators if the archive jobs are taking longer than expected to run. Additionally, IMS Automation checks that the number of OLDS data sets with an 'OTHER-STS' of 'BACKOUT' has not exceeded the value specified in the BACKOUT operand. By monitoring the archive jobs, IMS Automation helps operators detect possible problems early so they can be corrected before the OLDS fill and IMS stops.

Attention: Investigate any problems with OLDS immediately and correct. Failure to do so can cause IMS to stop.

```
  Format
 OLDS[MINIMUM=nn]
     [SPARES=(nn,nn...)]
     [ARCHIVETIME=hh:mm:ss]
     [RETRYCNT=n]
     [BACKOUT=nn]
```

Keyword and Parameter Definitions

MINIMUM=nn
> The minimum number of OLDS that must be available at all times. The default minimum is 50% of the *normal number* of OLDS. The normal number of OLDS display when you issue the DISPLAY OLDS command (DIS OLDS). The normal number can not be less than three and includes all OLDS that IMS started at initialization or that operators started. OLDS that are defined but are not listed in the response to the DISPLAY OLDS command are not included in calculating the normal number of OLDS.

SPARES=(nn,nn...**)**
> The spares are OLDS that IMS Automation activates when the number of available OLDS drops below the minimum. The names for the spares are the two-digit numbers taken from the end of the ddname. For example, DFSOLP99 is the spare named 99. Be sure that the names of the spares match the names of existing OLDS.

ARCHIVETIME=hh:mm:ss
> The archive time is the maximum length of time archive jobs take to run. Set this timer to be longer than the archive jobs take to execute. The default setting is 00:10:00 (10 minutes).

RETRYCNT=n
The retry count is the number of times that IMS Automation will attempt to acquire an outstanding reply ID when activating or deactivating a spare OLDS. The default is 5 retries.

BACKOUT=nn
The maximum number of OLDS that can have an OTHER-STS of BACKOUT. Set this number to match the total number of acceptable OLDS data sets with an OTHER-STS of BACKOUT.

Comments and Usage Notes

1. Make sure that the OLDS you designate as spares exist.

   Note: To define spares as dynamically allocated additional OLDS, preallocate and catalog candidate data sets and specify data set names using the dynamic allocation macro, DFSMDA. Provide DFSMDA members for all OLDS. For information on using DFSMDA, refer to the *IMS/ESA Utilities Reference*.

2. Do not include the spare OLDS in DFSVSM00, which is the member that tells IMS which data sets to start when IMS initializes. IMS Automation will stop any spare OLDS and remove them from SDF if the minimum number of OLDS are available.

3. If you activate a spare OLDS manually while the number of available OLDS is equal to or above the minimum, IMS Automation removes the spare from IMS and deletes any SDF entries for the spare.

   Note: IMS Automation does *not* send notifications when it stops an OLDS that is designated for use only as a spare.

4. If you define new OLDS, you must stop and restart IMS to use these settings.

5. For IMS Automation to monitor OLDS, you must define an OLDS entry.

6. The value of OLDS BACKOUT can not exceed the total number of OLDS data sets defined to the IMS subsystem. The value should represent an acceptable maximum number of OLDS data sets with an OTHER-STS of BACKOUT.

OLDS

Examples of Usage

```
   COMMANDS  HELP
 ------------------------------------------------------------------------------
                              User Defined Data            Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : OLDS

 To change keyword-data pair, specify the following:

 Keyword
 Data
 MINIMUM_____
 5_____
 _____

 SPARES_____
 (97,98,99)_____
 _____

 ARCHIVETIME_____
 00:20:00_____
 _____

 RETRYCNT_____
 7_____
 _____

 BACKOUT_____
 2_____
 _____
 ------------------------------------------------------------------------------
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

For this example, the minimum number of OLDS that must be kept available at all
times is five. If the number of available OLDS is less than five, IMS Automation
starts one of the spares (DFSOLP97, DFSOLP98, or DFSOLP99). If IMS Automation
does not receive an outstanding reply ID within seven tries, no further attempts
are made. IMS Automation also sends notifications to operators if the archive jobs
take longer than 20 minutes to execute or the total number of OLDS data sets with
an OTHER-STS of BACKOUT exceeds a maximum of 2.

POSTCHKP—Issue commands after IMS shutdown checkpoint

This entry defines the commands that may be issued immediately after the IMS shutdown checkpoint has been issued.

```
┌─ Format ──────────────────────────────────────────────┐
  POSTCHKP[CMD=(,,command_text)]
          [CMD=(,,command_text)]
          [REP=(,n,ims_command)]
          [REP=(,n,ims_command)]
└───────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

command_text
    Any appropriate command may be coded.

ims_command
    Any appropriate IMS command may be coded.

Comments and Usage Notes
⇒ **Not required**.

For DB control regions, use the **CMD=** syntax only.

Attention: Although replies are supported, their use at this point in the processing of an IMS is not recommended. Because POSTCHKP occurs after the checkpoint call has been issued, it is likely that no outstanding reply is available. Improper use of the reply function with POSTCHKP can cause performance problems.

These commands (CMD=) and IMS replies (REP=) are issued just after the IMS shutdown checkpoint has been issued. There may be multiple commands and IMS replies.

Two variables are available for use in the commands:

&EHKVAR1   This is the value of the VTAM IMS APPLID.

&EHKVAR2   This is the IMS subsystem ID (*subid*) as specified in the **IMSID** field of the IMS ENVIRON policy item.

POSTCHKP

Examples of Usage

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                              CMD Processing              Row 1 to 2 of 20
 Command ===>                                             SCROLL===> PAGE

 Entry Type : Application         PolicyDB Name  : SCENARIO
 Entry Name : IMSIMSZ            Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : POSTCHKP

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 _____  _____
 MVS $AQ,C=123_____
 _____

 _____  _____
 MVS S IMSJOB_____

 _____
  F1=HELP     F2=SPLIT     F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN      F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

PRECHKP—Issue commands prior to IMS shutdown checkpoint

This entry defines the commands that may be issued immediately prior to the IMS shutdown checkpoint being issued but prior to HOLDQ.

```
 ┌─ Format ────────────────────────────────────────────────────────
 │ PRECHKP[CMD=(,,command)]
 │        [CMD=(,,command)]
 │        [REP=(,n,ims_command)]
 │        [REP=(,n,ims_command)]
 └─────────────────────────────────────────────────────────────────
```

Keyword and Parameter Definitions

command_text
> Any appropriate command may be coded.

ims_command
> Any appropriate IMS command may be coded.

Comments and Usage Notes
⇒ **Not required**.

For DB control regions, use the **CMD=** syntax only.

These commands (CMD=) and IMS replies (REP=) are issued just prior to the IMS shutdown checkpoint being issued. There may be multiple commands and IMS replies.

Examples of Usage

```
   COMMANDS  HELP
  ----------------------------------------------------------------------------
                              CMD Processing               Row 1 to 2 of 20
   Command ===>                                            SCROLL===> PAGE

   Entry Type : Application        PolicyDB Name  : SCENARIO
   Entry Name : IMSIMSZ            Enterprise Name : TEST

   Subsystem  : IMSIMSZ
   Message ID : PRECHKP

   Enter commands to be executed when resource issues the selected message.

   Pass/Selection Automated Function/'*'
   Command Text

   _____  _____
   MVS $HQ,123_____
   _____


   _____  _____
   MVS S IMSJOB_____
   _____
    F1=HELP     F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
    F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

PRECHKP

```
  COMMANDS  HELP
 -------------------------------------------------------------------------------
                             Reply Processing              Row 1 to 7 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application         PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZ            Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : PRECHKP

 Enter the replies to be issued when this resource issues the selected message.

 Pass/      Retry Reply Text
 Selection  Count
 _____    5_   /STO TRAN ALL_____
 _____    5_   /STO NODE ALL_____
 _____   __    _____
 _____   __    _____
 _____   __    _____
 _____   __    _____
 _____   __    _____

  F1=HELP    F2=SPLIT    F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP      F8=DOWN     F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

This is an example for an IMS control region where CMD and REP attributes are
coded.

RECONS—Set monitoring interval for RECONS

The RECONS keyword turns on active monitoring at regular intervals for RECONS. RECONS are critical for IMS recovery control. At the interval specified on the RECONS entry, IMS Automation checks to make sure that a spare RECON is available. If no spare is available, IMS Automation sends a notification to the operators.

IMS Automation also does the following RECONS automation:
- Informs operators when there has been a switch from one RECON to another
- Checks for exception conditions, such as I/O errors

For these automation actions, which occur in response to error messages, you do not need to code any entries in the policy database.

Format

```
RECONS MONITOR=hh:mm:ss
       [RETRY=nn DELAY=nn]
```

Keyword and Parameter Definitions

MONITOR=*hh:mm:ss*
> The setting on the MONITOR keyword determines how often IMS Automation checks the RECONS to make sure a spare is available.

RETRY=*nn*
> Specify a value between 0 and 99. This value determines how many times RECONS monitoring will retry the "/RMLIST DBRC='RECON STATUS'." command after finding the spare RECONS data set in UNAVAILABLE status before issuing either the EVI830W or EVI832W error message.
>
> A value of 0 (default) indicates no retry is to be done. If a non-zero value is specified for the RETRY parameter then a non-ZERO value is required on the DELAY parameter.

DELAY=*nn*
> Specify a value between 0 and 59 (seconds). This value determines how long RECONS monitoring will wait before reissuing the '/RMLIST DBRC='RECON STATUS'." command as determined by the RETRY= parameter.
>
> A value of 0 (default) indicates no retry is to be done. If a non-zero value is specified for the DELAY parameter then a non-ZERO value is required on the RETRY parameter.

Comments and Usage Notes

1. IMS Automation does some RECONS checking even if no RECONS keyword is defined:
   - IMS Automation checks RECONS at IMS initialization by invoking the EVIECR04 module in the MAT.
   - IMS Automation notifies operators when there is a switch from one RECON to another.
2. For IMS Automation to actively monitor the RECONS for spares, you must specify a monitoring interval on the RECONS keyword.
3. Occasionally the spare IMS RECONS data set returns a status of UNAVAILABLE which is only temporary. With the RETRY and DELAY parameters, the user has the option to reissue the status command to determine

whether the UNAVAILABLE status is temporary or permanent, rather than issue the EVI830W or EVI832W error message immediately.

Examples of Usage

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                              User Defined Data            Row 1 to 2 of 20
 Command ===>                                                SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS10AA             Enterprise Name : TEST

 Subsystem  : IMS10AA
 Message ID : RECONS

 To change keyword-data pair, specify the following:

 Keyword
 Data
 MONITOR_____
 00:15:00_____
 _____

 RETRY_____
 3_____
 _____

 DELAY_____
 5_____
 _____

  F1=HELP       F2=SPLIT      F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP         F8=DOWN       F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

In this example, IMS Automation checks the RECONS for spares every 15 minutes. When the spare IMS RECONS data set returns a status of UNAVAILABLE the status command is reissued three times every five seconds.

RELEASEQ—Issue commands after shutdown completes

This entry defines the commands that may be issued after IMS shutdown completion.

Format

```
RELEASEQ CMD=(,,command_text)
         [CMD=(,,command_text)]
```

Keyword and Parameter Definitions

command_text
> Any appropriate MVS command may be coded.

Comments and Usage Notes
⇒ **Not required**.

These MVS commands are issued after an IMS control region has completed shutdown. Any appropriate MVS/JES commands may be coded, however, the original intent was to perform the reverse function of the HOLDQ commands (refer to "HOLDQ—Issue commands at shutdown" on page 91), that is, to release the job classes held during the shutdown process.

Examples of Usage

```
   COMMANDS  HELP
  ----------------------------------------------------------------------------
                               CMD Processing              Row 1 to 2 of 20
  Command ===>                                             SCROLL===> PAGE

  Entry Type : Application         PolicyDB Name   : SCENARIO
  Entry Name : IMSIMSZ            Enterprise Name : TEST

  Subsystem  : IMSIMSZ
  Message ID : RELEASEQ

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text
  _____   _____
  MVS $AQ,C=123_____
  _____


  _____   _____
  _____
  _____
   F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
   F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

RESTARTABORT—Emergency restart commands

This entry defines the emergency restart commands required (for response to the DFS166 and DFS0618A messages) following an abort termination of the restart process.

```
┌─ Format ─────────────────────────────────────────────────────────┐
│                                                                    │
│  IMS Control Region:                                               │
│  RESTARTABORT REP=(OVERRIDE,n,command_text)                        │
│               REP=(BACKUP,n,command_text)                          │
│               REP=(ERE,n,command_text)                             │
│                                                                    │
│  DB Control Region:                                                │
│  RESTARTABORT CMD=(OVERRIDE,,command_text)                         │
│               CMD=(ERE,,command_text)                              │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

OVERRIDE

When message DFS618A is issued, the OVERRIDE reply will be issued for a non-XRF IMS subsystem, or an XRF IMS *active* subsystem when its partner subsystem is not currently active.

BACKUP

When message DFS618A is issued, the BACKUP reply will be issued for an XRF IMS alternate subsystem when its partner is currently active.

ERE

When message DFS166 is issued for an IMS subsystem and a valid checkpoint was not found during a restart, the ERE reply will be issued.

command

Code the *command* for OVERRIDE, BACKUP, and ERE replies as specified in the example below.

Comments and Usage Notes

This reply is issued in response to the RESTARTABORT for either the DFS0618A or DFS166 message.

Examples of Usage

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                              Reply Processing                Row 1 to 7 of 20
 Command ===>                                                 SCROLL===> PAGE

 Entry Type : Application           PolicyDB Name    : SCENARIO
 Entry Name : IMSIMSZ              Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : RESTARTABORT

 Enter the replies to be issued when this resource issues the selected message.

 Pass/       Retry Reply Text
 Selection   Count
 OVERRIDE__  5_    /ERE OVERRIDE._____
 BACKUP____  5_    /ERE BACKUP._____
 ERE_____  5_    /ERE._____
 _____    __    _____
 _____    __    _____
 _____    __    _____
 _____    __    _____

  F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

This example shows an entry for an IMS control region.

SHUTTYPES—Issue commands at operator shutdown

This entry specifies the commands that are associated with the so-called shutdown options. One such an option must be specified in the shutdown commands under the SHUTDOWN policy item; this is the default option. You can override the default by specifying another shutdown option in the **Appl Parms** field of the INGREQ input panel. For INGREQ, see Chapter 8, "Starting and Stopping Resources" on page 163.

```
┌─ Format ──────────────────────────────────────────────────────────┐
│  SHUTTYPES REP=(shutoption,n,command),                             │
│             .                                                      │
│             .                                                      │
│             .                                                      │
│             REP=(shutoption,n,command),                            │
│             CMD=(NODUMP,,command_text),                            │
│             CMD=(DUMP,,command_text)                               │
│             CMD=(PURGE,,command_text)                              │
│             CMD=(FREEZE,,command_text)                             │
└────────────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

shutoption
> One SHUTTYPES entry must be specified for every IMS control region, with four REP attributes (each specifying a command for one of the shutdown option DUMPQ, BACKUP, FREEZE or PURGE), and two CMD attributes, one with DUMP and the other with NODUMP as its first value.

command,*command_text*
> The IMS command to be issued, related to the *shutoption* value specified in the REP or CMD attribute of the SHUTTYPES entry. The variable &EHKVAR1 may be used in the command; its value will be the IMS control region jobname.

Comments and Usage Notes

The IMS command issued depends on the shutdown type (NORM, IMMED or FORCE) selected for the shutdown request and on the default shutdown options specified in the commands for the different phases of the SHUTDOWN policy item; operator selection (optional) of a shutdown option on the INGREQ panel will override the default specification.

The *shutoption* parameter coded in the commands for the shutdown phases corresponds to the *shutoption* value coded in the REP or CMD attributes of the SHUTTYPES entry. For instance, consider the example below. If the operator selects the shutdown type NORM and does not specify a shutdown option, then the /CHE DUMPQ command will be issued because the command for the NORM phase contains the DUMPQ shutdown option, and this shutdown option is associated with the /CHE DUMPQ command in the SHUTTYPES entry.

If, however, the operator overrides DUMPQ with the FREEZE shutdown option, then the command that is associated with FREEZE in the SHUTTYPES entry will be issued, that is, the /CHE FREEZE command.

For shutting down a subsystem from the operator interface, refer to "Shutdown" on page 165.

Examples of Usage

```
   COMMANDS  HELP
-----------------------------------------------------------------------------
                             Reply Processing                Row 1 to 7 of 20
Command ===>                                                 SCROLL===> PAGE

Entry Type : Application           PolicyDB Name   : SCENARIO
Entry Name : IMSIMSZ              Enterprise Name : TEST

Subsystem  : IMSIMSZ
Message ID : SHUTTYPES

Enter the replies to be issued when this resource issues the selected message.

Pass/      Retry Reply Text
Selection  Count
DUMPQ_____  5_   /CHE DUMPQ_____
BACKUP____  5_   /CHE BACKUP_____
FREEZE____  5_   /CHE FREEZE_____
PURGE_____  5_   /CHE PURGE_____
_____  __    _____
_____  __    _____
_____  __    _____

 F1=HELP     F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

This is the REP part of the SHUTTYPES entry for an IMS control region.

SNAPQ—Issue SNAPQ checkpoint command

This entry specifies the IMS command to be issued when a SNAPQ checkpoint is required.

Format

```
SNAPQ REP=(,n,/CHE SNAPQ)
```

Keyword and Parameter Definitions

/CHE SNAPQ

This is the command issued when a SNAPQ checkpoint is required. Code exactly as specified.

Comments and Usage Notes

This IMS command is issued when a SNAPQ checkpoint is required.

Code this entry as described above (specify number of retry attempts).

Examples of Usage

```
 COMMANDS  HELP
--------------------------------------------------------------------------
                            Reply Processing              Row 1 to 7 of 20
 Command ===>                                             SCROLL===> PAGE

 Entry Type : Application        PolicyDB Name  : SCENARIO
 Entry Name : IMSIMSZ           Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : SNAPQ

 Enter the replies to be issued when this resource issues the selected message.

 Pass/      Retry Reply Text
 Selection  Count
 _____   5_   /CHE SNAPQ_____
 _____   __   _____
 _____   __   _____
 _____   __   _____
 _____   __   _____
 _____   __   _____
 _____   __   _____
  F1=HELP    F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP      F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

START—Define start commands

START defines IMS start commands for XRF dependent regions. It is only needed when the **External Startup** field of the AUTOMATION INFO policy item is not set to ALWAYS for the XRF dependent region.

Format

```
START CMD=(,,MVS S job_name)
```

Keyword and Parameter Definitions

job_name
    The name of the job specified for the application in the policy database.

Comments and Usage Notes

If you want to use the START message ID, you must specify EVIEI00T as the startup command in the STARTUP policy item.

Examples of Usage

```
  COMMANDS   HELP
 -----------------------------------------------------------------------------
                               CMD Processing              Row 1 to 2 of 20
  Command ===>                                               SCROLL===> PAGE

  Entry Type : Application        PolicyDB Name   : SCENARIO
  Entry Name : IMS722F1           Enterprise Name : TEST

  Subsystem  : IMS722F1
  Message ID : START

  Enter commands to be executed when resource issues the selected message.

  Pass/Selection Automated Function/'*'
  Command Text

 _____    _____
 MVS S IMS722F1_____
 _____

 _____    _____
 _____
 _____
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

STOPBMPREGION—Stop batch message regions

This keyword serves to define commands to stop batch message regions. It must be specified under the MESSAGES/USER DATA item of the control region associated with the BMP region. This entry is only used when the **External Shutdown** field of the **Automation Info** policy item is not set to ALWAYS for the batch message region.

In this case, shutdown commands must be defined for the batch message region in its SHUTDOWN policy item. These commands are passed a shutdown type which corresponds to a **Pass/Selection** value of the STOPBMPREGION entry. When one of these commands is issued, it invokes in its turn the command of that attribute instance of STOPBMPREGION that contains the corresponding first value. For an example of this mechanism, see "Other Region Types" on page 46.

Format

```
IMS Control Region:
STOPBMPREGION,
REP=(NORMAL,n,/STOP REGION &EHKVAR1)
REP=(ABEND,n,/STOP REGION &EHKVAR1 ABDUMP)
[REP=(CANCEL,n,/STOP REGION &EHKVAR1 ABDUMP)]
REP=(CANCEL,n,/STOP REGION &EHKVAR1 CANCEL)

DB Control Region:
STOPBMPREGION
CMD=(NORMAL,,command_text),
CMD=(ABEND,,command_text),
[CMD=(CANCEL,,command_text),]
CMD=(CANCEL,,command_text),
```

Keyword and Parameter Definitions

n   The number of retries.

Comments and Usage Notes

1. The first values of the CMD and REP attributes correspond to the shutdown types as follows:
   - NORMAL equates to the shutdown type NORM.
   - ABEND equates to the shutdown type IMMED.
   - CANCEL equates to the shutdown type FORCE.
2. IMS requires that a /STOP REGION *xxxx* ABDUMP be issued before a /STOP REGION *xxxx* CANCEL. Therefore, it is recommended that two CANCEL entries be coded in the STOPBMPREGION entry, the first specifying ABDUMP and the second specifying CANCEL (see the example below). Coding two CANCEL statements (as shown below) means there is some redundancy during normal SHUTDOWN escalation because the /STOP REGION *xxxx* ABDUMP will be issued for SHUTIMMED/ABEND before escalation procedes to SHUTFORCE/CANCEL.

Examples of Usage

```
   COMMANDS  HELP
 ------------------------------------------------------------------------------
                                 CMD Processing              Row 1 to 2 of 20
 Command ===>                                                SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIMSZA            Enterprise Name : TEST

 Subsystem  : IMSIMSZA
 Message ID : STOPBMPREGION

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 NORMAL____        _____
 MVS &EHKVAR7STOP REG &EHKVAR1_____
 _____

 ABEND_____        _____
 MVS &EHKVAR7STOP REG &EHKVAR1 ABDUMP_____
 _____
 CANCEL____        _____
 MVS &EHKVAR7STOP REG &EHKVAR1 ABDUMP_____
 _____
 CANCEL____        _____
 MVS &EHKVAR7STOP REG &EHKVAR1 CANCEL_____
 _____
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

This is an example for a DB control region.

STOPFPREGION—Stop fast path regions

This keyword serves to define commands to stop fast path regions. It must be specified under the MESSAGES/USER DATA item of the *control* region associated with the FP region. This entry is only used when the **External Shutdown** field of the **Automation Info** policy item is not set to ALWAYS for the fast path region.

In this case, shutdown commands must be defined for the fast path region in its SHUTDOWN policy item.These commands must be passed a shutdown type which corresponds to a **Pass/Selection** value of the STOPFPREGION entry. When one of these commands is issued, it invokes in its turn the command of that attribute instance of STOPFPREGION that contains the corresponding first value. For an example of this mechanism, see "Other Region Types" on page 46.

```
Format
STOPFPREGION
REP=(NORMAL,n,/STOP REGION &EHKVAR1)
REP=(ABEND,n,/STOP REGION &EHKVAR1 ABDUMP)
[REP=(CANCEL,n,/STOP REGION &EHKVAR1 ABDUMP)]
REP=(CANCEL,n,/STOP REGION &EHKVAR1 CANCEL)
```

Keyword and Parameter Definitions

n   The number of retries

Comments and Usage Notes

1. Not applicable for DB control regions.
2. The first values of the CMD and REP attributes correspond to the shutdown types as follows:
   - NORMAL equates to the shutdown type NORM.
   - ABEND equates to the shutdown type IMMED.
   - CANCEL equates to the shutdown type FORCE.
3. IMS requires that a /STOP REGION *xxxx* ABDUMP be issued before a /STOP REGION *xxxx* CANCEL. Therefore, it is recommended that two CANCEL entries be coded in the STOPFPREGION entry, the first specifying ABDUMP and the second specifying CANCEL. Coding two CANCEL statements means there is some redundancy during normal SHUTDOWN escalation because the /STOP REGION *xxxx* ABDUMP will be issued for SHUTIMMED/ABEND before escalation procedes to SHUTFORCE/CANCEL.

STOPREGION—Stop IMS dependent message region

This keyword serves to define commands to stop dependent message regions. It must be specified under the MESSAGES/USER DATA item of the *control* region associated with the dependent message region. This entry is only used when the **External Shutdown** field of the **Automation Info** policy item is not set to ALWAYS for the dependent region.

In this case, shutdown commands must be defined for the dependent message region in its SHUTDOWN policy item. These commands must be passed a shutdown type which corresponds to the first value of an attribute of the STOPREGION entry. When one of the shutdown commands is issued, it invokes in its turn the command of that attribute instance of STOPREGION that contains the corresponding first value. For an example of this mechanism, see "Other Region Types" on page 46.

```
┌─ Format ──────────────────────────────────────────────────────────┐
│  STOPREGION REP=(NORMAL,n,/STOP REG  &EHKVAR1)                      │
│             REP=(ABEND,n,/STOP REG &EHKVAR1 ABDUMP &EHKVAR2)        │
│             [REP=(CANCEL,n,/STOP REG &EHKVAR1 ABDUMP &EHKVAR2)]     │
│             REP=(CANCEL,n,/STOP REG &EHKVAR1 CANCEL)                │
└────────────────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

n   This entry specifies that IMS Automation will retry the operation *n* times.

ABEND

This entry specifies the IMS command required to stop an IMS dependent message region when the normal stop command fails.

CANCEL

This entry specifies the IMS command required to stop an IMS dependent message region when NORMAL and ABEND stop commands fail.

NORMAL

This entry specifies the IMS command required to stop an IMS dependent message region as a normal part of the shutdown process.

Comments and Usage Notes

1. Not applicable for DB control regions.
2. Two variables are available, &EHKVAR1 and &EHKVAR2. &EHKVAR1 is a variable containing the IMS region number of the dependent region that is being stopped. &EHKVAR2 is the name of the active transaction.
3. Code this entry exactly as described above.
4. IMS requires that a /STOP REGION *xxxx* ABDUMP be issued before a /STOP REGION *xxxx* CANCEL. Therefore, it is recommended that two CANCEL entries be coded in the STOPREGION entry, the first specifying ABDUMP and the second specifying CANCEL. Coding two CANCEL statements means there is some redundancy during normal SHUTDOWN escalation because the /STOP REGION *xxxx* ABDUMP will be issued for SHUTIMMED/ABEND before escalation procedes to SHUTFORCE/CANCEL.

TCO—Issue commands for time-driven procedures

This entry allows commands to be issued to initiate, change, start, or stop
time-driven procedures for any IMS operation.

```
┌─ Format ─────────────────────────────────────────────────────
│
│  TCO REP=(INIT,,DFSTCF LOAD DFSTCF  .),
│      REP=(SPEC,,DFSTCF LOAD &EHKVAR1  .),
│      REP=(START,,/START LTERM DFSTCF1  .),
│      REP=(STOP,,/PSTOP LTERM DFSTCFI   .)
│
```

Keyword and Parameter Definitions

INIT
   This entry specifies the IMS command required to start the initial time driven
   procedure DFSTCF.

SPEC
   This entry specifies the IMS command required to change from the current
   TCO script to a script name entered from the operator interface.

START
   This entry specifies the IMS command required to start TCO.

STOP
   This entry specifies the IMS command required to stop TCO.

Comments and Usage Notes

1. For more information on TCO, refer to the *IMS Operations Guide*.

2. &EHKVAR1 is the name entered from the IMS Automation operator interface
   TCO function.

TCOMEMBERS—Define TCO Members

This keyword serves to create a list of members that appear in a pop-up IMS Automation TCO Member Load panel. The panel is shown in the example below:

```
 EVIKMT10      IMS Automation: TCO Member Load              Page: 1 of 1
                                                            Date: 05/23/02
 Resource/Domain  => IMS711C4/APL/KEY1                      Time: 13:01
                                                            Domain: IPSNM

 TCO Status . . . . . . . : Available
        +--------------------------------------------------------------+
 Member | Select one of the user supplied TCO members                  |
        | _ STRTLNES        START LINES                                |
 Alterna| _ ASGNTRAN        ASSIGN TRANSACTIONS                        |
        | _ DISPROG         DISPLAY ACTIVE PROGRAMS                    |
 Maximum|                                                              |
        |                                                              |
        |                                                              |
        |                                                              |
        |                                                              |
        |                                                              |
        |                                                              |
        |                                                              |
        |  F1=Help    F3=Cancel                                        |
        +--------------------------------------------------------------+
 Command ===>
 F1=Help    F2=End      F3=Return   F4=IMS Menu            F6=Roll
```

Format

```
 TCOMEMBERS NAME=(membername,'comment')
```

Keyword and Parameter Definitions

membername
> This is the 8-character name of the member previously defined in the IMS TCO member library. The library is associated with the DFSTCF DD statement in the IMS start up JCL.

comment
> This is a comment, up to 20 characters long.

Comments and Usage Notes
For more information on TCO, refer to Chapter 9, "TCO Functions" on page 167.

TCOMEMBERS

Examples of Usage

```
   COMMANDS  HELP
 --------------------------------------------------------------------------
                              User Defined Data            Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMS01A               Enterprise Name : TEST

 Subsystem  : IMS01A
 Message ID : TCOMEMBERS

 To change keyword-data pair, specify the following:

 Keyword
 Data
 NAME_____
 (STRTLNES,'START LINES)_____
 _____


 NAME_____
 (ASGNTRAN,'ASSIGN TRANSACTIONS')_____
 _____
   F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
   F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

TPABEND—Dependent region abend

This entry is required for a dependent region abend with a U0002 return code.

Format

```
TPABEND CODE=(U0002,,,ABENDING)
```

Comments and Usage Notes

If a dependent region abend with a U0002 return code occurs, all active MPP and
BMP dependent regions are forced to terminate, and are prevented from restarting
before the control region goes to abend status.

Examples of Usage

```
  COMMANDS  HELP
 ----------------------------------------------------------------------------
                             Code Processing              Row 1 to 6 of 21
 Command ===>                                             SCROLL===> PAGE

 Entry Type : Application        PolicyDB Name  : SCENARIO
 Entry Name : IMSIMSZ           Enterprise Name : TEST

 Subsystem  : IMSIMSZ
 Message ID : TPABEND
 Enter the value to be passed to the calling CLIST when this resource
 issues the selected message and the following codes are contained in
 the message.

    Code 1          Code 2          Code 3            Value Returned
 U0002_____ _____ _____ ABENDING_____
 _____ _____ _____ _____
 _____ _____ _____ _____
 _____ _____ _____ _____
 _____ _____ _____ _____
 _____ _____ _____ _____
 F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

UNLKAVM—Reply to message AVM005A

This command is used in reply to the outstanding reply of message AVM005A to inform the alternate that input/output prevention is complete.

Format

```
UNLKAVM CMD=(,,MVS R &EHKVAR1,UNLOCK)
```

Comments and Usage Notes

Required only for an XRF IMS subsystem.

This IMS command is issued as part of the XRF takeover process. This entry will normally be specified as described above.

The variable &EHKVAR1 contains the outstanding reply that is associated with the AVM manager AVM005E.

Examples of Usage

```
   COMMANDS  HELP
 ----------------------------------------------------------------------------
                               CMD Processing              Row 1 to 2 of 20
 Command ===>                                              SCROLL===> PAGE

 Entry Type : Application        PolicyDB Name   : SCENARIO
 Entry Name : IMSIA0CA           Enterprise Name : TEST

 Subsystem  : IMSIA0CA
 Message ID : UNLKAVM

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text

 _____     _____
 MVS R &HKVAR1,UNLOCK_____
 _____


 _____     _____
 _____
 _____
  F1=HELP     F2=SPLIT    F3=END     F4=RETURN   F5=RFIND    F6=RCHANGE
  F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

UNLOCK—Input/output prevention completed

This command informs the alternate XRF subsystem that input/output prevention is complete.

Format

```
UNLOCK REP=(,n,/UNLOCK SYSTEM.)
```

Keyword and Parameter Definitions

n   specifies the number of retry attempts.

Comments and Usage Notes
Required only for an XRF or FDR IMS subsystem.

This IMS command is issued as part of the XRF takeover process. This entry will normally be specified as described above, where *n* is the number of retry attempts.

UP—Initiate FDR Startup

This entry defines the Automation procedure that will be run to initate the startup of the FDR region when the IMS Control region changes to 'UP' status.

Format
```
UP CMD=(,,EVISTFDR)
```

Comments and Usage Notes

This is only applicable to FDR-enabled IMS regions.

When the IMS Control region status changes to 'UP', Automation will run REXX procedure EVISTFDR to initiate the startup of the FDR region on the NetView domain as specified in the Automation policy.

You can avoid using the UP keyword by defining the relations between the FDR region and the control region as described in "Defining an FDR Environment" on page 40.

VTAMTERMS—Release terminal sessions

This entry specifies the command IMS Automation will issue to release terminal sessions with IMS, accelerating IMS shutdown and XRF IMS takeover.

```
  Format
VTAMTERMS
CMD=(SHUTDOWN,,V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND)
CMD=(TAKEOVER,,V NET,TERM,LU=&EHKVAR1,TYPE=FORCE)
```

Keyword and Parameter Definitions

SHUTDOWN

This entry specifies the command IMS Automation will issue to release terminal sessions with IMS and accelerate IMS shutdown.

TAKEOVER

This entry specifies the command IMS Automation will issue to release terminal sessions with IMS to accelerate XRF IMS takeover.

Comments and Usage Notes

Required for an XRF IMS subsystem.

This MVS command is issued to release terminal sessions with IMS to accelerate IMS shutdown and XRF IMS takeover.

The variable &EHKVAR1 contains the actual VTAM LU name.

$PI—Stop BMP tasks on active subsystem

This entry specifies the commands required to stop BMP tasks on the old *active* IMS subsystem's CPC during takeover, so that the BMP workload can be transferred to the *alternate* IMS subsystem's (new *active*) CPC.

Format

```
$PI CMD=(,,MVS $PIm-n)
```

Keyword and Parameter Definitions

m-n
> The range of initiator numbers to be stopped.

Comments and Usage Notes

⇒ **Not required**. Applicable to an XRF environment only.

These MVS/JES commands are issued to stop BMP initiators on the old *active* on a takeover so that BMP workload can be transferred to the system where the new *active* is running. There is no restriction on the new and old active systems being in the same domain.

$SI—Issue start initiator commands after XRF takeover

These start initiator commands are issued after an XRF takeover is complete, immediately following the $TI commands.

Format

```
$SI CMD=(,,MVS $SIm-n)
```

Keyword and Parameter Definitions

m-n
> The range of initiator numbers to be started.

Comments and Usage Notes

⇒ **Not required**. Applicable to an XRF environment only.

These MVS/JES commands are issued after an XRF takeover is complete. They are issued on the system of the new *active*. While you can code any MVS/JES commands for this entry, it is specifically intended as a means to adjust the initiator structure after an XRF takeover. There is no restriction on the new and old active systems being in the same domain.

$TI—Assign job classes after XRF takeover

These commands are issued after an XRF takeover is complete, immediately preceding the $SI commands, in order to assign the job classes which an initiator is to process.

Format

$TI CMD=(,,MVS $TI*m-n*,*jobclass1*[,*jobclass2*...])

Keyword and Parameter Definitions

m-n
> The range of initiator numbers to which specified job classes are to be assigned.

jobclassn
> The job classes to assign to these initiators.

Comments and Usage Notes

⇒ **Not required**. Applicable to an XRF environment only.

This entry is similar to the $SI entry. The commands specified here are issued immediately preceding the $SI commands. There is no restriction on the new and old active systems being in the same domain.

Examples of Usage

```
   COMMANDS  HELP
 ----------------------------------------------------------------------------
                               CMD Processing               Row 1 to 2 of 20
 Command ===>                                               SCROLL===> PAGE

 Entry Type : Application          PolicyDB Name   : SCENARIO
 Entry Name : IMSIAOCA            Enterprise Name : TEST

 Subsystem  : IMSIAOCA
 Message ID : $TI

 Enter commands to be executed when resource issues the selected message.

 Pass/Selection Automated Function/'*'
 Command Text
 _____  _____
 MVS $TI1-12,A_____
 _____


 _____  _____
 _____
 _____
  F1=HELP      F2=SPLIT     F3=END      F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Chapter 6. Common Routines

This chapter describes IMS Automation common routines which request information or perform tasks associated with IMS Automation. You can use these common routines in automation procedures you create. Examples, sample routines, and data area information are given to show how this might be done.

IMS Automation provides new routines to retrieve and update IMS Automation-unique information. These routines can also be used in user-written extensions of IMS Automation. The following routines are arranged alphabetically for easy reference.

EVIEX002—Retrieve status file data

The purpose of this command is to retrieve data from the IMS status file extension record.

Syntax

EVIEX002 *subsystem,keyword1,...keywordN*

Keyword and Parameter Definitions

subsystem
> The name of the subsystem for which the information will be retrieved.

keywordN
> The name of the keyword of the data to return. The possible values are:
>
> ALTSYS, AVM, BUILDQCKPT, BUILDQHSB, DBRCRESRV, DCSTATUS, DEPREGID, EDNDT, LASTABENDCODE, GENAPPLID, LHSBID, LOGCKPT, RSTCHKPT, RSTHSBID, RUNAPPLID, RUNSTARTYPE, SERVSTARTDT, SERVENDDT, STARTDT, STARTHSB, STARTOPT, STARTTYPE, TOCODE, VER, XRFMODE, XRFSTATUS.
>
> For more information about the keywords, refer to common routine .

Comments and Usage Notes

When the EVIEX002 command is issued, the values will be returned in the following task global variables:

1. EVISTSV*n*

   These variables will contain the value of the keywords in the order in which they were requested. These variables are used for any number of keywords requested. If the keyword requested is invalid, the value returned will be blanks.

2. EVISTSN

   This variable contains the value of the number of keywords requested.

Examples of Usage

Example 1

Coding this:
```
EVIEX002 IMSIAOCA,XRFSTATUS
```

assigns these values:

EVISTSV1	Value of the XRFSTATUS data field
EVISTSN	1

Example 2

Coding this:
```
EVIEX002 IMSIAOCA,XRFSTATUS,ALTSYS,XRFMODE
```

assigns these values:

EVISTSV1   Value of the XRFSTATUS data field

EVISTSV2   Value of the ALTSYS data field

EVISTSV3   Value of the XRFMODE data field

EVISTSN   3

Example 3

The following example shows how to call EVIEX002 from within a REXX CLIST. The results (as they would appear at a NCCF operator session) are displayed after the sample CLIST.

```
/******************************************************************/
 TRACE OFF
 SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX002'
 'EVIEX002 IMS05Z,STARTTYPE,RSTCKPT'
 'GLOBALV GETT EVISTSV1 EVISTSV2 EVISTSN'
 SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1
 SAY 'EVISTSV2 VARIABLE IS: ' EVISTSV2
 SAY 'EVISTSN VARIABLE IS:  ' EVISTSN
      RETURN
```

Note:

```
      Sample output
        THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX002
        EVISTSV1 VARIABLE IS:  AUTO
        EVISTSV2 VARIABLE IS:  93301/081251
        EVISTSN VARIABLE IS:  2
```

EVIEX003—Update status file data

The purpose of this command is to update data on the IMS status file extension record.

Syntax

```
EVIEX003 subsystem,keyword1=value1,...keywordN=valueN
```

Keyword and Parameter Definitions

subsystem
> The name of the subsystem for which the information will be updated.

keywordN
> The name of the field to be updated.

valueN
> The value of the field to be updated.

The following keywords can be specified:

ALTSYS=
> Identifies the status of the alternate system in an XRF environment.
> **OFF**   The alternate system is inactive
> **ON**   The alternate system is active

AVM=
> Indicates if the Availability Manager is available. The Availability Manager is the controller of an XRF system that determines when a partner or backup system will takeover.
> **OFF**   Availability Manager not available
> **ON**   Availability Manager available

BUILDQCKPT=
> Specifies the last checkpoint ID that is valid for loading DC blocks.
>
> The value is extracted from message DSF3804I, which indicates the completion of an IMS checkpoint.
>
> The format is yyddd/hhmmss.

BUILDQHSB=
> When running in an XRF environment, this specifies which IMS system (HSBID) wrote the CHKPT. The value is extracted from message DSF3804I, which indicates the completion of an IMS checkpoint and the IMS system (HSBID) which wrote the CHKPT.

DBRCRESRV=
> Specifies the DBRC (IMS Database Recovery Control) reserve indicator. This will tell the old active that a reserve has been detected. The old active will then cancel its DBRC to free the reserve, allowing the new active to proceed with the takeover. The possible values are RESERVE or blank.

DCSTATUS=
> Specifies the value of the Data Communication Status field.
>
> The possible values are UP, STOPPED, or DOWN.

DEPREGID=
Exists only for dependent regions; specifies the ID of the dependent processing region: Fast Path, Message, or Batch.

ENDDT=
Specifies the date and time that the IMS subsystem terminated. Normally, this is when IMS shutdown message DFS994I has been processed.

The format is dd/mm/yy,hh:mm:ss.

The value is set to '--/--/--,--:--:--', when the termination of the IMS subsystem is initiated (due to the fact that a stop request has been issued, or when an IMS subsystem is about to abend and message DFS629I is being processed).

LASTABENDCODE=
Specifies the abend code for the last abnormal termination of the particular IMS subsystem.

GENAPPLID=
Specifies the generic VTAM applid used during the current or last execution of the particular IMS subsystem.

LHSBID=
When running in an XRF environment, this identifies the last-active IMS system.

The possible values are 1 or 2.

LOGCKPT=
When running in an XRF environment, this indicates the checkpoint at which IMS starts reading the input log.

This is triggered by message DFS3804I in the format of 'LOG READ CHKPT: yyddd/hhmmss'. This form of the message is issued only after a takeover occurred, and when the system checkpoint needed for IMS recovery was written before the LAST DC CHKPT and before the takeover.

The format of the ID is yyddd/hhmmss.

RSTCKPT=
When running in an XRF environment, this indicates the restart checkpoint ID.

The format of the ID is yyddd/hhmmss.

RSTHSBID=
When running in an XRF environment, this indicates which IMS system (HSBID) wrote the checkpoint.

RUNSTARTYPE=
Specifies the last requested start type. The field is populated after receiving message DFS629I, DFS627I, or IEF450I, indicating that an IMS has terminated.

The possible values are:

AUTO
The IMS system has been started with 'AUTO=Y'.

BUILDQ
The IMS system has been started using the BUILDQ entry from the STARTUP policy item.

A BUILDQ start implies that all messages on the queues are to be saved across restarts.

COLD
The IMS system has been COLD started.

MANUAL
The IMS system has been started using the IMS subsystem RESTART command, entered by the user.

WARMSDBL
The IMS system has been started by loading the Main Storage Date Base (MSDB) during a WARM start.

RUNAPPLID=
Specifies the specific applid. The value is taken from the IMSCNTL APPLID ACF entry when it is loaded.

SERVSTARTDT=
Specifies the time and date when IMS startup message DFS994I was processed.

When running in an XRF environment, this specifies the date and time that the alternate IMS system started. At the same time, the XRFMODE is set to 'BACKUP'.

The format is dd/mm/yy,hh:mm:ss.

The value is set to '--/--/--,--:--:--' during IMS startup when message DFS3410I is received.

SERVENDDT=
Specifies the date and time when the termination of the IMS subsystem has been initiated (due to the fact that a stop request has been issued, or when an IMS subsystem is about to abend and message DFS629I is being processed).

The format is dd/mm/yy,hh:mm:ss.

STARTDT=
Specifies the time and date when message DFS3410I has been received during IMS startup. At the same time, SERVSTARTDT is set to a value of '--/--/--,--:--:--'.

The format is dd/mm/yy,hh:mm:ss.

STARTHSB=
When running in an XRF environment, this specifies the HSBID used to start the IMS system.

The possible values are 1 or 2.

STARTOPT=
When running in an XRF environment, this indicates whether the IMS system is meant to be started as the active or the alternate IMS system.

ACTIVE
Start as the active IMS system.

XRF
This is treated like a value of ACTIVE.

BACKUP
Start as the alternate IMS system.

NOXRF
IMS system is not part of an XRF complex.

STARTTYPE=
Specifies the start type to be used to start the IMS system. The possible values are:

AUTO
The IMS system will be started with 'AUTO=Y'.

BUILDQ
The IMS system will be started using the BUILDQ entry from the STARTUP policy item.

A BUILDQ start implies that all messages on the queues are to be saved across restarts.

COLD
The IMS system will be COLD started using the 'COLD' entry from the STARTUP policy item.

MANUAL
The IMS system will be started using RESTART commands, entered by the user.

WARMSDBL
The IMS system will be started by loading the Main Storage Date Base (MSDB) during a WARM start.

TOCODE=
When running in an XRF environment, this specifies the reason code for the takeover. The reason code is extracted from messages DFS3890I or DFS3869I when the takeover processing begins.

Possible values are:

Code (Hex)	Meaning
80	VTAM termination
40	RDS surveillance
20	System LNK surveillance
10	LOG surveillance
08	/SWITCH entered
04	IRLM failure
02	ESTAE exit entered

VER=
Specifies the IMS product executing in the particular IMS subsystem, for instance:

Value	Meaning
5.1	IMS/ESA Version 5 Release 1
6.1	IMS/ESA Version 6 Release 1
7.1	IMS/ESA Version 7 Release 1

XRFMODE=
When running in an XRF environment, this identifies the IMS system as the ACTIVE or BACKUP partner in the XRF pair.

XRFSTATUS=
When running in an XRF environment, this identifies the status of the IMS subsystem. Possible values are:

ACTIVE
Indicates that this is the ACTIVE partner in the XRF pair. The status change is triggered by message DFS0488I, indicating that the UNLOCK command has completed.

ACTIVEDOWN
Indicates that the IMS subsystem which ended was the ACTIVE partner in the XRF pair.

BACKUP
Indicates that this is the ALTERNATE partner in the XRF pair. The status change is triggered by message DFS3838I or DFS3839I, indicating that the initial DB/DC PRE-OPEN has completed.

BACKUPDOWN
Indicates that the IMS subsystem which ended was the BACKUP partner in the XRF pair.

WAITBACKUP
Indicates that the alternate partner is being shutdown.

UP
Valid only for alternate IMS message regions and only when the SYNCH parameter is set to a value of 'S'.

The SYNCH parameter can be specified in the IMS-specific IMS ENVIRON policy item and controls the type and degree of region synchronization between the active and the alternate IMS subsystem.

When the automation status of the active partner's message region is UP, the XRFSTATUS of the alternate message region will be set to UP.

STOPPED
Valid only for alternate IMS message regions and only when the SYNCH parameter is set to a value of 'S'.

The SYNCH parameter can be specified in the IMS specfic IMS ENVIRON policy item and controls the type and degree of region synchronization between the active and the alternate IMS subsystem.

When the automation status of the active partner's message region is not UP, the XRFSTATUS of the alternate message region will be set to STOPPED.

DOWN
Valid only for alternate IMS message regions and only when the SYNCH parameter is set to a value of 'S'.

The SYNCH parameter can be specified in the IMS-specific IMS ENVIRON policy item and controls the type and degree of region synchronization between the active and the alternate IMS subsystem.

When a message region of the active region terminates, and message DFS552I is received, the XRFSTATUS of the alternate's message region will be set to DOWN.

NOXRFC
During startup of an IMS system, message DFS3802W, DFS3898W, or DFS3899W have been received. The messages indicate that the system being started will not be XRF capable. This may be as expected in the case of an IMS system genned with XRF but not using it at present.

When message DFS3802 is received, the IMS system is terminated if it is an alternate. Otherwise, startup will continue.

IOPREV

On the active system (which is about to become the old active system), a takeover condition has been reached. The takeover should also start soon on the alternate subsystem.

IOFREE

The IOFREE status means that I/O Prevention has completed on the old active subsystem. This is the status needed on the old active system before the I/O Toleration process can be completed on the new active (by issuing the /UNLOCK command).

TAKEOA

Indicates that IMS takeover is in progress. All class 2 sessions will be inactivated on the old active system. This is triggered by message DFS3891I, indicating that the IMS system log from the active system has been processed by the alternate system.

TAKEOB

The alternate system has initiated a takeover and is about to become the new active.

IOT

This is the status of the new active subsystem after a takeover, during the I/O Toleration process, and prior to becoming a true active system.

Comments and Usage Notes

When the EVIEX003 command is issued, the values for the named subsystem will be changed in the status file.

No validation is performed on the values you specify. If you specify incorrect values, unpredictable results can occur.

Examples of Usage

Example 1

If the following statement were coded:

```
EVIEX003 IMSIAOCA,XRFMODE=ACTIVE
```

the IMS status extension record for the subsystem IMSIAOCA would be updated to reflect the value of ACTIVE for the data field XRFMODE.

Example 2

The following example shows how to call EVIEX003 from within a REXX
CLIST. The results can be displayed by calling the common routine EVIEX002.
The command results (as they appear in an operator NCCF session) are
shown after the program example.

```
/********************************************************************/
TRACE OFF
SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE
EVIEX003 & EVIEX002'

'EVIEX003 IMS05Z,XRFMODE=ACTIVE'        /* CHANGES VALUE OF
                                           XRFMODE to 'ACTIVE' */

'EVIEX002 IMS05Z,XRFMODE'               /* RETRIEVE NEW VALUE
                                           OF XRFMODE          */

'GLOBALV GETT EVISTSV1 EVISTSN'

SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1  /* DISPLAY VALUE        */
SAY 'EVISTSN VARIABLE IS:  ' EVISTSN

'EVIEX003 IMS05Z,XRFMODE=BACKUP         /* RESET VALUE OF
                                           XRFMODE             */

'EVIEX002 IMS05Z,XRFMODE'               /* RESET VALUE OF
                                           XRFMODE             */
'GLOBALV GETT EVISTSV1 EVISTSN'
SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1  /* DISPLAY VALU E       */
RETURN
```

Note:

```
      Sample output
         C AOF05    EVISTSV1 VARIABLE IS:  ACTIVE
         C AOF05    EVISTSN VARIABLE  IS:  1
         C AOF05    EVISTSV1 VARIABLE IS:  BACKUP
```

EVIEX004—Retrieve XRF partner data

The purpose of this command is to retrieve data from the IMS status file extension record of the partner subsystem in an XRF environment.

Syntax

EVIEX004 *partner_subsystem,partnerdom,keyword1...,keywordN*

Keyword and Parameter Definitions

partner_subsystem
> The name of the partner subsystem for which the information will be retrieved.

partnerdom
> The domain ID of the partner's NetView that is controlling the partner subsystem.

keywordN
> The name of the field to be returned. The possible values are:
>
> ALTSYS, AVM, BUILDQCKPT, BUILDQHSB, DBRCRESRV, DCSTATUS, DEPREGID, EDNDT, LASTABENDCODE, GENAPPLID, LHSBID, LOGCKPT, RSTCHKPT, RSTHSBID, RUNAPPLID, RUNSTARTYPE, SERVSTARTDT, SERVENDDT, STARTDT, STARTHSB, STARTOPT, STARTTYPE, TOCODE, VER, XRFMODE, XRFSTATUS.
>
> For more information about the keywords, refer to common routine .

Comments and Usage Notes

When the EVIEX004 command is issued, the values for the partner subsystem will be returned in the following task global variables:

1. EVISTSVn

   These variables will contain the value of the keywords in the order in which they were requested. These variables are used for any number of keywords requested. If the keyword requested is invalid, the value returned will be blanks.

2. EVISTSN

   This variable contains the value of the number of keywords requested.

Examples of Usage

Example 1

If you code this:
```
EVIEX004 IMSIAOCA,CNM02,XRFSTATUS
```

the request will retrieve the data from the status file in domain CNM02 and the following values will be assigned:

EVISTSV1 Value of the XRFSTATUS data field for the partner subsystem in domain CNM02

EVISTSN 1

Example 2

Coding this:
```
EVIEX004 IMSIAOCA,CNM02,XRFSTATUS,ALTSYS,XRFMODE
```

Assigns these values:

EVISTSV1   Value of the XRFSTATUS data field for the partner subsystem in domain CNM02

EVISTSV2   Value of the ALTSYS data field for the partner subsystem in domain CNM02

EVISTSV3   Value of the XRFMODE data field for the partner subsystem in domain CNM02

EVISTSN   3

Example 3

The following example shows how to call EVIEX004 from within a REXX CLIST. The results (as they appear on an operator NCCF session) follow the program sample.

This common routine can only be used for an XRF environment running in a dual CPC configuration.

```
/********************************************************************/
 TRACE OFF
/* SAMPLE REXX EXEC                                               */
 SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX004'

 'EVIEX004 IMS01E,AOF01,AVM,DCSTATUS'

 'GLOBALV GETT EVISTSV1 EVISTSV2 EVISTSN'

 SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1
 SAY 'EVISTSV2 VARIABLE IS: ' EVISTSV2

 SAY 'EVISTSN VARIABLE IS: ' EVISTSN
 RETURN
```

Note:

```
     Sample output
     C AOF10    THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX004
     C AOF10    EVISTSV1 VARIABLE IS:  OFF
     C AOF10    EVISTSV2 VARIABLE IS:  DOWN
     C AOF10    EVISTSN VARIABLE IS:  2
```

EVIEX005—Retrieve XRF partner data using a local subsystem name

The purpose of this command is to retrieve data from the IMS status file extension record of the partner subsystem in an XRF environment without knowing the partner's subsystem name or domain name.

Syntax

```
EVIEX005  local_subsystem,keyword1...,keywordN
```

Keyword and Parameter Definitions

local_subsystem
> The name of the subsystem for whose partner selected information will be retrieved.

keywordN
> The name of the field to be returned.

Comments and Usage Notes

When the EVIEX005 command is issued, the values for the partner subsystem will be returned in the following task global variables:

1. EVISTSVn

   These variables will contain the value of the keywords in the order in which they were requested. These variables are used for any number of keywords requested. If the keyword requested is invalid, the value returned will be blanks.

2. EVISTSN

   This variable contains the value of the number of keywords requested.

   Warning: The task global variable from a previous call to EVIEX005 is not erased. This means that care should be taken if the result variable EVISTSVn is used without checking the value of EVISTSN. If the call is unsuccessful (rc ¬=0) the result variables will contain values from the last successful invocation of EVIEX005.

Examples of Usage

Example 1

If you code this:
```
EVIEX005 IMSIAOCA,XRFSTATUS
```

the request will retrieve the data from the status file in domain CNM02 and the following values will be assigned:

EVISTSV1   Value of the XRFSTATUS data field for the partner subsystem (IMSIAOCB) in domain CNM02.

EVISTSN   1

Example 2

Coding this:

```
EVIEX005 IMSIAOCA,XRFSTATUS,ALTSYS,XRFMODE
```

assigns these values:

EVISTSV1 Value of the XRFSTATUS data field for the partner subsystem (IMSIAOCB) in domain CNM02

EVISTSV2 Value of the ALTSYS data field for the partner subsystem (IMSIAOCB) in domain CNM02

EVISTSV3 Value of the XRFMODE data field for the partner subsystem (IMSIAOCB) in domain CNM02

EVISTSN 3

Example 3

The following example shows how to call EVIEX004 from within a REXX CLIST. The results (as they appear on an operator NCCF session) follow the program sample.

This common routine can only be used for an XRF environment running in a dual CPC configuration.

```
/********************************************************************/
 TRACE OFF
/* SAMPLE REXX EXEC                                                */
 SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX004'

 'EVIEX004 IMS01E,AOF01,AVM,DCSTATUS'

 'GLOBALV GETT EVISTSV1 EVISTSV2 EVISTSN'

 SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1
 SAY 'EVISTSV2 VARIABLE IS: ' EVISTSV2

 SAY 'EVISTSN VARIABLE IS: ' EVISTSN
 RETURN
```

Note:

```
      Sample output
      C AOF10    THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX004
      C AOF10    EVISTSV1 VARIABLE IS:  OFF
      C AOF10    EVISTSV2 VARIABLE IS:  DOWN
      C AOF10    EVISTSN VARIABLE IS:  2
```

IMSBMSG—Build message processor

This processor allows programs written in REXX, NetView command lists, and high level languages to build a single or multi-line message and have it delivered to a designated task on any connected NetView domain.

Syntax

```
IMSBMSG START,domainid,taskid
IMSBMSG DATA,linetype,data
IMSBMSG END
IMSBMSG CANCEL
```

The parameters are positional.

START

This parameter is used for the first call to IMSBMSG. Parameters on the START indicate the target of the message or messages, where *domainid* is the identifier of the domain to which the message is to be routed and *taskid* is the identifier of the task on that domain. The *taskid* can be a real operator name or the name of an autotask on the receiving system.

DATA

This identifies a user message being passed to the IMSBMSG processor where *linetype* and *data* is the message that is to be passed to the target task to which the message is to be routed. The line type of the element of a multi-line message, valid values, and their meanings are as follows.

C	Control line
L	Label line
D	Data line
E	End line
F	End-with-data line
S	Single line.

If a multi-line message is to be issued, a valid combination of line types must be passed.

END

Indicates to the IMSBMSG processor that processing is complete. It causes the processor to route previously issued data lines to the target indicated on the IMSBMSG START command. All storage obtained during processing is released. An IMSBMSG START must be issued before any other DATA items.

CANCEL

This indicates to the IMSBMSG processor that processing is not to continue. All storage is released without any message being issued. Use CANCEL to abort the message.

Comments and Usage Notes

1. All calls must be made from the same invocation of the same program.
2. A correct sequence of line types should be passed. However, if the processor detects that a multi-line message is being passed and no E or F type has been received by the time the IMSBMSG END is received, then message **EVI698 END generated** is added to the group.
3. Single-line and multi-line messages can be mixed in the same START-END bracket.

4. The return codes are:

0	OK.
4	Invalid type. Type should be START, DATA, END or CANCEL.
8	Invalid line type. Type should be C, L, D, E, F, or S.
12	Execution failed. A message is issued that describes the failure.
16	DATA, END or CANCEL issued without START.
20	START issued while previous start in effect.
24	Wrong number of parameters.
28	Invalid domainid. Ensure the domainid specified is correct.
32	The domain identified by domainid is not active.
36	The taskid specified is invalid or not active.
40	The value of the CGLOBAL EVI_BASE_PRODUCT is NULL or invalid.

Examples of Usage

Example

The following example shows how to call IBMBMSG from within a REXX CLIST. The results (as shown on an operator NCCF session) follow the code sample.

```
/*****************************************************************/
 TRACE OFF
 TRACE OFF
SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE IMSBMSG'

'IMSBMSG START,AOF05,ROMAN'
'IMSBMSG DATA,S,This is a test................. '
'IMSBMSG END'

SAY 'RC FROM IMSBMSG IS : ' RC

 RETURN
```

Note:

```
Sample output
* AOF10    CLISTX
C AOF10    THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE IMSBMSG
U AOF10    THIS IS A TEST.................
C AOF10    RC FROM IMSBMSG IS :  0
```

Note: Data is echoed back to caller.

IMSCMD—Issue IMS commands

This common routine is used to perform the requested function on the domain where the named IMS resides, whether local or remote. The calling program does not have to be aware of where the IMS resides. It is particularly useful with single-point-of-control as IMSCMD first determines the domain in which the subsystem resides before building and issuing the request. It then either calls the requested function if the subsystem is on the local domain, or it forwards the command to the remote domain, thus allowing cross-domain communications.

Syntax

IMSCMD *subsystem IMS_command*

Keyword and Parameter Definitions

subsystem
> The IMS subsystem that can be controlled by the domain. It can be either a local or a remote IMS subsystem.

IMS_command
> The actual IMS command to be issued.

Comments and Usage Notes

1. Results from a command are returned in a multi-line message.
2. The IMS command must be the full IMS command or a subset of the characters. For example, /CHECKPOINT and /CHE are valid. Command synonyms are not supported, such as /CHKPT for /CHECKPOINT. Use of command synonyms will result in message EVI501 with return code 110.
3. The IMS commands are restricted to those commands allowed for an IMS automated operator program. This list of commands is documented in the *IMS Customization Guide* under the heading "Automated Operator Commands". The list is also documented in the *IMS Operator's Guide* under the "Transaction-Entered Commands" topic. If a restricted command is issued, you will receive message EVI501I with return code 110 (if rejected by the IMSCMD routine) or message EVI152 (if rejected by IMS).
4. If IMS is customized to restrict the commands the IMS Automation BMP program can issue, message EVI152 will be received for any restricted commands.
5. The IMSCMD routine performs input validation and security checking before passing the command to IMS. If an error is detected, message EVI501 will be issued with a return code and the return code will be passed back to any calling routine. The possible return codes are:

   0   Passed edits. Waiting for PPI processing or IMS command processing messages.

   1-100   Return code from IMSQRY. Refer to return codes for IMSQRY.

   103   Input error. The IMS name is missing, the command is missing, or both the IMS name and the command are missing.

   104   The name entered is not an IMS subsystem.

   110   The specific IMS command is not supported thru the IMS automated operator function.

Examples of Usage

Example 1

If you issue:

```
IMSCMD IMS10AA /DIS A
```

IMS Automation will respond:

```
 U AOF01    EVI120I COMMAND ACCEPTED FOR IMS10AA , APPLID = IMS10AA
 = AOF01
EVI690I PPI RESPONSE FROM IMS10AA  FOR FUNCTION IMSCMD   083 00655
EVI691I  ** SUBSYSTEM NAME **
EVI691I      IAOC
EVI691I  REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS      CLASS
EVI691I      3 MSGIMSAB  TP                       WAITING      5, 6, 7,
EVI691I      2 MSGIMSAA  TP                       WAITING      1, 2, 3,
EVI691I       1 PPIIMSA   BMP   PPIIMSA   EVISPPII
EVI691I        FPRGN     FP    NONE
EVI691I  REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS
EVI691I        DBRIMSAA  DBRC
EVI691I        DLIIMSAA  DLS
EVI691I  VTAM ACB OPEN        -LOGONS ENABLED
EVI691I  LINE ACTIVE-IN -   1 ACTIV-OUT -   0
EVI691I  NODE ACTIVE-IN -   0 ACTIV-OUT -   4
EVI691I  LINK ACTIVE-IN -   0 ACTIV-OUT -   0
EVI691I  *91154/111742*  IAOC     ACTIVE
EVI692I END
```

Example 2

If you issue:

```
IMSCMD IMS10AA /DIC A (error)
```

IMS Automation will respond:

```
 U AOF01    EVI120I COMMAND ACCEPTED FOR IMS10AA , APPLID = IMS10AA
 = AOF01
EVI690I PPI RESPONSE FROM IMS10AA  FOR FUNCTION IMSCMD   083 00045
EVI691I DFS107 11:22:36 REQUIRED KEYWORD NOT PRESENT
EVI692I END
```

IMSFWM—Add messages to SDF

This routine adds messages to SDF critical message panels.

Syntax

```
IMSFWM msgtext
IMSFWM TYPE=t,msgtext
```

Keyword and Parameter Definitions

msgtext
> The message text and message identifier passed to the Status Display Facility critical message panel.

TYPE=*t*
> A 1-character value corresponding to an SDF CRITMSG type entry in the STATUS DETAILS policy object. A, E, I, and W are supplied already. Other values may be specified, provided a SDF CRITMSG*t* in the control file corresponds to that type value.

Comments and Usage Notes

This must be called from the automation table because it uses the jobname associated with the message.

If the TYPE=*t* parameter is not specified, *t* is set to the last character of the message ID, and the search is made. If no CRITMSG*t* entry is found, the CRITMSG value will be used.

Examples of Usage

Example 1

```
IF  MSGID='DFS0414I' & TOKEN(2) = 'PERMANENT'
    & TEXT=MESSAGE
    THEN EXEC(CMD('IMSFWM ' MESSAGE) ROUTE(ONE *)) ;
```

Example 2

```
IF  MSGID='DFS3257I' & TEXT=MESSAGE
   THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ALL *));
```

> **Example 3**
>
> If you wish to see certain IRLM messages in blue reverse video, add an entry like this in the STATUS DETAILS policy object:
>
> ```
> SDF CRITMSGU,CO=B, PR=500,HL=R
> ```
>
> and call IMSFWM from the message table as follows:
>
> ```
> IF  MSGID='DXR002I' | MSGID = 'DXR007E' |
>     MSGID='DXR011I' | MSGID = 'DXR018E' |
>     MSGID='DXR030I' | MSGID = 'DXR034I' |
>     MSGID='DXR035I' | MSGID = 'DXR045W' |
>     MSGID='DFS626I')
>     & TEXT=MESSAGE
>   THEN EXEC(CMD('IMSFWM TYPE=U'MESSAGE) ROUTE(ALL *));
>           DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
> ```
>
> **Note:** A comma is required as a delimiter after the type=U portion of the command.

IMSPOST—Post an external event

Use this routine to set trigger conditions. Internally, the SA OS/390 commmand INGEVENT is called. Therefore, the TYPE parameter is ignored. For information on INGEVENT, see *System Automation for OS/390 Operator's Commands*.

Syntax

```
IMSPOST NAME=subsys,FUNCTION={SET|UNSET},EVENT=event
        [,TYPE={STARTUP|SHUTDOWN}]
```

Keyword and Parameter Definitions

NAME=
  Is used to define the symbolic name by which this IMS subsystem is known to SA OS/390.

FUNCTION=
  Specified whether the trigger is to be SET or UNSET.

EVENT=
  The name of the external condition that this trigger represents.

TYPE=
  Specifies whether this is a STARTUP or SHUTDOWN trigger. This keyword is ignored by INGEVENT.

Comments and Usage Notes

As IMSPOST calls INGEVENT internally, the TYPE parameter is ignored.

IMSQRY—Name lookup

Use this routine to retrieve IMS subsystem information.

Note that IMSQRY does not recognize subsystems that are in FALLBACK or MOVED status.

```
┌─ Syntax ─────────────────────────────────────────────────┐
│                                                           │
│  IMSQRY REQ={VALIDATE|GET}                                │
│          [,NAME={subsystem|resource_name|jobname}]        │
│          [,TYPE={IMS|GROUP|DOMAIN|ANY|JOBNAME}]           │
│                                                           │
└───────────────────────────────────────────────────────────┘
```

Keyword and Parameter Definitions

REQ=
The request type. The request types are:

VALIDATE
IMS Automation searches for the name (NAME=) and type (TYPE=) specified to validate the name.

GET
IMS Automation searches for a specific IMS subsystem to retrieve the subsystem characteristics. NAME= is valid with this request.

NAME={*subsystem* | *resource_name* | *jobname*}
Used with VALIDATE to provide a specific subsystem, resource or job name for the search. Used with GET to provide a specific subsystem value. Valid values for the NAME= variables are:

subsystem	The name by which an IMS subsystem is known to SA OS/390
resource_name	The resource name in the *name*/APL/*system* format. Thus, the type must be APL; APG, for example, is not valid.
jobname	The jobname by which an IMS subsystem is known to SA OS/390.

TYPE=
Used to provide a specific type. The types are:

IMS
Search for a specific IMS subsystem name, as it is known to SA OS/390. This is the default type. IMS is used with VALIDATE.

ANY
Search for an IMS name first, then a domain, then a group name. If the name is longer than 5 characters the search for a domain is bypassed. ANY is used with VALIDATE.

DOMAIN
The NetView domain name coded in DSIDMNK with the NCCFID parameter.

GROUP
If you specify GROUP, IMSQRY returns the name of the group to which the subsystem belongs in the EVILOOKUP_GROUP variable.

JOBNAME
Used with GET to provide a specific jobname. Works only when NAME=jobname.

Comments and Usage Notes

1. Valid IMSQRY return codes include:

RC	Meaning
0	Good
4	An internal error occurred
8	A timeout occurred on a request forwarded to a remote system
12	An internal error occurred
20	A subsystem, group, or domain was not found for the search criteria specified
24	The parameters for this request are invalid
28	An internal error occurred

2. The following are set in the caller's variable pool:

EVILOOKUP_NAME
> Set to the value of the NAME= parameter, otherwise set to null.

EVILOOKUP_TYPE
> Set to the value of the TYPE= parameter, unless TYPE=ANY in which case it is set to IMS or DOMAIN or GROUP as appropriate.

EVILOOKUP_JOBNAME
> The jobname associated with the subsystem.

EVILOOKUP_DOMAIN
> The NetView domain on which the SA OS/390 managing this subsystem is running.

EVILOOKUP_AUTOOPS
> The NetView automated operator that handles automation for this subsystem.

EVILOOKUP_USERVAR
> The VTAM USERVAR (or generic application ID) associated with this subsystem. This is set to '********' if a VTAM USERVAR is not defined.

EVILOOKUP_SUBTYPE
> The value of the **Subtype** field in the IMS ENVIRON policy item .

EVILOOKUP_APPLID
> The specific VTAM application ID associated with this subsystem.

EVILOOKUP_RESHOME
> The location of the resource in the following format:
>
> `sysplex.domain.system\VxRyMz`

EVILOOKUP_RESLIST
> The resource name in the following format
>
> `name/type/system`

EVILOOKUP_AGENTDATA
> Information about the agent responsible for the subsystem in the following format
>
> `agent_name sysplex_name system domain agent_version [netview_version]`

EVILOOKUP_GROUP
> The name of the group(s) to which the resource belongs.

IMSRCMD—Request an IMS function

This common routine is used to perform the requested function (CMD=) on the domain where the named IMS resides, whether local or remote. The calling program does not have to be aware of where the IMS resides. It is particularly useful with single-point-of-control as IMSRCMD first determines the domain in which the subsystem resides before building and issuing the request. It then either calls the requested function if the subsystem is on the local domain, or it forwards the command to the remote domain, thus allowing cross-domain communications.

Syntax

```
IMSRCMD NAME=subsys,[RESP=YES|NO|ACK,]
        [OPER=operator,]CMD=cmd
```

Keyword and Parameter Definitions

NAME=
   The name by which the target IMS subsystem is known to SA OS/390, as defined in the APPLICATION policy object.

RESP=
   Send back a response (YES or NO) or just send an acknowledgement (ACK).

OPER=
   The operator, on the target domain, that will execute this command. If this is omitted, BASEOPER will be used.

CMD=
   The requested function to be performed. This may be delimited by single quotes, double quotes, or slashes.

Comments and Usage Notes

The return codes are:

RC	Meaning
0	Good
4	Subsystem name was not supplied
8	Function to be performed was not supplied
12	Incorrect keyword supplied
16	Incorrect parameter for RESP supplied
20	Subsystem was not found on any domain

XRFCMD—Set XRF Events

Use this routine to set the trigger conditions that control

- which subsystem will be started as the active subsystem, and
- whether the alternate subsystem should be started.

The command assumes that the two XRF subsystems are linked to a trigger that uses the IsActive, IsBackup, and StartBck events as defined in the sample database.

Syntax

```
XRFCMD XRFGROUP=appl_group,ACTION={SWITCH|BACKUP|NOBACKUP}
```

Keyword and Parameter Definitions

XRFGROUP=
The name of the application group that contains both XRF subsystems as members; see "Defining an XRF Complex" on page 41.

ACTION=
SWITCH
Causes the IsActive and IsBackup events to be switched for the two XRF subsystems. The XRF subsystems must not be running when the command is issued.
BACKUP
Sets the StartBck event which allows the backup system to be started.
NOBACKUP
Unsets the StartBck event preventing the backup subsystem from being started.

Comments and Usage Notes

The SWITCH action will cause automation to switch the roles of the XRF subsystems. When automation next starts the XRF group the old backup will be started as the active. You must use the BUILDQ start type after such a switch.

INGIMS - Issue List of Defined Transactions and View the Output

The INGIMS command lets you:

- Issue any console-enabled IMS command
- Broadcast messages to all or selected IMS users
- Issue a list of defined transactions and view the output
- Display the output of IMS transactions in full-screen or pipeable line mode

For a detailed description of the INGIMS command, refer to *System Automation for OS/390 Operator's Commands*.

Part 3. Using IMS Automation

This part describes the tasks of the operator who manages IMS subsystems through IMS Automation.

Chapter 7. Using Panels and Working with Subsystems

This chapter explains how to use the IMS Automation panels and to work with subsystems. We assume that you have used and are familiar with the SA OS/390 operator interface. This chapter is devoted to describing those characteristics unique to IMS Automation. We try to provide plentiful details to guide you. But to thoroughly understand your role as the IMS Automation operator, you will need some hands-on experience with SA OS/390.

Using IMS Automation Panels

This section explains:
- Panel characteristics
- How to select tasks from panels
- How to use panel fields and function (F) keys
- Options available from the Main Menu.

To start an IMS Automation operator session and display the IMS Automation Main Menu, enter **IMS** on a NetView command line.

Panel Characteristics

All IMS Automation panels have similar characteristics, as illustrated in Figure 32.

```
EVIK0000                   SA OS/390 - Command Dialogs
Domain ID   = IPSFM    ---------- IMS      ----------    Date = 06/18/02
Operator ID = OPER                                       Time = 10:04:13

Resource    1       =>                          Format: name/type/system
System      2       =>              System name, domain ID or sysplex name

     1. Inquire          Display detailed status of an IMS subsystem
     2. Start            Start an IMS subsystem      INGREQ REQ=START
     3. Shutdown         Shutdown an IMS subsystem   INGREQ REQ=STOP
     4. Triggers         Display trigger conditions    DISPTRG
     5. Service Periods   Perform scheduling functions INGSCHED
     6. Master Terminal  Perform Master Terminal Cmds INGIMS REQ=CMD
     7. Critical messages Display critical messages
     8. Broadcast        Send message to users       INGIMS REQ=BROADCAST
     9. TCO Management    Load/Start/Stop TCO
    99. Local Functions  Provide access to user defined local functions




ING006I THERE IS NO RESOURCE IMS711X1/APL/KEY1     3
Command ===>     4
  PF1=Help      PF2=End     PF3=Return                     PF6=Roll     5
```

Figure 32. Common Characteristics of IMS Automation Panels

The list below identifies common characteristics of IMS Automation panels:

1　　In this field, you must specify the resource you will work with. Most of the IMS Automation panels have this field, allowing you to change resources without having to return to the main menu.

　　　　By entering a question mark ('?') at position one of the **Resource** field and pressing ENTER, you can call a list of all IMS control regions from the current sysplex that are defined to SA OS/390. You can then select a

subsystem from this list. To get this list for another sysplex other than the current one, specify the respective sysplex in the **System** field, before you press ENTER.

Note however that many of the menu options are not valid for remote sysplexes.

2 In this field, you can specify the system, domain, or sysplex to which the command that you call by selecting an option is to be routed. If you specify a sysplex name and then enter a question mark in the **Resource** field you call a list of all the IMS control regions in the specified sysplex. For the current sysplex, you need not enter the sysplex name.

3 IMS Automation messages display here.

4 From the command line, you can enter IMS Automation operator commands, as well as any other command that can be entered from a NetView panel. You also select from the current menu by entering the option number in the command line.

5 You use the function keys primarily to navigate through the panels. Each function key has a self-explanatory label. However, if you see an unfamiliar function key, refer to the panel help information by pressing PF1.

Using the Main Menu

The main menu panel lists all of the tasks available with the operator interface.

```
EVIK0000                   SA OS/390 - Command Dialogs
Domain ID   = IPSFM      ---------- IMS      ----------     Date = 06/18/02
Operator ID = OPER                                          Time = 10:04:13


Resource       =>                                    Format: name/type/system
System         =>                         System name, domain ID or sysplex name


      1. Inquire           Display detailed status of an IMS subsystem
      2. Start             Start an IMS subsystem         INGREQ REQ=START
      3. Shutdown          Shutdown an IMS subsystem      INGREQ REQ=STOP
      4. Triggers          Display trigger conditions     DISPTRG
      5. Service Periods   Perform scheduling functions INGSCHED
      6. Master Terminal   Perform Master Terminal Cmds INGIMS REQ=CMD
      7. Critical messages Display critical messages
      8. Broadcast         Send message to users          INGIMS REQ=BROADCAST
      9. TCO Management    Load/Start/Stop TCO
     99. Local Functions   Provide access to user defined local functions




ING006I THERE IS NO RESOURCE IMS711X1/APL/KEY1
Command ===>
  PF1=Help     PF2=End     PF3=Return                      PF6=Roll
```

Figure 33. IMS Automation Main Menu

The following list describes the options you can select from the main menu:

Inquire
Use this option to retrieve detailed information for an IMS resource. See "Getting Detailed Status" on page 157.

Start Use this option to initiate the startup process of a resource. By choosing this option you call the INGREQ command of SA OS/390. See "Start" on page 163.

Shutdown

Use this option to initiate the shutdown process of a resource. By choosing this option you call the INGREQ command of SA OS/390. See Chapter 8, "Starting and Stopping Resources" on page 163.

Triggers

Use this option if you want to display the triggers associated with a resource. By choosing this option you call the DISPTRG command of SA OS/390. See *System Automation for OS/390 Operator's Commands*.

Service Periods

Use this option if you want to display or override the schedule associated with a resource. By choosing this option you call the INGSCHED command of SA OS/390. See *System Automation for OS/390 Operator's Commands*.

Master Terminal

Use this option to issue a command to a specific subsystem. By choosing this option you call the INGIMS REQ=CMD command. See *System Automation for OS/390 Operator's Commands* for details.

Critical Messages

This facility displays critical messages in a scrollable format. Once displayed, messages can be deleted. See Chapter 10, "Displaying Critical Messages" on page 175.

Broadcast

Use this option to send a message to a specific subsystem. By choosing this option you call the INGIMS REQ=BROADCAST command of SA OS/390. See *System Automation for OS/390 Operator's Commands* for details.

TCO Management

Use this option to manage the TCO functions of a specific subsystem. See Chapter 9, "TCO Functions" on page 167 for details.

Local Functions

IMS Automation allows your system programmer to add functions to this operator interface. If functions have been added at your installation, you would select this option to view a menu of them.

> **Important**
>
> The options 1 (Inquire) and 6 through 99 are only valid for the local sysplex. You cannot access a remote sysplex with any of these functions.

Using Fast Path

IMS Automation offers a *fast path* option that lets you go directly to any panel in its interface from NetView. Panels have identifying numbers based on their option number from the main menu. You *fast path* by entering a panel's identifying number on the command line. Several of the second-level panels have numbered menu options on them which take you to third-level panels. To get to the second-level panels, add a period and the number of the selection. The format to access third-level panels is *num.num.num*.

A fast path to the main menu is defined to a function key. You can return to the main menu panel from other panels by pressing PF4.

To fast path to an IMS Automation panel from NetView, enter:

```
    IMS nn
or
    IMS subsys
or
    IMS nn,subsys
```

on the command line, where *nn* is the number corresponding to the option panel
and *subsys* is the name of the IMS subsystem you want to work with.

Example
Entering `IMS 9.1,IMS10AA`

will take you from NetView into IMS Automation to the **Program-to-Program
Interface** panel, with the subsystem IMS10AA specified.

Selecting and Viewing Subsystems

This section explains how to select a resource from a list of available resources and
how to display detailed information about a subsystem.

Selecting a Subsystem

You can specify the resource you want to work with by simply entering its name
in the **Subsystem** or **Resource** field of the respective panel. On the IMS-specific
panels you can call up a list of the available IMS control regions by entering a
question mark at position one of the **Subsystem** or **Resource** field. On the panels
that belong to basic SA OS/390 commands (INGREQ, INGSCHED, DISPTRG), you
can use an asterisk (*) as a wildcard.

Figure 34 shows a list of IMS control regions generated with the question mark
function.

```
 INGKYSTS              SA OS/390 - Command Dialogs    Line  1    of 34
 Domain ID   = IPSFM      ----- Selection Panel ------   Date = 07/10/00
 Operator ID = SCHR          Sysplex = KEY1PLEX          Time = 06:27:50

 CMD: S Select                                              / scroll
 CMD Name        Type System   Compound       Desired      Observed     Nature
 --- ----------- ---- -------- ------------ ------------ ---------- --------
 s   IMS641C4    APL  KEY2     INHIBITED    AVAILABLE    UNKNOWN
     IMS711C4    APL  KEY1     INAUTO       UNAVAILABLE  STARTING
     IMS711DL    APL  KEY1     INAUTO       AVAILABLE    STARTING
     IMS711F1    APL  KEY1     SATISFACTORY UNAVAILABLE  SOFTDOWN
     IMS711M1    APL  KEY1     SATISFACTORY UNAVAILABLE  SOFTDOWN
     IMS711RC    APL  KEY1     INAUTO       AVAILABLE    STARTING
     IMS712CX    APL  KEY1     SATISFACTORY AVAILABLE    AVAILABLE
     IMS712DL    APL  KEY1     SATISFACTORY AVAILABLE    AVAILABLE
     IMS712F1    APL  KEY1     PROBLEM      UNAVAILABLE  HARDDOWN
     IMS712M1    APL  KEY1     PROBLEM      UNAVAILABLE  HARDDOWN
     IMS712RC    APL  KEY1     SATISFACTORY AVAILABLE    AVAILABLE
     IMS713C4    APL  KEY1     AWAITING     UNAVAILABLE  AVAILABLE
     IMS713DL    APL  KEY1     INAUTO       UNAVAILABLE  STOPPING


 Command ===>
  PF1=Help    PF2=End     PF3=Return                        PF6=Roll
              PF8=Forward  PF9=Refresh PF10=Previous PF11=Next   PF12=Retrieve
```

Figure 34. Selection Panel for IMS Resources

The list contains all IMS subsystems of the KEY1PLEX sysplex (see line 3 of the
panel) that are defined to SA OS/390. You can use it not only to select a
subsystem (by entering s in the **CMD** column), but also to get an overview of the

sysplex. Columns 3 through 5, for example, contain status information for the subsystems; for more details on the different status types, see *System Automation for OS/390 User's Guide*. You can scroll horizontally through the list to the right by pressing PF11, and to the left by pressing PF10. The first three colums, which make up the resource name of the subsystem, are fixed.

Getting Detailed Status

To view in detail the status of a specified subsystem, select option **1** on the main menu panel to display the following **Inquire Subsystem Components** panel:

```
 EVIKQ100        IMS Automation: Inquire Subsystem Components  Page:   1 of   1
                                                               Date: 04/19/02
 Resource/Domain  => IMS713C4/APL/KEY1          (? for list) Time: 15:31
                                                            Domain:


 Select an option . . . . .     1  Detailed Subsystem status
                                2  Defined dependent Regions
                                3  Active dependent Regions
                                4  Shutdown status
                                5  Explanation of Takeover reason code





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

Figure 35. Inquire Subsystem Components Panel

From this screen, you can select the type of status display desired. These options include:

Resource/Domain   Displays detailed status of the specified resource and domain. See Figure 36 on page 158 for an IMS control region example, or Figure 37 on page 158 for a DB control region example.

Defined Regions   Displays the dependent regions as defined in the policy database for the specified subsystem and the status of each.

Active Regions   Invokes the Subsystem/Active Regions Display panel, which lists the subsystem status, any active regions, jobnames, and tasknames.

        This screen contains an extra refresh key, PF9. Pressing PF9 sets the screen to update displayed information approximately every ten to twelve seconds. Pressing PF9 a second time stops the refresh cycle.

Shutdown   Invokes the **Shutdown Status Display** panel, which lists the active elements, either nodes (terminals), lines, or links during shutdown processing.

Takeover reason code   Displays the reason code and explanation for a takeover. If no takeover has occurred, the Takeover Cause field will display, `No Takeover Reason Code available`. See Figure 42 on page 161.

You can display the detailed status for another resource or domain by entering the resource or domain name, selecting option 1 and pressing ENTER.

```
 EVIKQI00        IMS Automation: Subsystem Information
                                                           Date: 04/08/02
 Resource/Domain  => IMS721C4/APL/KEY2              (? for list) Time: 15:12
                                                            Domain:

 Subsystem status . . : UP              Since . . . . . : 11:11    04/08/02
 Job . . . . . . . . : IMS721C4         Job number  . . :
 NetView domain . . . : IPSFN           CQS name  . . . :

 VTAM information
   Specific appl. name: IPSANI71        DC status . . . : DOWN
   Generic appl. name : IPSANI71        XRF . . . . . . :
   Active sessions  . :                 XRF mode  . . . :
   Pending sessions . :                 XRF status  . . :

 Last start                             Last shutdown
   Initiated : 11:08:12    04/08/02       Initiated . . : 09:21:44    09/05/01
   Completed : --:--:--    --/--/--       Completed . . : 07:44:33    03/15/02
   Start type: AUTO                       Abend code  . :
 Next start  : NONE      NONE          Next shutdown  : NONE      NONE

 Command ===>
 F1=Help     F2=End       F3=Return    F4=IMS Menu  F5=Refresh   F6=Roll
```

Figure 36. Detailed Subsystem Information Panel for an IMS Control Region. If the startup or shutdown occurred while NetView was up and running, the "Last start" and "Last shutdown" data will be accurate. If NetView was not running when the startup/shutdown occurred, the data may be inaccurate.

```
 EVIKQID0            IMS Automation: Subsystem Information    Page:  1 of 1

                                                           Date: 09/22/02
 Resource/Domain  => IMS711C4/APL/KEY1             (? for list) Time: 17:04:00
                                                            Domain: IPSNM
 Subsystem status . . : UP              Since . . . . . : 16:42    10/10/93
 Job . . . . . . . . : IMS401C          Job number  . . : 9843
 NetView domain . . . : AOF01           IMSID . . . . . : I41C
 Command Character  . : ?




 Last start                             Last shutdown
   Initiated : 16:40:28    09/10/02       Initiated . . : 17:06:00    09/05/02
   Completed : 16:42:35    09/10/02       Completed . . : 17:07:56    09/05/02
   Start type: AUTO                       Abend code    :

 Next start  :          none           Next shutdown  :          none

 Command ===>
 F1=Help     F2=End       F3=Return    F4=IMS Menu  F5=Refresh   F6=Roll
```

Figure 37. Detailed Subsystem Information Panel for a DB Control Region. If the startup or shutdown occurred while NetView was up and running, the "Last start" and "Last shutdown" data will be accurate. If NetView was not running when the startup/shutdown occurred, the data may be inaccurate.

To display any regions that are defined to the subsystem, select option 2 from the Inquire Subsystems Components panel. Figure 38 on page 159 shows the panel for an IMS control region, and Figure 39 on page 159 shows the panel for a DB control region.

```
EVIKQA00   IMS Automation: Subsystem/Defined Regions Display   Page:   1 of   1
                                                               Date: 05/22/00
Resource/Domain  => IMS711C4/APL/KEY1          (? for list) Time: 17:04:30
                                                             Domain: IPSNM
 Subtype: CTL      Subsystem: IMS10A1   JOB: IMS10AA    Status: UP
 Outstanding Reply: 11      AVM: ON     XRF: YES    DC Status: UP
 XRF Mode: ACTIVE   XRF Status: ACTIVE      ALTSYS: ON   Last HSBID: 1
-------------------------------------------------------------------------------
Subtype Region    Job       Status      Subtype Region    Job       Status
------- --------- --------  ----------  ------- --------- --------  --------
DBRC    DBRIMSAA  DBRIMSAA  UP          |
DLS     DLIIMSAA  DLIIMSAA  UP          |
TP      MSGIMSAA  MSGIMSAA  UP          |
TP      MSGIMSAB  MSGIMSAB  UP          |
TP      MSGIMSAC  MSGIMSAC  UP          |
FP      FPIMSA1   FPIMSA1   STOPPED     |
BMP     BMPIMSA1  BMPIMSA1  STOPPED     |
                                        |
                                        |
                                        |
                                        |
                                        |
Command ===>
 F1=Help    F2=End     F3=Return   F4=IMS Menu   F5=Refresh   F6=Roll
```

Figure 38. Subsystem/Defined Regions Display Panel for an IMS Control Region

All defined regions display whether or not they are currently active. In Figure 38,
the subsystem is IMS10A1 and the job is IMS10AA. For each region, the panel
displays the subtype, region, job name, and status. Subtypes include:

DBRC   Database recovery control region
DLS   Data language interface control region
TP   Normal online message region (transaction processing)
FP   Fast path message region
BMP   Batch message processing region

```
EVIKQAD0   IMS Automation: Subsystem/Defined Regions Display   Page:   1 of   1
                                                               Date: 05/22/00
                                                               Time: 17:04:30
 Subtype: CTL      Subsystem: IMS01C    JOB: IMS401C    Status: UP
 IMSID:   I41C      Command Character: ?

-------------------------------------------------------------------------------
Subtype Region    Job       Status      Subtype Region    Job       Status
------- --------- --------  ----------  ------- --------- --------  --------
DBRC    DBRIMSC   DBRIMSC   UP          |
DLS     DLIIMSC   DLIIMSC   UP          |
BMP     BMPIMSA1  BMPIMSA1  STOPPED     |
BMP     BMPIMSCA  BMPIMSCA  STOPPED     |
BMP     BMPIMSCB  BMPIMSCB  STOPPED     |
BMP     BMPIMSCC  BMPIMSCC  UP          |
                                        |
                                        |
                                        |
                                        |
                                        |
                                        |
Command ===>
 F1=Help    F2=End     F3=Return   F4=IMS Menu   F5=Refresh   F6=Roll
```

Figure 39. Subsystem/Defined Regions Display Panel for a DB Control Region

To display only the dependent regions that are currently active, select option 3
from the Inquire Subsystems Components panel. Figure 40 on page 160 shows the

panel for an IMS control region, and Figure 41 shows the panel for a DB control region.

```
EVIKQD00  IMS Automation: Subsystem/Active Regions Display    Page:   1 of   1
                                                              Date: 05/22/02
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 17:05:00
                                                              Domain: IPSNM
     Subsystem Job: IMS10AA        DBRC Job: DBRIMSAA     DLI Job: DLIIMSAA
  Subsystem Status: UP           DBRC Status: ACTIVE     DLI Status: ACTIVE
        DC Status: UP              XRFMODE: ACTIVE       XRF Status: ACTIVE
-------------------------------------------------------------------------------
     Subsystem Name: IMS10A1               Active Regions: 4
  ID    Type   Job         Taskname    ID    Type   Job         Taskname
  --    ----   --------    ---------   --    ----   --------    --------
  4     TP     MSGIMSAB    WAITING     |
  3     TP     MSGIMSAC    WAITING     |
  2     TP     MSGIMSAA    WAITING     |
  1     BMP    PPIIMSA     PPIIMSA     |
                                       |
                                       |
                                       |
                                       |
                                       |
                                       |
                                       |
Command ===>
F1=Help     F2=End       F3=Return   F4=IMS Menu  F5=Refresh   F6=Roll
                         F9=Autoref
```

Figure 40. Subsystem/Active Regions Display for an IMS Control Region

```
EVIKQDD0  IMS Automation: Subsystem/Active Regions Display    Page:   1 of   1
                                                              Date: 05/22/00
                                                              Time: 17:05:00
     Subsystem Job: IMS401C        DBRC Job: DBRIMSC      DLI Job: DLIIMSC
  Subsystem Status: UP           DBRC Status: ACTIVE     DLI Status: ACTIVE
  Command Character: ?

-------------------------------------------------------------------------------
     Subsystem Name: IMS401C               Active Regions: 1
  ID    Type   Job         Taskname    ID    Type   Job         Taskname
  --    ----   --------    ---------   --    ----   --------    --------
  1     BMP    BMPIMSCC    WAITING     |
                                       |
                                       |
                                       |
                                       |
                                       |
                                       |
                                       |
                                       |
Command ===>
F1=Help     F2=End       F3=Return   F4=IMS Menu  F5=Refresh   F6=Roll
                         F9=Autoref
```

Figure 41. Subsystem/Active Regions Display for a DB Control Region

To display the reason code associated with an XRF takeover, select option 5 from the Inquire Subsystem Components panel.

```
EVIKQT00            IMS Automation: Takeover Reason Code        Page:   1 of   1
                                                                Date: 05/22/02
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 17:06:00
                                                              Domain: IPSNM
Subsystem Name   . . . IMS10AA


        Takeover Reason Code: 54

        Takeover Cause:
            04  -  IRLM failure.
            10  -  Timeout of LOG surveillance.
            40  -  Timeout of RDS surveillance.
```

Figure 42. Takeover Reason Code Panel

Chapter 8. Starting and Stopping Resources

IMS Automation uses the INGREQ command of SA OS/390 for starting and stopping resources. For information on INGREQ, see *System Automation for OS/390 Operator's Commands*. In this section, only the special points you have to observe when starting or stopping an IMS resource will be described in detail.

To select a resource, refer to "Selecting a Subsystem" on page 156.

Start

When you select option 2, Start, from the main menu panel, the INGREQ command dialog of SA OS/390 is displayed.

```
 INGKYRU0                  SA OS/390 - Command Dialogs
 Domain ID  = IPSFM    ---------- INGREQ  ----------    Date = 07/21/00
 Operator ID = SCHR                                     Time = 11:39:44

   Resource    =>  IMS711C4/APL/KEY1                 format: name/type/system
   System      =>               System name, domain ID or sysplex name

   Request     => START      Request type (START, UP or STOP, DOWN)
   Type        => NORM       Type of processing (NORM/IMMED/FORCE/user) or ?
   Scope       => ONLY       Request scope (ONLY/CHILDREN/ALL)
   Priority    => LOW        Priority of request (FORCE/HIGH/LOW)
   Expire      =>          ,        Expiration date(yyyy-mm-dd), time(hh:mm)
   Timeout     => 0   / MSG    Interval in minutes / Option (MSG/CANCEL)
   AutoRemove  =>                    Remove when (SYSGONE, UNKNOWN)
   Restart     => NO         Restart resource after shutdown (YES/NO)
   Override    => NO                    (ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)
   Verify      => YES        Check affected resources (YES/NO/WTOR)
   Precheck    => YES        Precheck for flags and passes (YES/NO)
   Appl Parms  =>


 Command ===>
   PF1=Help    PF2=End     PF3=Return                    PF6=Roll
                                                         PF12=Retrieve
```

Figure 43. Input Panel for the INGREQ Command

The IMS-specific features concern the **Type** and the **Appl Parms** fields:

Type   In this field, you can specify the startup type. The IMS-specific values are:

Startup Type	Explanation
AUTO	Uses the restart data set to determine the startup type.
BUILDQ	All messages on the queues are saved across restarts.
COLD	Initiates a cold start.
NORM	This is the default. It is specified in the **Default start type** field of the IMS-specific IMS ENVIRON policy item of the respective application.
WARMSDBL	Loads the Main Storage Data Base (MSDB) during a warm start.

To see the start types that have been defined for the subsystem to be started, enter a question mark in the **Type** field and press ENTER. You see a panel similar to the following:

```
AOFKSEL3                   SA OS/390 - Command Dialogs    Line   1   of 5
Domain ID   = IPSFM      ---------- INGREQ  ----------    Date = 04/10/02
Operator ID = ASTA                                        Time = 16:39:22

The following start types are defined for IMS721C4/APL/KEY2
Select one item to be processed, then press ENTER.

        Sel    Start types
        ---    -----------
               AUTO
               BUILDQ
               COLD
               NORM
               WARMSDBL




Command ===>
    PF1=Help     PF2=End      PF3=Return
    PF6=Roll                                          PF12=Retrieve
```

Enter s in the **Sel** column to select the desired type.

Note: When you select a startup type that is valid for IMS, but has not been defined in the STARTUP policy item of the target resource, INGREQ issues the command defined for the NORM startup type in the STARTUP item. If that entry does not exist either, the command MVS START *jobname* is issued.

Appl Parms

In this field, you can pass IMS-specific information to IMS. One application of this is in connection with the MANUAL start type. When you select this start type, you can define your own startup command by coding /*user_ims_cmd* in the **Appl Parms** field.

You can specify more than one parameter in this field. The entries must be separated by a blank or a comma.

If you have not changed the default value of YES for the **Verify** field, IMS Automation will display a verification panel (see Figure 44 on page 165) after you have pressed ENTER. This panel displays the target resource and in addition all the resources which SA OS/390 will try to start because the startability of the selected resource directly or indirectly depends on them.

```
AOFKVFY1                  SA OS/390 - Command Dialogs      Line  1   of 3
Domain ID   = IPSFM      ---------- INGREQ   ----------    Date = 05/03/00
Operator ID = SCHR                                         Time = 12:34:11

 Verify list of affected resources for request START

CMD: S show overrides   T show trigger details   V show votes
Cmd Name         Type System   TRG SVP  W  Action Type    Observed Stat
--- ----------- ---- -------- --- ---- - ------ -------- -------------
    IMSIMSZ      APL  KEY2               Y         AUTO    UNAVAILABLE
    JES2         APL  KEY2                         AUTO    UNAVAILABLE
    VTAM         APL  KEY2                         AUTO    UNAVAILABLE






Command ===>
 PF1=Help  PF2=End    PF3=Return                           PF6=Roll
                                    PF10=GO    PF11=CANCEL  PF12=Retrieve
```

Figure 44. Verification Panel for INGREQ

For more information on the verification panel of INGREQ, see *System Automation for OS/390 Operator's Commands*.

Note: For starting message regions, use the INGLIST command of SA OS/390.

Shutdown

When you select option 3, Shutdown, from the main menu panel, the INGREQ command dialog of SA OS/390 is displayed.

```
INGKYRU0                  SA OS/390 - Command Dialogs
Domain ID   = IPSFM      ---------- INGREQ   ----------    Date = 05/03/00
Operator ID = SCHR                                         Time = 16:43:22

  Resource    =>  IMSIMSZ/APL/KEY2                 format: name/type/system
  System      =>             System name, domain ID or sysplex name

  Request     => STOP       Request type (START, UP or STOP, DOWN)
  Type        => NORM       Type of processing (NORM/IMMED/FORCE/user) or ?
  Scope       => ONLY       Request scope (ONLY/CHILDREN/ALL)
  Priority    => LOW        Priority of request (HIGH/LOW)
  Expire      =>        ,           Expiration date(yyyy-mm-dd), time(hh:mm)
  Timeout     => 0   / MSG   Interval in minutes / Option (MSG/CANCEL)
  AutoRemove  =>                   Remove when (SYSGONE, UNKNOWN)
  Restart     => NO         Restart resource after shutdown (YES/NO)
  Override    => NO                 (ALL/NO/TRG/FLG/DPY/STS/UOW/INIT)
  Verify      => YES        Check affected resources (YES/NO/WTOR)
  Precheck    => YES        Precheck for flags and passes (YES/NO)
  Appl Parms  =>


Command ===>
  PF1=Help     PF2=End     PF3=Return                       PF6=Roll
                                                            PF12=Retrieve
```

Figure 45. Input Panel for INGREQ Command

The IMS-specific features concern the **Type** and the **Appl Parms** fields:

Type   The possible types for a STOP request are NORM, IMMED, and FORCE. For each of these, a shutdown command must be specified in the

SHUTDOWN policy item of the respective application (IMSIMSZ in the example panel). This is the command that will be issued by INGREQ.

Appl Parms

You can specify the following parameters in this field:

Shutdown option

When an IMS subsystem is defined as a control region, the shutdown commands require a so-called shutdown *option* as one of their parameters. Shutdown options are associated with an additional command (through the SHUTTYPES keyword, see "SHUTTYPES—Issue commands at operator shutdown" on page 106), and the command that is associated with the specified option will be issued during the shutdown process. A default option must be coded in the shutdown commands (see "Shutdown" on page 44). However, you can override that default. To do that you must specify the option of your choice in the **Appl Parms** field according to the following format

```
OPTION={DUMPQ|BACKUP|FREEZE|PURGE|DUMP|NODUMP}
```

For a description of the shutdown options see the IMS documentation.

Broadcast

This parameter specifies whether or not the operators are notified that the subsystem is going to be closed. The format is

```
BROADCAST={YES|NO|CTL}
```

NO is the default.

Note: For stopping message regions, use the INGLIST command of SA OS/390.

Chapter 9. TCO Functions

The master terminal functions provide a full-screen panel interface to perform TCO functions. Select option 9, TCO Management, to display the **TCO Main Menu**:

```
 EVIKMT00      IMS Automation: TCO Main Menu                  Page:   1 of   1
                                                              Date: 06/21/02
 Resource/Domain  => IMS721C4                  (? for list) Time: 12:17
                                                            Domain:


 TCO Status . . . . . . . . :

              1  Load a specific member
              2  Enable TCO processing
              3  Disable TCO processing
              4  View the Control file definitions






 Command ===>
 F1=Help    F2=End     F3=Return   F4=IMS Menu            F6=Roll
```

Figure 46. TCO Main Menu

The Timer-Controlled Operations (TCO) interface lets you issue TCO commands from the IMS Automation interface.

From this TCO Main Menu, you can perform four functions:

1. Load a specific member
2. Enable TCO processing
3. Disable TCO processing
4. View the control file definitions

TCO Main Menu Option 1 — Load a Specific Member

Select Option 1 from the TCO Main Menu to load a specific member. Panel EVIKMT10 is displayed. Enter a '?' to see a member list:

```
EVIKMT10        IMS Automation: TCO Member Load              Page: 1 of 1
                                                             Date: 05/23/00
Resource/Domain  => IMS711C4/APL/KEY1          (? for list) Time: 09:28:30
                                                             Domain: IPSNM

TCO Status . . . . . . . : Available

Member . . . . . . . . . ?_____     (? for list)

Alternate Destination  . . _____

Maximum Segment Size . . .  __




Command ===>
F1=Help    F2=End     F3=Return   F4=IMS Menu           F6=Roll
```

Figure 47. TCO Member Load Panel

Select a member:

```
EVIKMT10        IMS Automation: TCO Member Load              Page: 1 of 1
                                                             Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1          (? for list) Time: 09:29:00
                                                             Domain: IPSNM

TCO Status . . . . . . . : Available
        +------------------------------------------------------------
Member  |   Select one of the user supplied TCO members
        |   _ STRTLNES       START LINES
Alterna |   s ASGNTRAN       ASSIGN TRANSACTIONS
        |   _ DISPROG        DISPLAY ACTIVE PROGRAMS
Maximum |
        |
        |
        |
        |
        |
        |
        |
        |  F1=Help    F3=Cancel
        +------------------------------------------------------------
Command ===>
F1=Help    F2=End     F3=Return   F4=IMS Menu           F6=Roll
```

Figure 48. TCO Member Load Panel

The member name is inserted. Press ENTER to load:

```
EVIKMT10        IMS Automation: TCO Member Load           Page: 1 of 1
                                                          Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1          (? for list) Time: 09:29:30
                                                          Domain: IPSNM

TCO Status . . . . . . . : Available

Member . . . . . . . . . ASGNTRAN    (? for list)

Alternate Destination  . . _____

Maximum Segment Size . . .  __




Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu          F6=Roll
```

Figure 49. TCO Member Load Panel

You will receive a message that the command is in progress:

```
EVIKMT10        IMS Automation: TCO Member Load           Page: 1 of 1
                                                          Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1          (? for list) Time: 09:30:00
                                                          Domain: IPSNM

TCO Status . . . . . . . : Available

Member . . . . . . . . . ASGNTRAN    (? for list)

Alternate Destination  . . _____

Maximum Segment Size . . .  __




 COMMAND IN PROGRESS
Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu          F6=Roll
```

Figure 50. TCO Member Load Panel

TCO Main Menu Option 2 — Enable TCO Processing

Select Option 2 from the TCO Main Menu to enable TCO processing:

```
EVIKMT00        IMS Automation: TCO Main Menu              Page: 1 of 1
                                                           Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1        (? for list) Time: 09:30:30
                                                           Domain: IPSNM


TCO Status . . . . . . . . : Available

Select an option . . . . . 2 1  Load a specific member
                             2  Enable TCO processing
                             3  Disable TCO processing
                             4  View the Control file definitions











Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu            F6=Roll
```

Figure 51. TCO Main Menu

You will receive a message when the START command has completed:

```
EVIKMT00        IMS Automation: TCO Main Menu              Page: 1 of 1
                                                           Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1        (? for list) Time: 09:31:00
                                                           Domain: IPSNM


TCO Status . . . . . . . . : Available

Select an option . . . . . _ 1  Load a specific member
                             2  Enable TCO processing
                             3  Disable TCO processing
                             4  View the Control file definitions








DFS058I 10:00:44 START COMMAND COMPLETED   I31A
Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu            F6=Roll
```

Figure 52. TCO Main Menu - Enable TCO Processing

TCO Main Menu Option 3 — Disable TCO Processing

Select Option 3 from the TCO Main Menu to disable TCO processing:

```
 EVIKMT00         IMS Automation: TCO Main Menu                   Page: 1 of 1
                                                                  Date: 05/23/02
 Resource/Domain  => IMS711C4/APL/KEY1               (? for list) Time: 09:32:00
                                                               Domain: IPSNM


 TCO Status . . . . . . . . : Available

 Select an option . . . . . . 3 1  Load a specific member
                                2  Enable TCO processing
                                3  Disable TCO processing
                                4  View the Control file definitions










 Command ===>
 F1=Help     F2=End      F3=Return   F4=IMS Menu             F6=Roll
```

Figure 53. TCO Main Menu

You will receive a message when the PSTOP command has completed:

```
 EVIKMT00         IMS Automation: TCO Main Menu                   Page: 1 of 1
                                                                  Date: 05/23/00
 Resource/Domain  => IMS711C4/APL/KEY1               (? for list) Time: 09:32:30
                                                               Domain: IPSNM


 TCO Status . . . . . . . . : Available

 Select an option . . . . . . _ 1  Load a specific member
                                2  Enable TCO processing
                                3  Disable TCO processing
                                4  View the Control file definitions









 DFS058I 10:02:05 PSTOP COMMAND COMPLETED   I31A
 Command ===>
 F1=Help     F2=End      F3=Return   F4=IMS Menu             F6=Roll
```

Figure 54. TCO Main Menu - Disable TCO processing

TCO Main Menu Option 4 — View TCO related Message IDs

Select Option 4 from the TCO Main Menu to view the definitions of the TCO and
TCOMEMBERS reserved message IDs (see "TCO—Issue commands for time-driven
procedures" on page 114 and "TCOMEMBERS—Define TCO Members" on
page 115):

```
EVIKMT00        IMS Automation: TCO Main Menu                Page: 1 of 1
                                                             Date: 05/23/00
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 09:33:00
                                                             Domain: IPSNM


TCO Status . . . . . . . . : Available

Select an option . . . . . . 4 1  Load a specific member
                               2  Enable TCO processing
                               3  Disable TCO processing
                               4  View the Control file definitions










Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu              F6=Roll
```

Figure 55. TCO Main Menu

The following panel will be displayed. Select Option 1 to view the TCO command
entries:

```
EVIKMT40       IMS Automation: TCO Control File Display Menu   Page: 1 of 1
                                                               Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 09:33:30
                                                               Domain: IPSNM


TCO Status . . . . . . . . : Available

Select an option . . . . .  1 1  View the TCO command entries
                              2  View the TCO load members










Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu              F6=Roll
```

Figure 56. TCO Control File Display Menu

The following panel will be displayed:

```
AOFK3D0X              SA OS/390 - Command Response        Line  1    of 7
Domain ID   = IPSNM  ---------- DISPACF  ----------        Date = 05/14/02
Operator ID = ABCD                                         Time = 04:51:24


Command = ACF ENTRY=IMS711C4,TYPE=TCO,REQ=DISP
SYSTEM = KEY1      AUTOMATION CONFIGURATION DISPLAY - ENTRY= IMS711C4
------------------------------------------------------------
AUTOMATION CONFIGURATION DISPLAY - ENTRY= IMS711C4
 TYPE IS TCO
 REPLY           = (INIT,,'DFSTCF LOAD DFSTCF  .')
 REPLY           = (SPEC,,'DFSTCF LOAD &EHKVAR1 .')
 REPLY           = (START,,'/START LTERM DFSTCFI .')
 REPLY           = (STOP,,'/PSTOP LTERM DFSTCFI .')
END OF MULTI-LINE MESSAGE GROUP




   Action ===>
           PF1= Help              PF3= Return           PF5= Refresh
           PF6= Roll
```

Figure 57. TCO Configuration Display Panel

Select Option 2 from the TCO Control File Display Menu to view the TCO load members:

```
 EVIKMT40     IMS Automation: TCO Control File Display Menu     Page: 1 of 1
                                                                Date: 05/23/02
 Resource/Domain  => IMS711C4/APL/KEY1          (? for list) Time: 09:35:00
                                                              Domain: IPSNM
 TCO Status . . . . . . . . : Available

 Select an option . . . . .  2 1  View the TCO command entries
                               2  View the TCO load members










 Command ===>
 F1=Help     F2=End     F3=Return   F4=IMS Menu            F6=Roll
```

Figure 58. TCO Control File Display Menu

The following panel will be displayed:

```
AOFK3D0X                SA OS/390 - Command Response    Line  1   of 6
Domain ID   = IPSNM     ---------- DISPACF ----------   Date = 04/08/02
Operator ID = SCOT                                      Time = 06:43:41

Command = ACF ENTRY=IMS711C4,TYPE=TCOMEMBERS,REQ=DISP
SYSTEM = KEY1     AUTOMATION CONFIGURATION DISPLAY - ENTRY= IMS631C4
-------------------------------------------------------------------------------
AUTOMATION CONFIGURATION DISPLAY - ENTRY= IMS7311C4
 TYPE IS TCOMEMBERS
 NAME          = (STRTLNES,'START LINES')
 NAME          = (ASGNTRAN,'ASSIGN TRANSACTIONS')
 NAME          = (DISPROG,'DISPLAY ACTIVE PROGRAMS')
END OF MULTI-LINE MESSAGE GROUP






Command ===>
    PF1=Help      PF2=End       PF3=Return              PF6=Roll
                                PF9=Refresh             PF12=Retrieve
```

Figure 59. TCO Configuration Display Panel

Chapter 10. Displaying Critical Messages

Through the Critical Message Manager, IMS Automation utilizes the SA OS/390 Status Display Facility (SDF). The Critical Message Manager displays critical messages in a prioritized, scrollable format. Messages are defined as critical in the MAT.

Select option 7 from the main menu to display Figure 60.

```
KEY1I    IMS MONITOR PANEL




        Archive

        MSC Links

        OLDS

        Recons



        Transactions

        CF Structures



                                                        06/18/02 10:38
 ===>
PF1=HELP 2=DETAIL 3=END 6=ROLL 7=UP 8=DN                    12=TOP
```

Figure 60. Critical Messages Manager Panel. If critical messages have accumulated for a category, the category will be highlighted.

SDF's IMS Monitor Panel, shown in Figure 60, lists categories of critical messages. If the CMM has accumulated any critical messages for a category, that category will be highlighted, according to severity of message.

To View Critical Messages

1. Tab to the category you wish to view.
2. Press PF2 to display the detailed list.

Figure 61 shows a sample critical message display.

```
              ---- DETAIL STATUS DISPLAY ----
                                              1 OF    6


      COMPONENT: DXR008E           SYSTEM   : SY1

      COLOR    : YELLOW            PRIORITY :     501

      DATE     : 05/22/00          TIME     : 17:07:30

      REPORTER : AUTO1             NODE     : CNM01

      REFERENCE VALUE: IMS10AA-DXR0108E

       JOB IMS10AA - DXR008E IRLM INITIALIZATION ABEND




  ===>
 1=HELP   3=RETURN       6=ROLL 7=UP 8=DOWN 9=ASSIST 10=DELETE 11=BOTTOM 12=TOP
```

Figure 61. Detail Display of Critical Message

Chapter 11. Broadcasting Messages, Issuing Commands, and Listing Information for an IMS Resource

With the INGIMS command dialog you can issue any IMS operator commands, broadcast messages to all or selected IMS users, and display information about IMS resources.

For a detailed description of the INGIMS command, refer to *System Automation for OS/390 Operator's Commands*.

```
EVIKYCMD                    SA OS/390 - Command Dialogs      Line
Domain ID   = IPSFM      ---------- INGIMS  ----------    Date = 04/08/02
Operator ID = ASTA                                        Time = 16:41:03

Resource          => IMS721C4/APL/KEY2              Format: name/type/system
System            =>                      System name, domain ID or sysplex name
Request           => BROADCAST            CMD, BROADCAST or INFO
IMS Command       => BROADCAST
IMS Route         => ACTIVE
IMS Message       =>
                  =>




Command ===>
   PF1=Help      PF2=End      PF3=Return   PF4=DISPINFO         PF6=Roll
                              PF9=Refresh                       PF12=Retrieve
```

Figure 62. INGIMS Command Dialog

You can specify the following:

Resource
> Specifies the name of an IMS resource. The format is name/type/system or name/type for sysplex resources. Wildcard characters are supported.

System
> Is the name of the system (system name, domain id, or syplex name) to which the command is issued. This is only necessary when the resource is not part of the local sysplex.

Request
> Specifies the request to be issued to the IMS subsystem. It can be one of the following: CMD, BROADCAST or INFO.

IMS Command
> Specifies the IMS transaction and its parameters to be executed. The IMS command field is automatically filled with the text above. You can make changes to it before pressing Enter to execute the transaction.

IMS Route
> Specifies the routing information. The routing information format is the same as specified for the /BRO command.

IMS Message
> Specifies the message to be sent to all or selected IMS users or terminals.For example, you can notify users of a planned shutdown.

With PF4 DISPINFO, you can display detailed information about the specified subsystem.

Chapter 12. The Status Display Facility

The Status Display Facility uses color to represent the various subsystem resource statuses such as error, warning, action, or informational states. Typically, a subsystem shown in green on a Status Display Facility status panel indicates that it is up, whereas red indicates a stopped or problem state.

The Status Display Facility status display panels can be tailored to present the status of system components in a hierarchical manner. The hierarchical display of status information is implemented using tree structures. A tree structure always starts with the system name as the root component. The "leaves" of the tree are the monitored resources.

Color can be propagated up or down the leaves of the tree structure based on the order of dependencies. The effect of propagation is to consolidate, at the root component, the status of all the monitored resources in that system. In this way, the color of the root component reflects the most important or critical status in a computer operations center. If all the monitored resources are green, the root component (the system) will be green.

IMS Automation provides additional Status Display Facility panels that monitor events that occur in the following areas for all IMS regions defined to IMS Automation:

Archive
> Shows the archive status of OLDS datasets.

MSC Links
> Shows the Status of MSC Links

OLDS Shows the error status of OLDS datasets.

Recons
> Shows the error status of RECONS datasets.

Transactions
> Shows any transactions that have encountered an error.

CF Structures
> Shows coupling facility structures used by IMS that have errors.

To use the IMS Automation Status Display Facility panels, enter SDF on a NetView panel command line. A panel similar to the following is displayed:

```
SYSTEM      SA OS/390 - SUPPORT SYSTEMS

 System  Subsystems   WTORs     Gateways   Products  System

 KEY1    IM631C4      NETBTST1  IPSFNO     C I D O   S C M B T U

 KEY2                                      C I D O   S C M B T U

 KEY3                                      C I D O   S C M B T U

 KEY4                                      C I D O   S C M B T U

 XXXX                                      C I D O   S C M B T U




                                                06/18/02 10:45
 ===>
 1=HELP 2=DETAIL 3=RETURN   6=ROLL   8=NEXT SCR   10=LEFT 11=RIGHT 12=TOP
```

Figure 63. Status Display Facility Main Panel

Note: Sample Status Display Facility panels are provided with IMS Automation.
The system programmer customizes the panels for your specific
environment, so the panels shown here will not look exactly like your
panels.

This could be your primary panel that lists the systems and their status. The color
of KEY1 through KEY4 will reflect the most critical status of any resource in that
system.

If you place the cursor under the letter C on the panel displayed in Figure 63 and
press PF8, the following panel displays (assuming you are using the default
sample panels):

```
KEY1I     IMS MONITOR PANEL



        Archive

        MSC Links

        OLDS

        Recons


        Transactions

        CF Structures


                                                06/18/02 10:38
 ===>
 PF1=HELP 2=DETAIL 3=END 6=ROLL 7=UP 8=DN                    12=TOP
```

Figure 64. IMS Monitor Panel

This shows several categories in which IMS status is important. If the letter C
shown on the previous panel was red, then at least one of the items on the IMS
Monitor panel will be red. Tab down to the red item and press PF8. This displays
the messages logged against that item, as shown in the following panel:

```
KEY1IOL1  IMS OLDS

 System  Message text
 06/18/02  10:59:31  IMS712CX  " IMS712CX NEEDS ANOTHER OLDS DATASET - OLDS SHO




                                                            06/18/02 10:59
 ===>
PF1=HELP 2=DETAIL 3=RET        6=ROLL 7=UP              11=RT 12=TOP
```

Figure 65. IMS Monitor Panel

Note: If the full message is not displayed on the screen, press PF11 to shift to the
right.

To see the detail of a message, tab down to that message and press PF2. This
displays a panel similar to the following:

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


       COMPONENT: IMS712CX             SYSTEM   : KEY1

       COLOR    : PINK                 PRIORITY :     601

       DATE    : 06/18/02             TIME    : 10:59:31

       REPORTER : KAT                  NODE    : IPSFM

       REFERENCE VALUE: IMS712CX_OL_DFS3260

        " IMS712CX NEEDS ANOTHER OLDS DATASET - OLDS SHORTAGE"




 ===>
1=HELP  3=RETURN 4=DELETE  6=ROLL 7=UP 8=DOWN 9=ASSIST 11=BOTTOM 12=TOP
```

Figure 66. Detail Status Display

To delete a message, press PF4 on this screen.

Note: If any of the panels have 1 of X in the upper-right corner of the screen, where X is a number greater than 1, subsequent panels contain additional data.

Press PF8 to scroll forward to view the information. Press PF7 to scroll back.

Chapter 13. NMC Display Support

The following messages and events will be displayed on NMC against the subsystem where they occur.

The alerts will be attached to the subsystem as a minor resource and will have the following resource names:

Messages	plexname.resname/APL/sysname.MSG/message_id
MSC Links	plexname.resname/APL/sysname.MSC/linkname
OLDS	plexname.resname/APL/sysname.OLDS/DFS3258 plexname.resname/APL/sysname.OLDS/DFS3260 plexname.resname/APL/sysname.OLDS/SWITCHING plexname.resname/APL/sysname.OLDS/ARCHIVE/ddname plexname.resname/APL/sysname.OLDS/ERROR/ddname plexname.resname/APL/sysname.OLDS/SPARES plexname.resname/APL/sysname.OLDS/MINIMUM
Recons	plexname.resname/APL/sysname.RECONS/ddname plexname.resname/APL/sysname.RECONS/SPARE plexname.resname/APL/sysname.RECONS/DUAL
Transactions	plexname.resname/APL/sysname.TRANS/tranname
CQS	plexname.resname/APL/sysname.CQS/structure

Glossary of IMS Automation Terms

This glossary defines special IMS terms used in the library and words used with other than their everyday meaning. In some cases, a definition may not be the only one applicable to a term, but it gives the particular sense in which it is used in the IMS Automation Option library.

abend. Abnormal end of task.

ACB. Access Method Control Block (VTAM and VSAM).

access method. A technique for moving data between main storage and input/output devices.

ANSI. American National Standards Institute.

AOST. Automated Operator Station Task.

APAR. Authorized program analysis report.

application program. A program written for or by a user that applies to the user's work. In data communication, a program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

automation. Computer system control of operation processes.

authorized program analysis report (APAR). A request for correction of a problem caused by a defect in a current unaltered release of a program.

batch. An accumulation of data to be processed.

batch message processing. In IMS/VS, a batch processing program that accesses online data bases and message queues.

BMP. Batch Message Processing Region.

CCTL. Coordinator Controller.

central processing complex (CPC). A conglomeration of several processors and other devices in one or more physical units. This usually means several processors running under the control of a single MVS/ESA operating system. For example, a 3090 model 400 processor complex can run as a four-processor CPC, or it can be partitioned into the equivalent of two 3090 model 200s, each of which runs as a CPC with its own operating system.

CICS. Customer Information Control System.

CLIST. Command List.

CMM. Critical Message Manager.

CNM. Communications Network Management.

command. In IMS, an instruction similar in format to a high-level programming language statement.

command list (CLIST). A list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in NetView Command List Language.

common state handler (CSH). Routine that IMS Automation calls from the NetView Automation Table to drive the actions defined in state/action tables.

concurrent. Pertaining to the occurrence of two or more activities within a given interval of time.

CPC. Central Processing Complex.

critical message manager (CMM). Facility in IMS Automation which displays critical messages in a scrollable format and enables operators to access information relating to the critical messages displayed.

CSA. Common Storage Area.

CSH. Common State Handler.

database. A collection of data fundamental to a system.

database backout. The function of removing changes made to user data sets by in-flight transactions.

database recovery. The function of restoring the user data sets, starting with a backup copy and applying all changes made to each data set after the backup was taken.

data security. The protection of data against unauthorized disclosure, transfer, modifications, or destruction, whether accidental or intentional.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DBCTL. Data Base Control.

DEDB. Data Entry Data Base.

DLISAS. Data Language Interface Separate Address Space (IMS Batch).

domain. In IMS, a set of subsystems on a specific NetView domain defined by the system programmer in the control file.

end user. In IMS, anyone using IMS to do a job, usually by interacting with an application program (transaction) by means of a terminal.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file, or using any resource.

Fast Path. IMS Automation function which enables the user to access any IMS Automation interface panel by entering = and an identifying number.

Fast Path Message Region. In IMS, a region that executes programs that require good response characteristics and that have large transaction volumes. Message processing is grouped for load balancing and synchronized for database integrity and recovery.

focal point system. In IMS, a system in which multiple subsystems are interconnected. One subsystem serves as a focal point of control, and the others are referred to as intermediate or distributed systems.

HM. Help Message.

HSBID. Hot Standby Identifier.

HSSP. High-Speed Sequential Processing.

initial program load (IPL). The initialization procedure that causes an operating system to commence operation.

initialization. Actions performed by IMS to construct the environment in the IMS region to enable IMS applications to be run. A process started by SA OS/390 and IMS Automation to construct the environment in which automation will occur.

installation. A particular computing system, in terms of the work it does and the people who manage it, operate it, apply it to problems, service it and use the work it produces. The task of making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it.

Installation Verification Procedure (INSTALL/IVP). Procedure distributed with the system that tests the newly generated system to verify that the basic facilities are functioning correctly.

INSTALL/IVP. Install/Installation Verification Procedure.

Integrated Resource Lock Manager (IRLM). In IMS Automation, this facility is used as a lock manager, both as a single lock manager and in a data sharing environment.

intercommunication facilities. A generic term covering intersystem communication (ISC) and multiregion operation (MRO).

intersystem communication (ISC). Communication between separate systems by means of SNA networking facilities or by means of the application-to-application facilities of an SNA access method. ISC links IMS systems, and it may be used for user application-to-user application communication, or for transparently executing IMS functions on a remote IMS system.

IPL. Initial Program Load.

IRC. Interregion communication.

IRLM. Integrated Resource Lock Manager.

ISC. Intersystem Communication.

IVP. Installation Verification Procedure.

keyword. A symbol that identifies a parameter. A part of a command operand that consists of a specific character string.

local. In data communication, pertaining to devices that are attached to a CPC by cables, rather than data links.

local device. A device, such as a terminal, whose control unit is directly attached to a computer's data channel. No data link is used. Contrast with remote device.

lock manager. Feature of IMS Automation responsible for serializing the recovery process in areas where multiple subsystems can invoke recovery actions.

member. See partitioned data set.

MPP. Message Processing Program.

MSC. Multiple Systems Coupling.

MSDB. Main Storage Data Base.

MTO. Master Terminal Operator.

Multiple Systems Coupling (MSC). An IMS/VS feature that permits geographically dispersed IMS/VS systems to communicate with each other.

NCCF. Network Communications Control Facility.

network. An interconnected group of nodes. The assembly of equipment through which connections are made between data stations.

network configuration. In SNA, the group of links, nodes, machine features, devices, and programs that make up a data processing system, a network, or a communication system.

Network Communications Control Facility (NCCF). IBM licensed program consisting of a base for command processors that can monitor, control, and improve network operations.

non-XRF (non-XRF IMS). Represent IMS in a non-XRF configuration.

NPDA. Network Problem Determination Aid/Application.

OLDS. Online Log Data Set.

online. Pertaining to a user's ability to interact with a computer. Pertaining to a user's access to a computer via a terminal.

panel. In IMS Automation, the set of information displayed on a single screen of the user interface.

parameter. (ISO) A variable that is given a constant value for a specified application and that may denote the application.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Synonymous with program library.

PDS. Partitioned Data Set.

PPI. Program-to-program interface.

preprocessor. Routine in IMS Automation that enables the programmer to define unique GLOBALV names to store the state value of certain processes.

processor (ISO). In a computer, a functional unit that interprets and executes instructions.

Program-to-program interface (PPI). A NetView component used by IMS Automation to enable users to send or receive data buffers from other programs. It also allows system and application programs to send alerts to the NetView hardware monitor.

PTF. Program Temporary Fix.

PUT. Program update tape.

RACF. Resource Access Control Facility.

RDS. Restart Data Set.

RECON. Recovery Control.

recovery routine. A routine entered when an error occurs during the performance of an associated operation. It isolates the error, assesses the extent of the error, and attempts to correct the error and resume operation.

remote. In data communication, pertaining to devices that are connected to a data processing system through a data link.

remote device. A device, such as a terminal, connected to a data processing system through a data link.

remote system. In IMS intercommunication, a system that the local IMS system accesses via intersystem communication or multiregion operation.

Resource Access Control Facility (RACF). A licensed program that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging detected unauthorized access attempts, and logging detected accesses to protected data sets.

RMF. Resource Management Facility.

roll. In IMS Automation, the option to begin/rollover to another NetView session. This action is assigned to the PF6 key.

SDF. Status Display Facility. The display facility for SA OS/390.

security. Prevention of access to or use of data or programs without authorization.

service. The carrying out of effective problem determination, diagnosis, and repair on a data processing system or software product.

single-point-of-control. Feature of IMS Automation enabling the operator to monitor and control IMS subsystems from a single NetView console.

SLDS. System Log Data Set.

SMU. Security Maintenance Utility.

SNA. Systems Network Architecture.

software. (ISO) Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system. Contrast with hardware.

startup. The operation of starting up IMS by the system operator.

state/action table. In IMS Automation, state/action tables are a matrix of system events, states, and actions created by the system programmer. When an event occurs, the system references the state/action table and takes appropriate action.

status code. In IMS/VS, a two-character code in the program communication block (PCB) mask that indicates the results of a DL/1 call.

system. In IMS, an assembly of hardware and software capable of providing the facilities of IMS for a particular installation.

system initialization table. A table containing user-specified data that will control a system initialization process.

systems network architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

task. (ISO) A basic unit of work to be accomplished by a computer. Under IMS, the execution of a transaction for a particular user.

TCO. Timer-Controlled Operations.

terminal. A point in a system or communication network at which data can either enter or leave. In IMS, a device, often equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

terminal operator. The user of a terminal.

transaction. A transaction may be regarded as a unit of processing (consisting of one or more application programs) initiated by a single request, often from a terminal. A transaction may require the initiation of one or more tasks for its execution.

update. To modify a file with current information.

VSCR. Virtual Storage Constraint Relief.

VTAM. Virtual Telecommunications Access Method. VTAM is one of the ways IMS communicates with terminals.

WTOR. Write To Operator with Reply.

XRF. Extended recovery facility, a software function that minimizes the effects of various failures on the end users.

Index

Special characters

Readers' Comments — We'd Like to Hear from You

System Automation for OS/390
IMS Automation
Programmer's Reference
and Operator's Guide
Version 2 Release 2

Publication No. SC33-7045-03

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.

IBM®

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

IBM Deutschland Entwicklung GmbH
Department 3248
Schönaicher Strasse 220
D-71032 Böblingen
Federal Republic of Germany
 72031-0000

Fold and Tape   **Please do not staple**   Fold and Tape

IBM ®

Program Number: 5645-006

Printed in U.S.A.