IBM Tivoli System Automation for z/OS

IMS Automation Programmer's Reference and Operator's Guide

Version 3 Release 1

SC33-8268-00

IBM Tivoli System Automation for z/OS

IMS Automation Programmer's Reference and Operator's Guide

Version 3 Release 1

SC33-8268-00

Note!

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

First Edition (September 2005)

This edition applies to IBM Tivoli System Automation for z/OS (5698-SA3) Version 3 Release 1, an IBM licensed program, and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

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

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

Internet: s390id@de.ibm.com

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

Contents

Figures

Tables

Notices

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

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

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

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

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

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

Programming Interface Information

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

Trademarks

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

CICS	IBM	IMS
IMS/ESA	MVS	MVS/ESA
NetView	OS/390	RACF
S/390	Tivoli	Tivoli Enterprise Console
VTAM	z/OS	

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS™ enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

About This Book

This book describes how to customize and operate IMS™ Automation. IMS Automation is a feature of IBM® Tivoli® System Automation for z/OS (SA z/OS) 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 or 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 z/OS, 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 z/OS 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.

Related Publications

The System Automation for z/OS Library

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

Table 1. System Automation for z/OS Library

Title	**Order Number**
IBM Tivoli System Automation for z/OS Planning and Installation	SC33-8261
IBM Tivoli System Automation for z/OS Customizing and Programming	SC33-8260
IBM Tivoli System Automation for z/OS Defining Automation Policy	SC33-8262
IBM Tivoli System Automation for z/OS User's Guide	SC33-8263
IBM Tivoli System Automation for z/OS Messages and Codes	SC33-8264
IBM Tivoli System Automation for z/OS Operator's Commands	SC33-8265
IBM Tivoli System Automation for z/OS Programmer's Reference	SC33-8266
IBM Tivoli System Automation for z/OS CICS Automation Programmer's Reference and Operator's Guide	SC33-8267
IBM Tivoli System Automation for z/OS IMS Automation Programmer's Reference and Operator's Guide	SC33-8268
IBM Tivoli System Automation for z/OS TWS Automation Programmer's Reference and Operator's Guide	SC23-8269
IBM Tivoli System Automation for z/OS End-to-End Automation Adapter	SC33-8271

The System Automation for z/OS books are also available on CD-ROM as part of the following collection kit:

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

SA z/OS Home Page

For the latest news on SA z/OS, visit the SA z/OS home page 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 z/OS base program.

Table 2. Related Products Books

Title	**Order Number**
CICS Transaction Server for OS/390 V1.3 Release Guide	GC34-5352
CICS Transaction Server for z/OS Release Guide	GC34-5983
CICS Transaction Server for z/OS V2.3 Release Guide	GC34-6218
CICS Transaction Server for z/OS V3.1 Release Guide	GC34-6421
DB2 Release Planning Guide	SC26-9943
DB2 UDB for z/OS V8 Release Planning Guide	SC18-7425
IMS V8 Release Planning Guide	GC27-1305
IMS V9 Release Planning Guide	GC17-7831
z/OS ISPF Dialog Developer's Guide	SC34-4821
z/OS ISPF Dialog Tag Language Guide and Reference	SC34-4824
z/OS ISPF User's Guide Vol I	SC34-4822
z/OS ISPF User's Guide Vol II	SC34-4823
z/OS MVS Planning APPC Management	SA22-7599

Table 2. Related Products Books (continued)

Title	**Order Number**
MVS/ESA™ Application Development Macro Reference	GC28-1822
z/OS MVS System Commands	SA22-7627
z/OS Hardware Configuration Definition (HCD) Planning	GA22-7525
z/OS Information Roadmap	SA22-7500
z/OS Introduction and Release Guide	GA22-7502
z/OS Licensed Program Specifications	GA22-7503
z/OS Migration	GA22-7499
z/OS MVS Setting Up a Sysplex	SA22-7625
z/OS Summary of Message and Interface Changes	SA22-7625
z/OS Security Server RACF Command Language Reference	SC28-0733
z/OS Parallel Sysplex Overview: An Introduction to Data Sharing and Parallelism	SA22-7661
z/OS Parallel Sysplex Application Migration	SA22-7662
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 z/OS Administration Reference	SC31-8854
Tivoli NetView for z/OS AON Customization Guide	SC31-8871
Tivoli NetView for z/OS AON User's Guide	GC31-8851
Tivoli NetView for z/OS Application Programming Guide	SC31-8855
Tivoli NetView for z/OS Automation Guide	SC31-8853
Tivoli NetView for z/OS Command Reference Vol 1	SC31-8857
Tivoli NetView for z/OS Command Reference Vol 2	SC31-8858
Tivoli NetView for z/OS Customization Guide	SC31-8859
Tivoli NetView for z/OS Customization: Using Assembler	SC31-8860
Tivoli NetView for z/OS Customization: Using PL/I and C	SC31-8861
Tivoli NetView for z/OS Customization: Using REXX and CLIST Language	SC31-8862
Tivoli NetView for z/OS Data Model Reference	SC31-8864
Tivoli NetView for z/OS Installation: Configuring Additional Components	SC31-8874
Tivoli NetView for z/OS Installation: Configuring Graphical Components	SC31-8875
Tivoli NetView for z/OS Installation: Getting Started	SC31-8872
Tivoli NetView for z/OS Installation: Migration Guide	SC31-8873
Tivoli NetView for z/OS Licensed Programming Specifications	GC31-8848
Tivoli NetView for z/OS Messages and Codes	SC31-8866
Tivoli NetView for z/OS MSM User's Guide	GC31-8850
Tivoli NetView for z/OS NetView Management Console User's Guide	GC31-8852
Tivoli NetView for z/OS RODM and GMFHS Programming Guide	SC31-8865
Tivoli NetView for z/OS Security Reference	SC31-8870
Tivoli NetView for z/OS SNA Topology Manager and APPN Accounting Manager Implementation Guide	SC31-8868

Table 2. Related Products Books (continued)

Title	Order Number
Tivoli NetView for z/OS User's Guide	GC31-8849
Tivoli Workload Scheduler for z/OS V8.1 General Information	GH19-4539
Tivoli Workload Scheduler for z/OS V8.2 General Information	SC32-1256
z/OS TSO/E REXX Reference	SA22-7790
z/VM V5R1.0 Group Control System	SC24-6098
VSE/ESA V2R4.0 Unattended Node Support	SC33-6712
VTAM® V4R4 Messages	GC31-8368
VTAM V4R4 Codes	GC31-8369
VTAM V4R4 Network Implementation Guide	SC31-8370
z/OS Language Environment Run-Time Messages	SA22-7566
zSeries® Application Programming Interfaces	SB10-7030

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM 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 use LookAt from these locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux™:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX® System Services).
- Your Microsoft® Windows® workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T4271).
- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Part 1. Introducing IMS Automation

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

Chapter 1. Principal Concepts of SA z/OS

This section sketches some fundamentals of SA z/OS. For more detailed information see the SA z/OS 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 z/OS in an automation *policy database*. This database contains the objects to be managed by SA z/OS, 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 *IBM Tivoli System Automation for z/OS 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 z/OS 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 z/OS.

Goal-Driven Automation

A basic concept of SA z/OS 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 z/OS always tries to keep the actual state in line with the desired state, but sometimes this is not possible.

SA z/OS 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 z/OS, this is a request to set the desired state of the resource to AVAILABLE. It is up to SA z/OS 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 z/OS 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 z/OS 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 z/OS 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 z/OS 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.

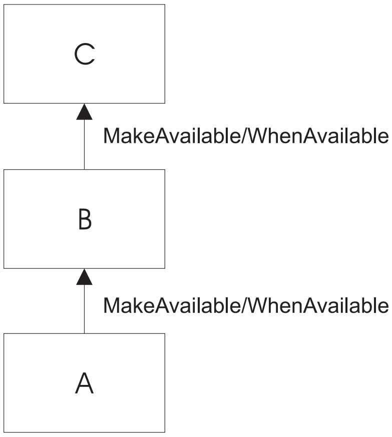


Figure 1. Example of Start Dependencies

When SA z/OS 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 z/OS 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 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 z/OS 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 z/OS 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 z/OS 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 z/OS 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.

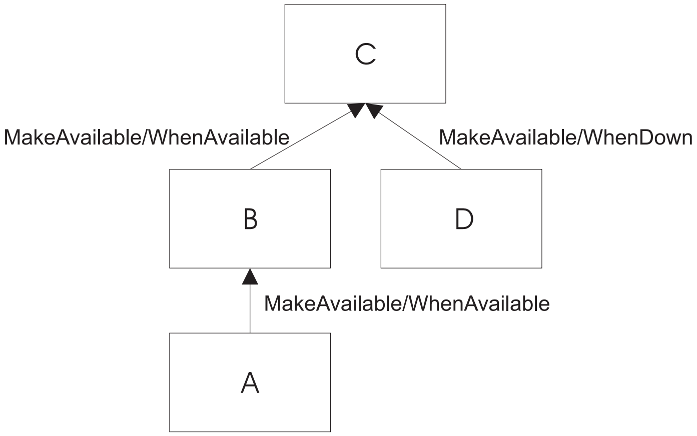


Figure 2. Example of Conflicting Requests

Now assume that first a request to start A and then a request to start D are passed to SA z/OS. 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 z/OS 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 z/OS 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 z/OS 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 z/OS, 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 z/OS 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 z/OS event represents an external event that is not under control of SA z/OS, 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 z/OS by *setting* or *unsetting* the SA z/OS 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 z/OS, 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 z/OS 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 z/OS 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 z/OS 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.

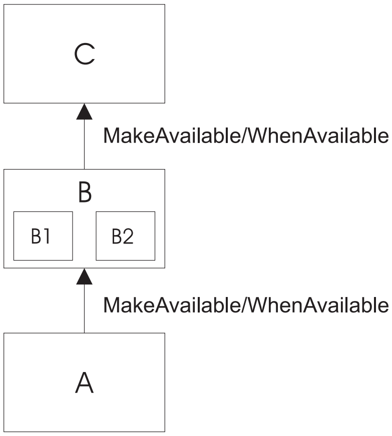


Figure 3. Example of a Request Involving a Group

Then, if you request A to be started, SA z/OS 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 z/OS 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 z/OS and the NetView Automation Table

The implementation of SA z/OS is based on NetView. One important area, where SA z/OS relies on NetView functionality, is the NetView Automation Table (AT). 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 AT. If it finds an entry for the message, it issues the command specified in that entry.

With applications controlled by SA z/OS, the command will typically be one of the generic routines that are shipped with SA z/OS (see *IBM Tivoli System Automation for z/OS 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 *IBM Tivoli System Automation for z/OS Defining Automation Policy*). If you connect this message ID with the generic routine ISSUECMD in the AT, 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 *IBM Tivoli System Automation for z/OS 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 AT 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 AT, see *Tivoli NetView for OS/390® Automation Guide*. IMS Automation also has some special generic routines, see Chapter 6, "Common Routines," on page 89.

Chapter 2. Special Functions of IMS Automation

IMS Automation is integrated into SA z/OS. 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 *optionally* uses the NetView PPI, among other things, to send IMS messages to NetView that drive resulting command lists or command processors. The standard method, however, is to use the REXX adapter rather than the PPI.

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).

Users may prefer to call the SA z/OS supplied exit EVInAOIX (alias DFSAOUE0) from their own user exit instead of the reverse as described above. In this case the user exit will call the EVInAOIX module (not the DFSAOUE0 alias which should be removed) and must provide an identical environment to that described in the IMS Customization manual for the IMS TYPE 1 AO exit. Also, the user exit must not modify the environment returned by the SA z/OS supplied exit or SA z/OS may not function correctly.

EVInAOIX requests storage from IMS via return code 16 when the message or /LOG command being processed is to be routed to the AO PPI BMP (EVISPINM AUTO=YES). If the SA z/OS exit is called from a user exit, and a storage request is made, then the user exit must honor the storage request as IMS would and redrive EVInAOIX with the entry registers set as described below.

EVInAOIX expects the following registers to be set on entry:

R0	**0** Initial Entry (first segment) **4** Subsequent entries (middle segment) **8** Final Entry (last segment) **12** Storage obtained (UEHUBUFF contains the address of the storage obtained). **16** No message is presented to the exit.
R1	Address of the UEHB.
R7	Address of the communication terminal block (CTB).
R9	Address of the communication line block (CLB) or partition specification table (PST).
R11	Address of the system contents directory (SCD).
R13	Address of the save area.
R14	Return address to IMS.
R15	Entry point of exit routine.

EVInAOIX sets the following registers and return codes on exit:

RC=0 Insert and ENQ this segment	
R0	Address of the alternate destination (PPI BMP).
R1	Address of the message segment to be queued.
R15	**0** Insert and ENQ this segment.
RC=4 Ignore this segment	
R15	**4** Ignore this segment.
RC=8 Processing complete - ENQ this (or previous) segments	
R1	**0** or address of the message segment.
R15	**8** Processing complete.
RC=12 Cancel this message	
R15	**12** Cancel this message.
RC=16 Storage request	
R1	Size of storage requested.
R15	**16** Storage request.

Refer to *IMS Version 8: Customization Guide* for more detail on IMS Type 1 Exit entry and exit requirements.

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 shutdown.

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 32.

Recovery Overview

IMS Automation does not provide any action related to FDR recovery.

Assume that IMS is running on system *SYS1* and FDR is running on system *SYS2*. The sequence of events during IMS Control Region failure and FDR recovery 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 *SYS1* completes I/O Prevention, message AVM006E is issued by the Availability Manager on *SYS1*.
3. IMS FDR automatically detects that the AVM006E message has been issued and that I/O Prevention is complete and cancels message DFS4167A. This allows FDR on *SYS2* to commence recovery processing.

If, for any reason, message AVM006E is not issued and I/O Prevention does not complete, then message DFS4167A remains outstanding and manual operator intervention is required.

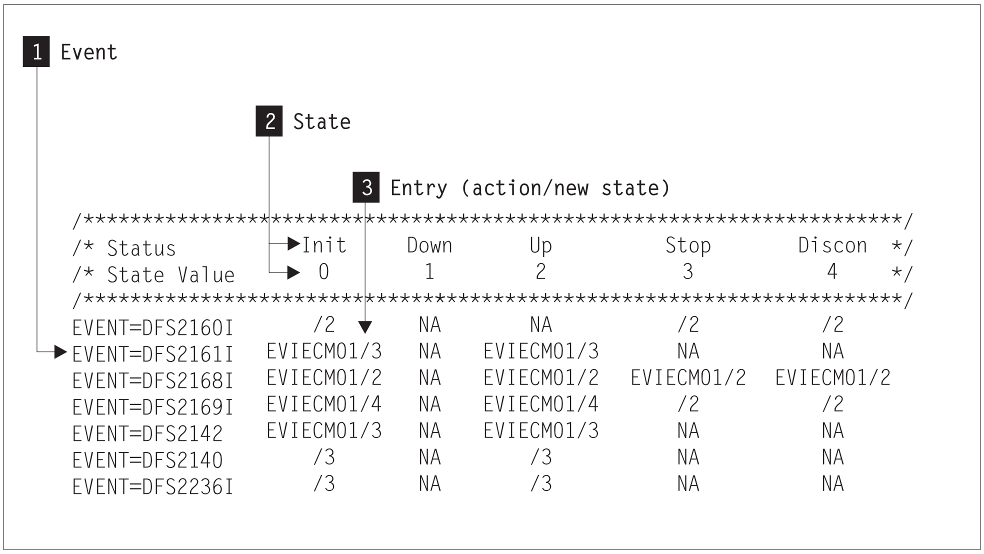
Shutdown Overview

The FDR address space will automatically terminate after recovery is complete. The FDR address space will also terminate automatically whenever the IMS Control region is shut down.

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 AT.

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 4 shows a sample state/action table.



```
/************************************************************************/
/* 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 4. 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.

IMS Message Processing

SA z/OS can only automate messages that are issued via WTO. Most IMS system messages that are important are WTO’d, However, there are many IMS messages that are only logged internally. Some of these messages may be useful in automation situations. To enable SA z/OS to process these messages, exits are installed to WTO messages that would not be WTO’d by IMS.

In addition, user code might produce messages that are written to IMS. Some of these messages might be of interest in automation situations. SA z/OS installs an exit to WTO these messages. SA z/OS installs an exit for the AOE Type 2 exit of IMS Control regions.

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 3.1 only supports IMS V8 and higher.

Chapter 3. Customizing IMS Automation

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

IMS Automation Definitions

You customize IMS Automation for your specific installation by modifying the policy database in the customization dialog. To show you what kind of definitions you need in your policy database, SA z/OS comes with an add-on sample policy database named *IMS.

See *IBM Tivoli System Automation for z/OS Defining Automation Policy* for:

- How to import the add-on sample *IMS
- How to modify the definitions that this import has added to your policy database

Step 1: Provide Basic IMS Automation Common Policy Definitions

Substep 1a: Define the Automation Operators

The *IMS add-on sample policy database supplies all automation operator definitions that are required for IMS automation as well as some that are optional. They all reside in the IMS_AUTO_OPS policy object in entry type AOP. The ones that are required are the following:

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

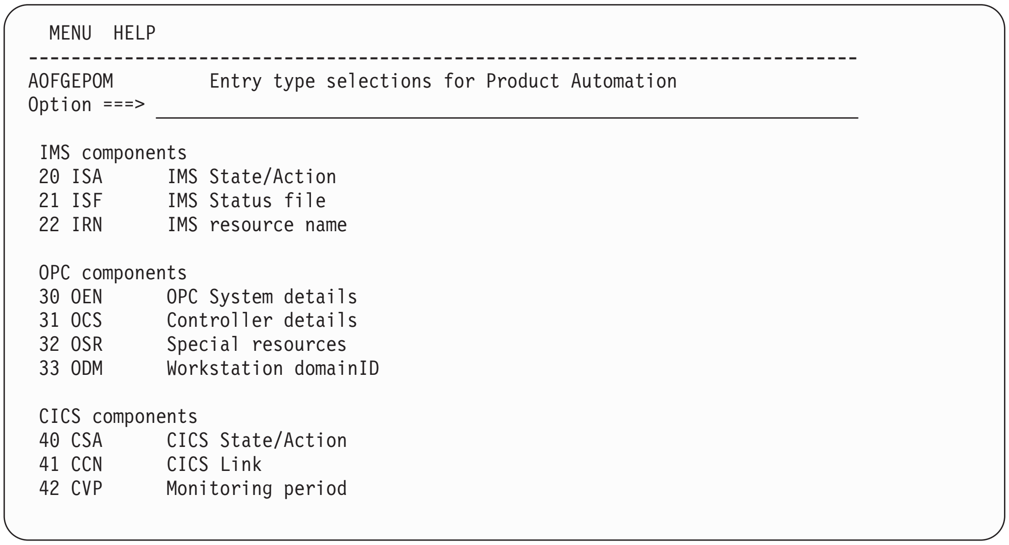
Note: Make sure that these operator IDs are defined in the DSIOPF member in the DSIPARM data set of NetView.

Substep 1b: Define State/Action Tables

Define sets of State/Action Tables in the ISA entry type as described in “How to Set Up the State/Action Tables” on page 45.

Note:

To find the ISA entry type, start by selecting the PRD entry type on the ***Entry Type Selection*** panel. This takes you to the ***Entry type selections for Product Automation*** shown in Figure 5 on page 20, where the ISA entry type resides.



```
  MENU  HELP
------------------------------------------------------------------------------
AOFGEPOM         Entry type selections for Product Automation
Option ===> __________________________________________________________________

 IMS components
 20 ISA      IMS State/Action
 21 ISF      IMS Status file
 22 IRN      IMS resource name

 OPC components
 30 OEN      OPC System details
 31 OCS      Controller details
 32 OSR      Special resources
 33 ODM      Workstation domainID

 CICS components
 40 CSA      CICS State/Action
 41 CCN      CICS Link
 42 CVP      Monitoring period
```

*Figure 5. **Entry type selections for Product Automation** Panel (to be found via entry type PRD)*

Step 2: Define IMS Regions

All IMS regions must be defined to SA z/OS as APPLICATION objects in the customization dialog; 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.

The *IMS add-on sample policy database provides you with a number of sample classes and instances of entry type APL that model IMS regions. Copy and modify them to tailor them to your needs.

- The names of the APL classes in *IMS can be recognized by this prefix:

  `CLASS_IMS_`
- The names of the APL instances in *IMS can be recognized by this prefix:

  `IMS`

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 *IBM Tivoli System Automation for z/OS 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 39.
- STATE ACTION TABLE

This item serves to link an application to a set of state/action tables that you created in “Substep 1b: Define State/Action Tables” on page 19. The State/Action Tables are used for component recovery. For more details see “How to Set Up the State/Action Tables” on page 45.

The following subsections inform you about special customization aspects 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 49.

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

Required	Keyword	Comments	See Page
✓	ABCODEPROG	Respond to BMP Region Abends.	50
✓	ABCODETRAN	Transaction Abend Recovery.	51
	BRO	IMS control regions only. Code to issue the appropriate broadcast message prior to shutdown.	52
	CHE	Issue a Checkpoint Command.	53
✓	CQS0031A	Confirm CQS Restart for Structure.	54
✓	CQS0032A	Respond to CQS Structure Restart.	55
✓	CQS0033A	Respond to Client Takeover Restart.	56
✓	CQSET	Issue Structure Checkpoint at CQS Termination.	57
✓	DFS2142	Respond to Stopped Logical Link Path Message.	58
✓	DFS2161I	Link Stopped by Other System.	59
✓	DFS2169I	Respond to MSC Link Disconnection Message.	60
✓	DFS3258A	No Online Data Sets Available.	61
✓	DFS554A	Respond to Program Abend.	62
✓	DFS810A	IMS control regions only. Code for response to message DFS810A.	63
✓	DFS989I	DB control regions only. Code for response to message DFS989I.	64
✓	DFS994I	Code for actions after message DFS994I.	65
	HOLDQ	Code to hold BMP initiators.	67
✓	IMSINFO	Display Information.	68
✓	OLDS	Define Recovery Criteria for OLDS.	69

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

Required	Keyword	Comments	See Page
	POSTCHKP	Code to reflect your installation’s procedure to issue commands after a shutdown checkpoint has been issued.	71
	PRECHKP	Code to reflect your installation’s procedure to issue commands prior to a shutdown checkpoint being issued.	72
✓	RECONS	Set Monitoring Interval for RECONS.	73
	RELEASEQ	Code to release BMP initiators.	75
	RESTARTABORT	Code to reflect desired response to the receipt of messages DFS0618 and DFS166I.	76
✓	SHUTTYPES	None.	77
	SNAPQ	XRF only. Code to reflect your installation’s procedure to issue the /SNAPQ command.	79
✓	STOPBMPREGION	IMS control regions only.	80
✓	STOPFPREGION	IMS control regions only.	82
✓	STOPREGION	IMS control regions only.	83
	TCO	None.	85
	TCOMEMBERS	None.	86
✓	TPABEND	Code exactly as shown with the appropriate subsystem identifier.	87

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 32.

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 33.

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 FINAL, 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 *IBM Tivoli System Automation for z/OS 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 49.

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.	69
✓	DFS3258A	Action to take on last OLDS.	61

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. For more information on IMS-specific MESSAGES/USER DATA keywords, see Chapter 5, "MESSAGES/USER DATA Entries for IMS Automation," on page 49.

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.	58
✓	DFS2161I	Code this entry to restart a link after it has been stopped by an IMS system.	59
✓	DFS2169I	Code this entry to restart a link after disconnection of a Multiple Systems Coupling (MSC) link between two IMS systems.	60

Step 5: Code the Entries for RECONs Recovery

For RECON recovery, specify the MESSAGES/USER DATA keywords shown in Table 6 on page 24. For more information on IMS-specific MESSAGES/USER

DATA keywords, see Chapter 5, “MESSAGES/USER DATA Entries for IMS Automation,” on page 49.

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.	73

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 49.

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.	50
✓	ABCODETRAN	Code this entry to specify system action in response to transaction abend codes.	51
✓	DFS554A	Code this entry to restart a transaction and program after an abend.	62

Step 7: Extended IMS Automation Definitions

If you want to 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 z/OS 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 z/OS status file in the **Partner status file** field.

4. The SA z/OS sample INGALLC2 specifies the DEFINE CLUSTER control statement used to establish the SA z/OS 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 (Optional): Defining NetView PPI Receiver Task

A PPI receiver *may* be used to allow IMS subsystems to communicate with NetView.

1. For sample definitions and classes, refer to the *IMS sample add-on policy.
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. Specify a HASPARENT relationship with the NetView SSI subsystem, if defined to SA.
5. Specify a Where Used APG that is connected to every system that runs an SA z/OS NetView Agent. This ensures that this resource will be created for every SA z/OS 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

This function is optional and is not required for most users.

1. For sample definitions and classes, refer to the *IMS sample add-on policy.
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 relationship: hasParent → IMS subsystem.
5. Specify a Where Used APG that the parent IMS subsystem is a member of.

Installing the IMS Message Exits

IBM Tivoli System Automation for z/OS Planning and Installation details the basic installation steps to install the exits into IMS. This section details the various parameters and commands that can be used to control the exits.

z/OS Exit Router Information

The out-of-the-box configuration of the exit uses the z/OS exit router to enable the user to specify exit modules at three exit points. The exit program is EVIPVEX0 and has a pre-built alias of DFSAOE00. The exit points defined are:

1. **DFSAOE00.CMD**

   This is invoked whenever DFSAOE00 is called with AOE0FUNC = 1 to initialize the routines and also when AOE0FUNC = 2 is called with AOE0FLG2 = X'80' (command entered at a terminal), X'20' (ICMD command) or X'10' (internal command).
2. **DFSAOE00.MSG**

   This is invoked whenever DFSAOE00 is called with AOE0FUNC = 1 to initialize the routines and also when AOE0FUNC = 2 is called with AOE0FLG2 = X'08' (message segment).
3. **DFSAOE00.CMDRESP**

This is invoked whenever DFSAOE00 is called with AOE0FUNC = 1 to initialize the routines and also when AOE0FUNC = 2 is called with AOE0FLG2 = X'40' (command response segment).

Specification of routines to run at each exit point is done via PROGxx members of SYS1.PARMLIB or via the SETPROG EXIT command.

You may enable or disable exits dynamically at any time.

A required definition of EVIPVEX1 is needed to process the exit information for SA z/OS. However, any number of exit routines may be added at any of the points. The z/OS exit router will execute them one after the other. If exit routines are not specified for an exit point, no action will be taken.

The definition of the required exit point is:

```
EXIT ADD EXITNAME(DFSAOE00.MSG) MODNAME(EVIPVEX1) DSNAME('ING.SINGMOD1') STATE(ACTIVE)
```

See *z/OS MVS Initialization and Tuning Reference* for additional parameters that can be supplied to the EXIT statement.

The SA z/OS exit routines EVIPVEX0 and EVIPVEX1 use the last three words of the storage pointed to by SXPLAWRK. If these values are changed, then unpredictable results may occur.

Using the SA z/OS Exit without the z/OS Exit Router

As detailed in the installation manual, it is possible to use the System Automation exit in a stand alone manner. In this mode, no z/OS exit router is enabled and dynamic management of the exit is not possible. The procedure to enable this function is detailed in the System Automation installation manual. It basically re-defines the DFSAOE00 alias from the EVIPVEX0 module to the EVIPVEX1 module.

Note:
Make sure that the ING.SINGMOD1 library is not concatenated before of the library that you have redefined the alias in. If it is, then the z/OS exit router module will be enabled at the DFSAOE00 exit point.

Calling the SA z/OS Exit from Your DFSAOE00 Module

If neither of the above methods suits your installation, it is possible to directly call the EVIPVEX1 module from your DFSAOE00 exit routine.

Input parameters are:

- 1. Register 1 points to the SXPL (standard Exit Parameter List) as supplied by IMS to the DFSAOE00 exit.

Used storage: the exit uses the last three words of the storage pointed to by SXPLAWRK are used by the exit to hold pointers to work area and policy information. If the values are changed, then unpredictable results may occur.

A sample call is as follows:

```
LA R1,SXPL   ; Load address of SXPL
L R15,=V(EVIPVEX1)  ; get address of routine
BALR R14,R15   ; invoke the exit
```

Using the IMS Automation Message Policy

To enable the IMS message exit to process messages, the messages must be defined in the policy database. The Messages/User Data policy item is used to define the messages. If there are no messages defined to use the exits, the SA z/OS exit will not process any messages.

Message policy data is processed on a subsystem basis. However, you can put the policy information in a CLASS and link the class to the subsystems to save on data entry.

Each subsystem has its own policy and does not share policy information, except for CLASS information, with other subsystems. This means that you can update the policy information for a subsystem or set of subsystems and will not affect other subsystems.

Defining IMS Messages

IMS messages that are not already WTO'd can be WTO'd by specifying the message in the customization dialog.

There are several special keywords that, when added to the message via the USER policy, will cause the message information to be loaded into the appropriate exit for processing. The keywords and their descriptions are:

Table 8. Keywords in IMS Message Definitions

Keyword	Description
OFFSET	This keyword is required to load the message into the exit. It specifies the position in the message that represents the message id. Its format is a single integer number. For instance, an OFFSET of 3 means: the third word in the message.
TOKEN	This keyword specifies the matching token values for user messages. It is optional, but if present only messages that match the values specified will be WTO'd. Its format is ►►—(—▼(*number*,*value*)—)—►◄ where *number* is the position to be checked and *value* is the text to be checked. Positions are specified by counting words from the beginning of the message starting from 1. A word is considered to be a contiguous set of alpha-numeric characters. All non-alpha-numeric characters are treated as delimiters. The value specification is checked for the length specified. This allows for trailing data to be ignored. The format of the number is a single integer. The format for the value is a single alpha-numeric word, no blanks or special characters are allowed. The CASE of the value is ignored.

To define a message to be WTO'd , do the following:

Step 1. Specify the message id in the MESSAGES/USER DATA policy item.

Step 2. Specify the OFFSET keyword in the USER part of the policy with the data being the number of the word position that the message id occurs at in the message.

Step 3. Optionally specify the TOKEN keyword in the USER part of the policy with the data being the words and their values to be matched.

Refreshing policy data

To load the information into IMS address spaces issue the INGAMS REFRESH command. As a part of the ACF load, each IMS that had changes to MESSAGES/USER DATA policy will be reloaded with any changes.

If you want to disable the exits, either delete the messages out of the policy or change the OFFSET keyword on every message to some other name, e.g. UFFSET, and re-build and re-load the ACF.

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 z/OS 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 *IBM Tivoli System Automation for z/OS 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)
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 131). 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 z/OS policy database.

General Restrictions

IMS Automation supports dependency relationships as provided with the SA z/OS 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 as the lowest numbered *hasParent* relationship to enable IMS Automation to determine which control region owns the dependent region. Note that you *must* number the *hasParent* relationship to the IMS control 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 6 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.

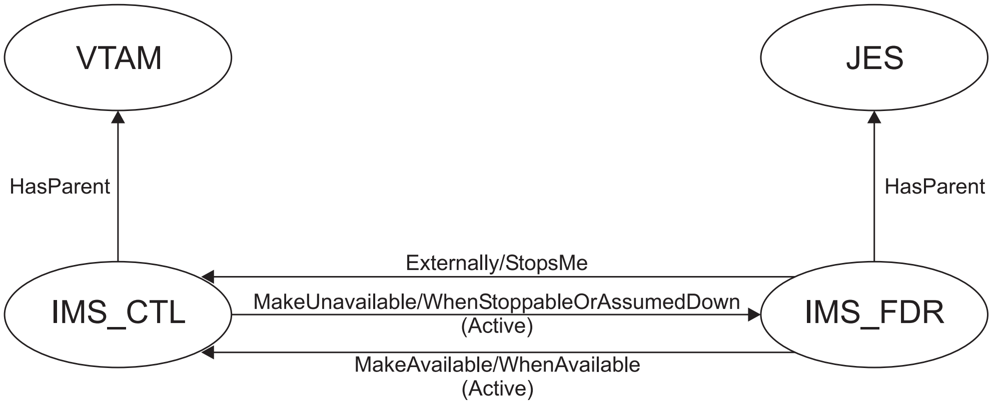


Figure 6. Relationships for IMS FDR

Note:
The IMS_FDR application should be defined with “External Shutdown” set to “FINAL” and “Restart Option” set to “ALWAYS” in the Automation Policy data base.

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 a CQS Complex

Figure 7 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.

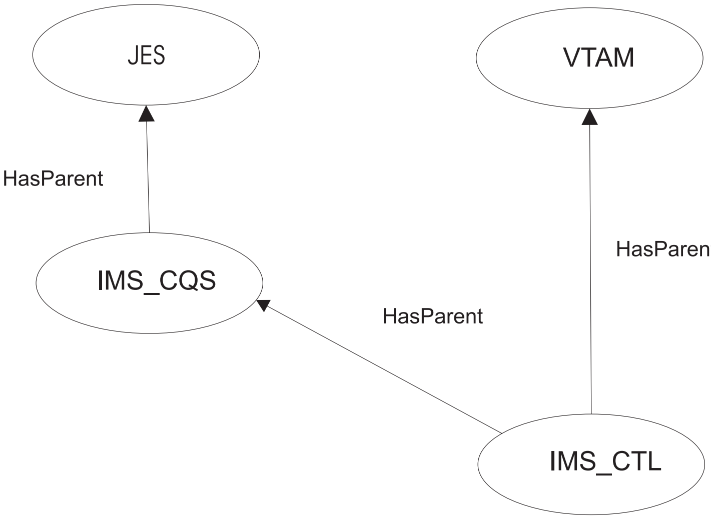


Figure 7. Relationships for IMS CQS

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 or user). However if you wish to reply to the DFS810A message using the MANUAL start type then you should enter a start command specifying AUTO=N. The **External Startup** field of the APPLICATION 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 117). 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 63).

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 77), which has to be defined in the MESSAGES/USER DATA policy item of the application.

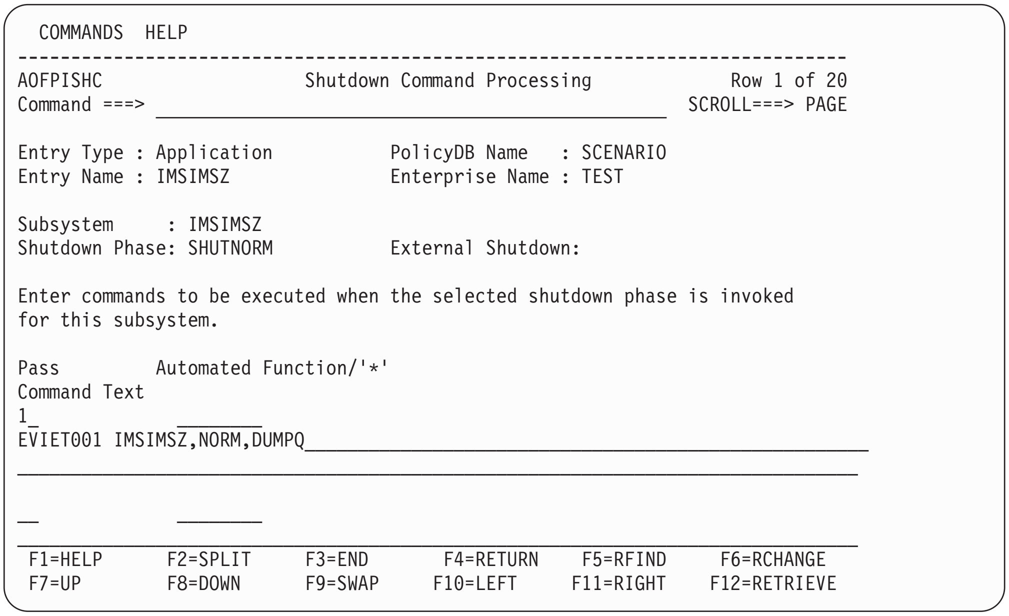
When you do a NORM or IMMED shutdown, the only options allowed are:

- For a DBCTL region: PURGE, FREEZE
- For an XRF CTL region in BACKUP mode: BACKUP
- For a CTL region: PURGE, FREEZE or DUMPQ

When you do a FORCE the only options allowed are DUMP or NODUMP.

The option specified for the respective phase is the default; you can override it when you put a shutdown request to SA z/OS 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:

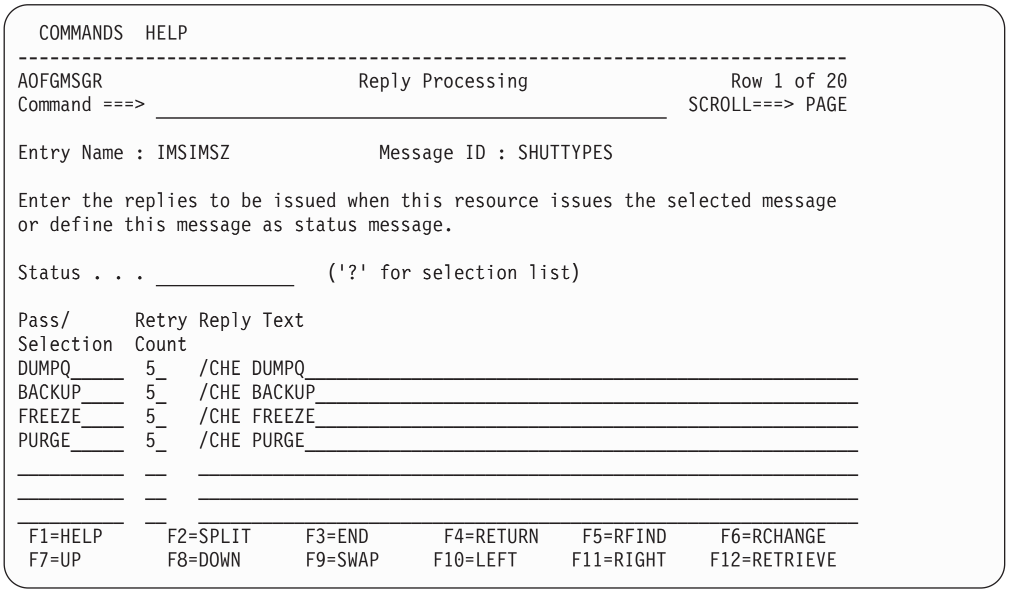


```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFPISHC                  Shutdown Command Processing             Row 1 of 20
Command ===> ________________________________________________  SCROLL===> PAGE

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

Subsystem     : IMSIMSZ
Shutdown Phase: SHUTNORM           External Shutdown:

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

Pass         Automated Function/'*'
Command Text
1_            ________
EVIET001 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 8. 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
------------------------------------------------------------------------------
AOFGMSGR                        Reply Processing                  Row 1 of 20
Command ===> ________________________________________________  SCROLL===> PAGE

Entry Name : IMSIMSZ              Message ID : SHUTTYPES

Enter the replies to be issued when this resource issues the selected message
or define this message as status message.

Status . . . _____________    ('?' for selection list)

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 9. 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 `EVIET001 IMSIMSZ,NORM,DUMPQ` is issued (see Figure 8 on page 35). This entails that the corresponding command of the SHUTTYPES entry, namely `/CHE DUMPQ`, is invoked (see Figure 9), 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 119.

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:

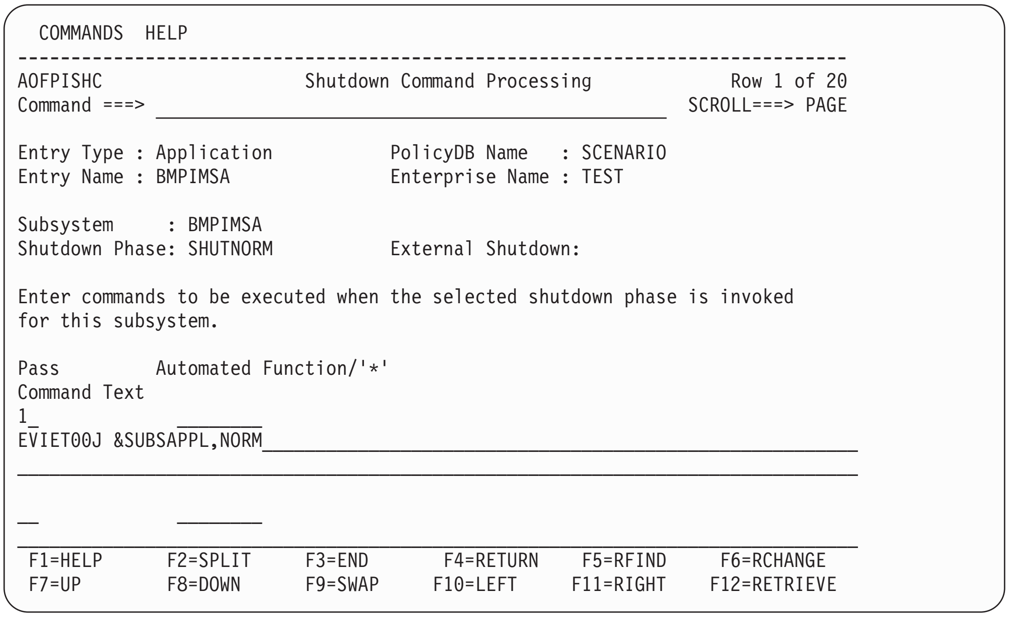
```
EVIET00J &SUBSAPPL,{NORM|IMMED|FORCE}
```

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 80.
FP regions	STOPFPREGION, see “STOPFPREGION—Stop Fast Path Regions” on page 82.
Dependent message regions	STOPREGION, see “STOPREGION—Stop IMS Dependent Message Region” on page 83.

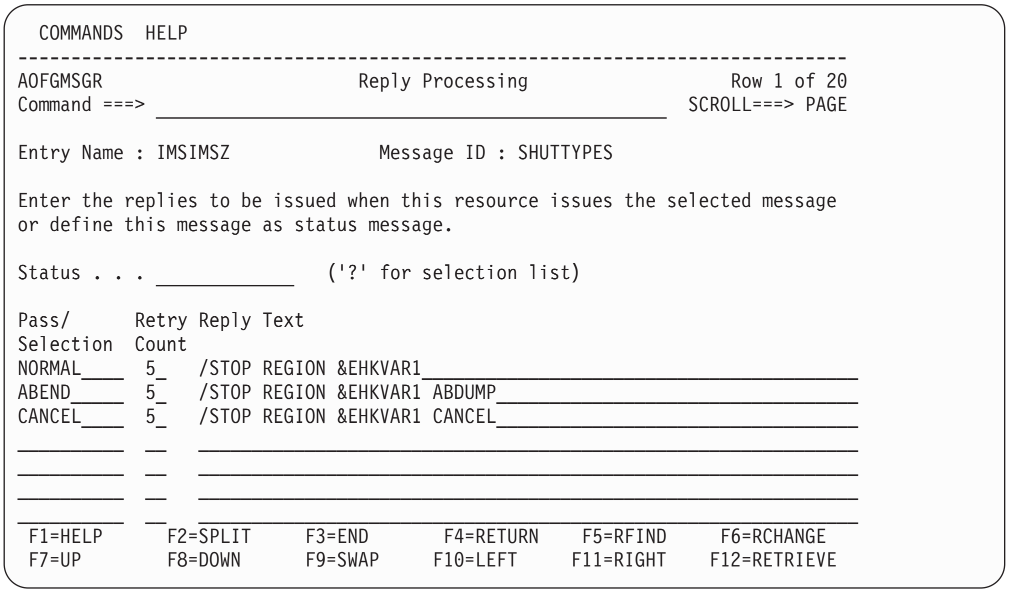
The following example illustrates the mechanism. Suppose that BMPIMSA is a BMP region, that the **External Shutdown** field of its AUTOMATION INFO policy item is not set to ALWAYS, and that the shutdown command for the NORM phase in its SHUTDOWN policy item is coded as follows:



```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFPISHC                   Shutdown Command Processing             Row 1 of 20
Command ===> ________________________________________________  SCROLL===> PAGE

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

Subsystem     : BMPIMSA
Shutdown Phase: SHUTNORM           External Shutdown:

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 10. 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
------------------------------------------------------------------------------
AOFGMSGR                        Reply Processing                  Row 1 of 20
Command ===> ________________________________________________  SCROLL===> PAGE

Entry Name : IMSIMSZ             Message ID : SHUTTYPES

Enter the replies to be issued when this resource issues the selected message
or define this message as status message.

Status . . . _____________   ('?' for selection list)

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 11. 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 z/OS, then EVIET00J is called as specified for the NORM phase of the SHUTDOWN policy item (see Figure 10 on page 37). EVIET00J determines that BMPIMS is a BMP region, consults the STOPBMPREGION entry of the associated IMS control region IMSIMSZA (see Figure 11), 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.

IMSPlex Support Subsystems

IMSPlex support that can be automated are the SCI, OM and RM.

Class definitions are in the *SYSPLEX sample.

All subsystems are defined as MVS subsystems.

The following classes can be used as a template or can be linked to for basic automation functions:

- For the SCI: CLASS_IMS_SCI
- For the OM: CLASS_IMS_OM
- For the RM: CLASS_IMS_RM

IMSPlex support regions must be defined so that they are started and running before any IMS regions that depend on their services are started.

In general this means that the SCI, OM and RM regions should be started before the Control Region, CQS and FDR.

The suggested approach is to put the three regions in a System Basic Group, with possibly the IRLM as well.

In the case of the RM, this can be a single plex-wide address space. If you choose to run with a single RM supporting the IMSPLex, then the suggested approach is to put the RM in a sysplex Move group that is separate from the SCI and OM basic group.

The suggested relationships are:

- For the RM in a Move Group:

  HasParent SCI and RM Basic group on each system, for example:

  ```
  HasParent SCIBasic/APG/=
  ```
- For the OM and RM in the same Basic Group as the SCI:

  HasParent SCI application, for example:

  ```
  HasParent SCI/APL/=
  ```
- For the OM in the same Basic Group as the SCI:

  HasParent SCI application. for example:

  ```
  HasParent SCI/APL/=
  ```
- For the Control Region and any other region that uses SCI/RM services:
  - If the RM is in a move group:

    ```
    HasParent RM-Move-Group/APG
    ```
  - If the RM is in the Basic Group:

    ```
    HasParent SCI-OM-RM-Group/APG/=
    ```
- For any region that uses just SCI services:

  ```
  HasParent SCIBasic/APG/=
  ```

Where:

```
SCIBasic/APG/= HasMembers SCI/APL/=, OM/APL/=
RM-Move-Group/APG HasMembers RM/APL/*
SCI-OM-RM-Group/APG/= HasMembers SCI/APL/=, OM/APL/=, RM/APL/=
```

The above relationships are suggestions only.

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 *IBM Tivoli System Automation for z/OS 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 *IBM Tivoli System Automation for z/OS 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 58), DFS2161I (see page 59), DFS2169I (see page 60)
OLDSs	OLDS (see page 69), DFS3258A (see page 61)
Transactions	ABCODETRAN (see page 51), DFS554A (see page 62)
Programs	ABCODEPROG (see page 50), DFS554A (see page 62)

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

How to Define Transaction Recovery

Customization of transaction recovery consists of:

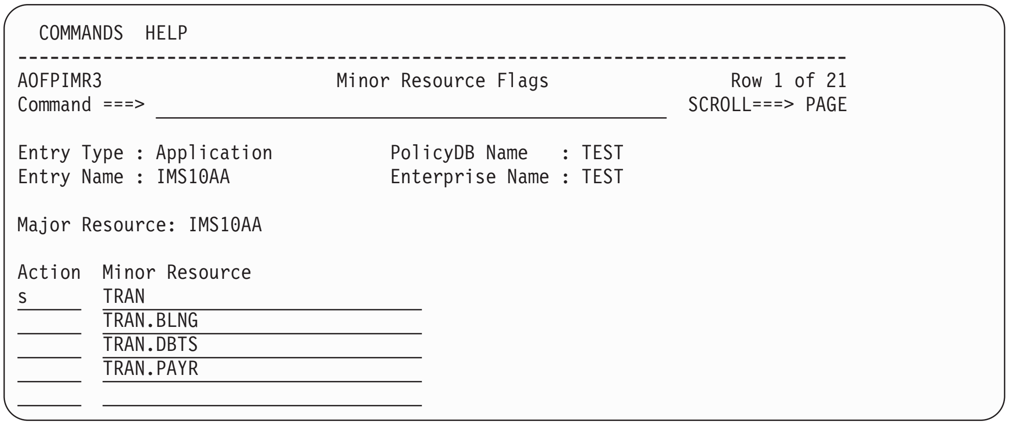
- Determining which application program (TP) transactions will have recovery automation
- Identifying the batch message region (BMP) transactions that will have recovery automation
- Specifying the error threshold level at which a recovery should stop
- Identifying specific abend codes for which you 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 AT 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

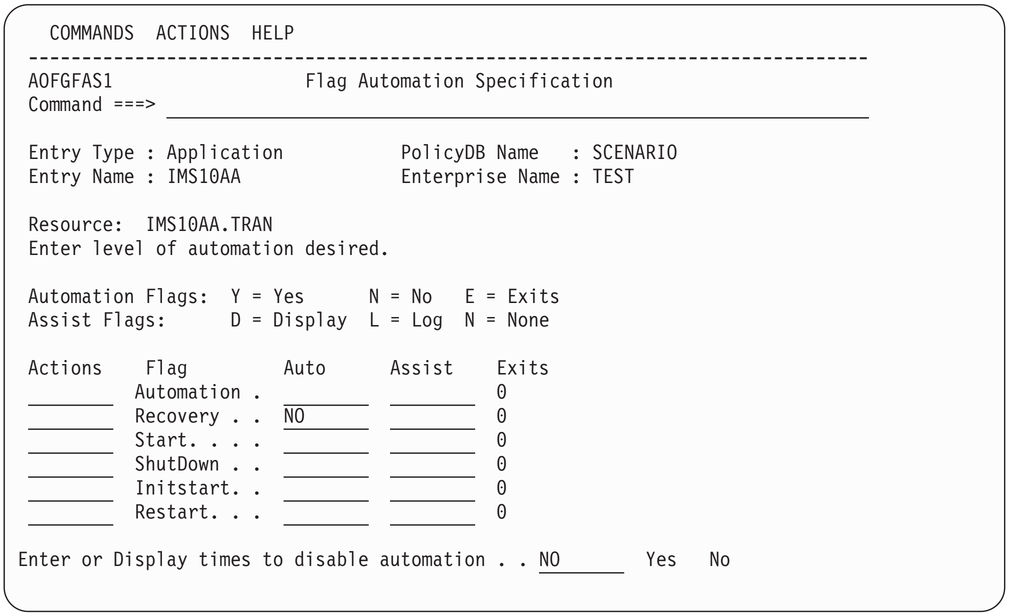
Suppose that recovery is enabled for the IMS10AA application on the application level, and that you want it also enabled for transactions PAYR, DBTS, and BLNG, but not for any other transaction. Then you must define four minor resources for IMS10AA in the customization dialog as follows:



```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFPIMR3                      Minor Resource Flags                Row 1 of 21
Command ===> ________________________________________________ SCROLL===> PAGE

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

Major Resource: IMS10AA

Action  Minor Resource
s_____  TRAN______________________________
______  TRAN.BLNG_________________________
______  TRAN.DBTS_________________________
______  TRAN.PAYR_________________________
______  __________________________________
```

Figure 12. 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 **Action** column and press ENTER. The following panel is displayed:



```
  COMMANDS  ACTIONS  HELP
------------------------------------------------------------------------------
AOFGFAS1                 Flag Automation Specification
Command ===> __________________________________________________________________

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

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

Enter or Display times to disable automation . . NO______  Yes   No
```

Figure 13. Automation Flag Panel

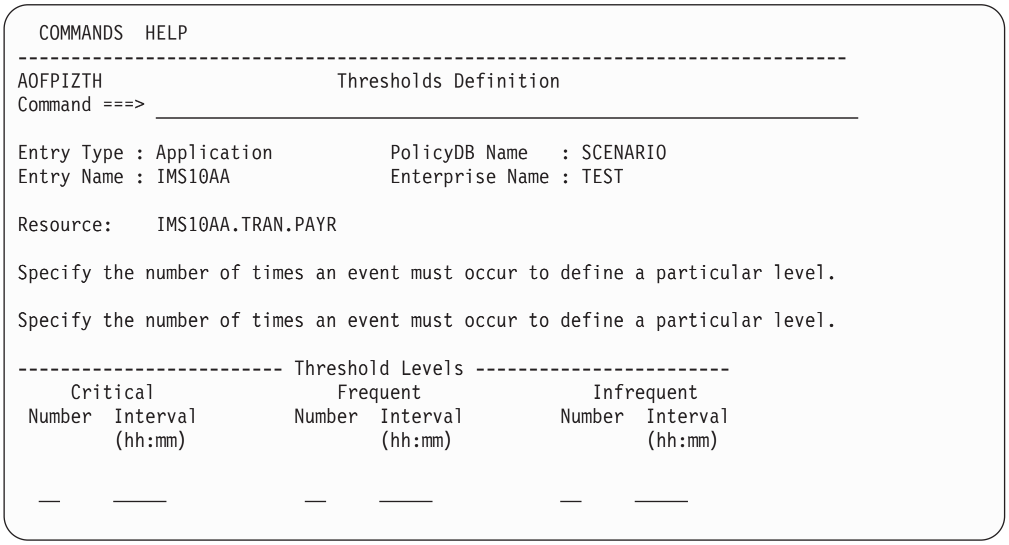
Here you specify which flags are set and which are not. For more information, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

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
------------------------------------------------------------------------------
AOFPIZTH                      Thresholds Definition
Command ===> __________________________________________________________________

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

Resource:    IMS10AA.TRAN.PAYR

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

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

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


  __     _____             __     _____             __     _____
```

Figure 14. Thresholds Definitions Panel

For more details, see *IBM Tivoli System Automation for z/OS Defining Automation Policy*.

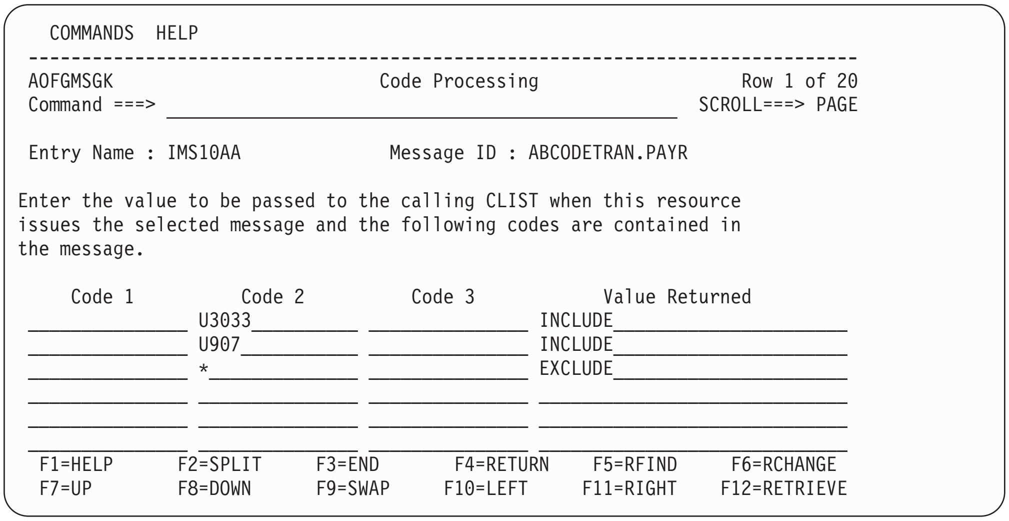
For recovery thresholds of programs, you must use PROG instead of TRAN as the first part of the thresholds' name.

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 need to create an ABCODETRAN.PAYR entry in the MESSAGES/USER DATA policy item and use action "code" on it. This will take you to the ***Code Processing*** panel shown in Figure 15 on page 44.



```
  COMMANDS  HELP
------------------------------------------------------------------------------
AOFGMSGK                         Code Processing                   Row 1 of 20
Command ===> ________________________________________________  SCROLL===> PAGE

Entry Name : 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
```

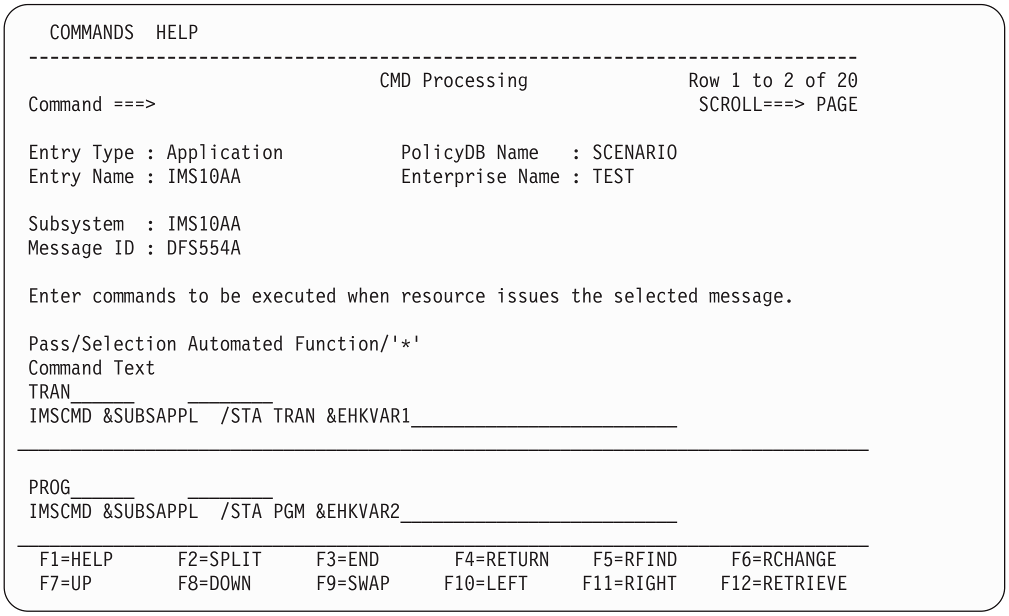
*Figure 15. **Code Processing** Panel*

For more details, see “ABCODETRAN—Transaction Abend Recovery” on page 51.

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 16. Command Processing Panel

For more details, see “DFS554A—Respond to Program Abend” on page 62.

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 14.

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 39.
- 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, to be found in the PRD 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 or state changes are performed by the common state handler routine. This routine is typically invoked from the AT. 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 41.

In addition to 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 100. 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 z/OS, you must enter any information for IMS Automation in the policy database via the customization dialog. In most cases, the customization dialog itself restricts you to 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 policy item of the application policy object in point; for the MESSAGES/USER DATA policy item, see *IBM Tivoli System Automation for z/OS 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. However, a general understanding of the MESSAGES/USER DATA policy item is assumed.

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. They need to be entered in the MESSAGE ID field on the ***Message Processing*** panel. 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 keyword 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 keyword. Transaction driven BMPs use the ABCODETRAN keyword to determine recovery actions.

Use action `code` on the ***Message Processing*** panel.

*Table 11. Data to Be Entered in the **Code Processing** Panel for ABCODEPROG*

Code 1	Code 2	Code 3	Value Returned
*	*acode*	*progid*	`INCLUDE` or `EXCLUDE`

Note: The ABCODEPROG keyword may be followed with a dot and a program ID *progid*.

Keyword and Parameter Definitions

ABCODEPROG[.*progid*]
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.

ABCODETRAN—Transaction Abend Recovery

Use this keyword 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.

Use action code on the *Message Processing* panel.

*Table 12. Data to Be Entered in the **Code Processing** Panel for ABCODETRAN*

Code 1	Code 2	Code 3	Value Returned
tran	*abend*	*pgm*	INCLUDE or EXCLUDE

Note: The ABCODETRAN keyword may be followed with a dot and a a transaction ID *tran*.

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.

abend
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.

BRO—Broadcast a Message Prior to Shutdown

This keyword is used to define 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.

Use action `rep` on the ***Message Processing*** panel.

*Table 13. Data to Be Entered in the **Reply Processing** Panel for BRO*

Pass/Selection	Retry Count	Reply Text
—	*n*	`/BRO ACTIVE`
—	*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.

CHE—Issue a Checkpoint Command

This keyword is used to specify the IMS command to be issued when a normal checkpoint is required.

- For IMS control regions, use action `rep` on the ***Message Processing*** panel and fill in the data in Table 14.
- For DB2 control regions, use action `cmd` on the ***Message Processing*** panel and fill in the data in Table 15.

*Table 14. Data to Be Entered in the **Reply Processing** Panel for CHE*

Pass/Selection	Retry Count	Reply Text
—	*n*	`/CHE`

*Table 15. Data to Be Entered in the **CMD Processing** Panel for CHE*

Pass/Selection	Automated Function/'*'	Command Text
—	—	`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 keyword is used to define the reply required in response to the WTOR message CQS0031A during CQS startup.

Use action `rep` on the ***Message Processing*** panel.

*Table 16. Data to Be Entered in the **Reply Processing** Panel for CQS0031A*

Pass/Selection	Retry Count	Reply Text
—	—	`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 keyword is used to define the reply required in response to the WTOR message CQS0032A during CQS startup.

Use action `rep` on the ***Message Processing*** panel.

Table 17. Data to Be Entered in the ***Reply Processing*** *Panel for CQS0032A*

Pass/Selection	Retry Count	Reply Text
—	—	`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 keyword is used to define the reply required in response to the WTOR message CQS0033A during CQS startup.

Use action `rep` on the ***Message Processing*** panel.

Table 18. Data to Be Entered in the ***Reply Processing*** *Panel for CQS0033A*

Pass/Selection	Retry Count	Reply Text
—	—	`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 keyword is used to define the /CQSET command that will be issued during IMS startup to cause a Structure Checkpoint at CQS shutdown.

Use action `cmd` on the ***Message Processing*** panel.

Table 19. Data to Be Entered in the ***CMD Processing*** *Panel for CQSET*

Pass/Selection	Automated Function/'*'	Command Text
—	—	`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 when the IMS control region issues message `DFS994I` *rtype* `START COMPLETED`. 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 keyword is used to define the commands required for response to the DFS2142 message following notification that a logical link path was stopped.

Use action rep on the *Message Processing* panel.

Table 20. Data to Be Entered in the ***Reply Processing*** *Panel for DFS2142*

	Pass/Selection	Retry Count	Reply Text
required	START	—	/START MSNAME &EHKVAR2
optional	RESTART	—	/PSTOP LINK &EHKVAR1
optional	RESTART	—	/START MSNAME &EHKVAR2
optional	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 keyword is used to define the command required (for response to the DFS2161I message) following the stopping of a link by the other system attached to it.

Use action `rep` on the ***Message Processing*** panel.

*Table 21. Data to Be Entered in the **Reply Processing** Panel for DFS2161I*

Pass/Selection	Retry Count	Reply Text
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 keyword is used to define 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.

Use action `rep` on the ***Message Processing*** panel.

*Table 22. Data to Be Entered in the **Reply Processing** Panel for DFS2169I*

Pass/Selection	Retry Count	Reply Text
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 keyword is used to define the commands required (for response to the DFS3258A message) following the use of either the last OLDS or all OLDS.

Use action `cmd` on the ***Message Processing*** panel.

*Table 23. Data to Be Entered in the **CMD Processing** Panel for DFS3258A*

Pass/Selection	Automated Function/'*'	Command Text
SYSTEM	—	*command_text_1*
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.

DFS554A—Respond to Program Abend

This keyword is used to define the commands required for response to the DFS554A message, following notification of a program abend.

Use action `rep` on the ***Message Processing*** panel with the data in Table 24 or action `cmd` with the data in Table 25.

*Table 24. Data to Be Entered in the **Reply Processing** Panel for DFS554A*

Pass/Selection	Retry Count	Reply Text
TRAN	—	*command*
PROG	—	*command*

*Table 25. Data to Be Entered in the **CMD Processing** Panel for DFS554A*

Pass/Selection	Automated Function/'*'	Command Text
TRAN	—	*command_text*
PROG	—	*command_text*

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 action `cmd` only. Note that transactions are not handled by DB control regions so that you need not code the row with `TRAN` in Table 25 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.

DFS810A—Define Restart Commands

This keyword is used to define the restart commands required (for response to the DFS810A message) following successful completion of IMS initialization.

Use action `rep` on the ***Message Processing*** panel.

Table 26. Data to Be Entered in the ***Reply Processing*** *Panel for DFS810A*

Pass/Selection	Retry Count	Reply Text
`COLD`	*n*	*command*
`BUILDQ`	*n*	*command*
`WARMSDBL`	*n*	*command*
user	*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.

user
Any user-defined start type may be specified.

When the operator selects a start type of ″user″ on the INGREQ panel, this parameter is required to define the restart command that is 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.

DFS989I—Define Restart Commands (DBCTL Only)

This keyword is used to define the restart commands required (for response to the DFS989I message) following successful completion of IMS initialization.

Use action `cmd` on the ***Message Processing*** panel.

*Table 27. Data to Be Entered in the **CMD Processing** Panel for DFS989I*

Pass/Selection	Automated Function/'*'	Command Text
`WARMSDBL`	`MVS &EHKVAR7`	*command_text*
`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.

DFS994I—Respond to Checkpoint Written to the IMS Log

This keyword is used to define the commands required for response to the DFS994I “*xxxx* START COMPLETED” message which follows the writing of a checkpoint to the IMS system log.

- For IMS control regions, use both action `rep` on the ***Message Processing*** panel with the data in Table 28 and action `cmd` with the data in Table 29.

Table 28. Data to Be Entered in the ***Reply Processing*** *Panel for DFS994I with IMS control regions*

Pass/Selection	Retry Count	Reply Text
ERE	*n*	*command*
COLD	*n*	*command*
BUILDQ	*n*	*command*
WARMSDBL	*n*	/STA DC
WARMSDBL	*n*	/CHE SNAPQ
MANUAL	*n*	*command*
WARM	*n*	/STA DC
WARM	*n*	/CHE SNAPQ
user	*n*	*command*

Table 29. Data to Be Entered in the ***CMD Processing*** *Panel for DFS994I with IMS control regions*

Pass/Selection	Automated Function/’*’	Command Text
WARM	—	*command*

- For DB2 control regions, use action `cmd` on the ***Message Processing*** panel with the data in Table 30.

Table 30. Data to Be Entered in the ***CMD Processing*** *Panel for DFS994I with DB2 control regions*

Pass/Selection	Automated Function/’*’	Command Text
COLD	—	*command*
ERE	—	*command*
MANUAL	—	*command*
WARM	—	*command*
user	—	*command*

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.

user

Any user-defined start type may be specified.

When the operator selects a start type of ″user″ on the INGREQ panel, this parameter is required to define the restart command that is to be issued in response to message DFS994I, 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 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.

HOLDQ—Issue Commands at Shutdown

This keyword is used to define the commands that may be issued at the initiation of the shutdown process and subsequent to PRECHKP.

Use action `cmd` on the ***Message Processing*** panel.

*Table 31. Data to Be Entered in the **CMD Processing** Panel for HOLDQ*

Pass/Selection	Automated Function/'*'	Command Text
—	—	*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.

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 *IBM Tivoli System Automation for z/OS Operator's Commands*.

Use action `cmd` on the ***Message Processing*** panel.

Table 32. Data to Be Entered in the ***CMD Processing*** *Panel for IMSINFO*

Pass/Selection	Automated Function/'*'	Command Text
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.

You may code multiple IMS commands, separated by a comma, in order to group results under a common description.

Comments and Usage Notes

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

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 70 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.

Use action `user` on the ***Message Processing*** panel.

*Table 33. Data to Be Entered in the **User Defined Data** Panel for OLDS*

Keyword	Data
`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.

The data portion of this entry can be replaced by ″DYNAM″ and SA z/OS will attempt to discover any spares data sets that have been made available to IMS Automation. This requires that the spares are defined dynamically to IMS Automation (refer to the IMS DFSMDA macro). When IMS Automation is cold started, the spares are discovered by SA z/OS and saved in the save/restore cache. When OLDS are required, the result of a /DIS OLDS is compared to the saved globals and treated as normal if found and spare if not.

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.

POSTCHKP—Issue Commands after IMS Shutdown Checkpoint

This keyword is used to define the commands that may be issued immediately after the IMS shutdown checkpoint has been issued.

Use both action rep on the ***Message Processing*** panel with the data in Table 34 and action cmd with the data in Table 35.

Table 34. Data to Be Entered in the ***Reply Processing*** *Panel for POSTCHKP*

Pass/Selection	Retry Count	Reply Text
—	*n*	*ims_command*
—	*n*	*ims_command*

Table 35. Data to Be Entered in the ***CMD Processing*** *Panel for POSTCHKP*

Pass/Selection	Automated Function/'*'	Command Text
—	—	*command_text*
—	—	*command_text*

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.

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.

Use both action `rep` on the ***Message Processing*** panel with the data in Table 36 and action `cmd` with the data in Table 37.

*Table 36. Data to Be Entered in the **Reply Processing** Panel for PRECHKP*

Pass/Selection	Retry Count	Reply Text
—	*n*	*ims_command*
—	*n*	*ims_command*

*Table 37. Data to Be Entered in the **CMD Processing** Panel for PRECHKP*

Pass/Selection	Automated Function/'*'	Command Text
—	—	*command*
—	—	*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.

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.

Use action `user` on the ***Message Processing*** panel.

*Table 38. Data to Be Entered in the **User Defined Data** Panel for RECONS*

	Keyword	Data
required	`MONITOR`	*hh:mm:ss*
optional	`RETRY`	*nn*
optional	`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. If you specify 24 for hours, specify 00 for minutes and seconds. A time period of zero cannot be specified.

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 AT.
   - 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.
4. It is possible to define RECONS as a minor resource and control RECONS automation by setting the minor resource flags as required.

RELEASEQ—Issue Commands after Shutdown Completes

This keyword is used to define the commands that may be issued after IMS shutdown completion.

Use action `cmd` on the ***Message Processing*** panel.

*Table 39. Data to Be Entered in the **CMD Processing** Panel for RELEASEQ*

Pass/Selection	Automated Function/'*'	Command Text
—	—	*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 67), that is, to release the job classes held during the shutdown process.

RESTARTABORT—Emergency Restart Commands

This keyword is used to define the emergency restart commands required (for response to the DFS166 and DFS0618A messages) following an abort termination of the restart process.

To create the definition, select the MESSAGES/USER DATA policy item to get to the ***Message Processing*** panel. Then enter the RESTARTABORT keyword in the MESSAGE ID field.

- For IMS control regions, use action `rep` on the ***Message Processing*** panel and fill in the data in Table 40.
- For DB2 control regions, use action `cmd` on the ***Message Processing*** panel and fill in the data in Table 41.

Table 40. Data to Be Entered in the ***Reply Processing*** *Panel for RESTARTABORT*

Pass/Selection	Retry Count	Reply Text
`OVERRIDE`	*n*	*command*
`BACKUP`	*n*	*command*
`ERE`	*n*	*command*

Table 41. Data to Be Entered in the ***CMD Processing*** *Panel for RESTARTABORT*

Pass/Selection	Automated Function/′*′	Command Text
`OVERRIDE`	—	*command*
`ERE`	—	*command*

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.

SHUTTYPES—Issue Commands at Operator Shutdown

This keyword is used to specify 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 117.

Use both action `rep` on the ***Message Processing*** panel with the data in Table 42 and action `cmd` with the data in Table 43.

*Table 42. Data to Be Entered in the **Reply Processing** Panel for SHUTTYPES*

Pass/Selection	Retry Count	Reply Text
shutoption	*n*	*command*

*Table 43. Data to Be Entered in the **CMD Processing** Panel for SHUTTYPES*

Pass/Selection	Automated Function/'*'	Command Text
`NODUMP`	—	*command_text*
`DUMP`	—	*command_text*
`PURGE`	—	*command_text*
`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 options DUMPQ, BACKUP, FREEZE or PURGE), and two CMD attributes, one with DUMP and the other with NODUMP as its first value. DUMPQ and BACKUP are not applicable to DB control regions.

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.

When you do a NORM or IMMED shutdown, the only options allowed are:

- For a DBCTL region: PURGE, FREEZE
- For an XRF CTL region in BACKUP mode: BACKUP
- For a CTL region: PURGE, FREEZE or DUMPQ

When you do a FORCE the only options allowed are DUMP or NODUMP.

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 119.

SNAPQ—Issue SNAPQ Checkpoint Command

This keyword is used to specify the IMS command to be issued when a SNAPQ checkpoint is required.

Use action `rep` on the ***Message Processing*** panel.

*Table 44. Data to Be Entered in the **Reply Processing** Panel for DFS2169I*

Pass/Selection	Retry Count	Reply Text
—	*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).

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 36.

- For IMS control regions, use action `rep` on the ***Message Processing*** panel and fill in the data in Table 45.
- For DB2 control regions, use action `cmd` on the ***Message Processing*** panel and fill in the data in Table 46.

*Table 45. Data to Be Entered in the **Reply Processing** Panel for STOPBMPREGION*

	Pass/Selection	Retry Count	Reply Text
required	`NORMAL`	*n*	`/STOP REGION &EHKVAR1`
required	`ABEND`	*n*	`/STOP REGION &EHKVAR1 ABDUMP`
optional	`CANCEL`	*n*	`/STOP REGION &EHKVAR1 ABDUMP`
required	`CANCEL`	*n*	`/STOP REGION &EHKVAR1 CANCEL`

*Table 46. Data to Be Entered in the **CMD Processing** Panel for STOPBMPREGION*

	Pass/Selection	Automated Function/'*'	Command Text
required	`NORMAL`	—	*command_text*
required	`ABEND`	—	*command_text*
at least once	`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.

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 36.

Use action `rep` on the ***Message Processing*** panel.

*Table 47. Data to Be Entered in the **Reply Processing** Panel for STOPFPREGION*

	Pass/Selection	Retry Count	Reply Text
required	`NORMAL`	*n*	`/STOP REGION &EHKVAR1`
required	`ABEND`	*n*	`/STOP REGION &EHKVAR1 ABDUMP`
optional	`CANCEL`	*n*	`/STOP REGION &EHKVAR1 ABDUMP`
required	`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 REP attribute 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` *xxx*`x 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 36.

Use action `rep` on the ***Message Processing*** panel.

*Table 48. Data to Be Entered in the **Reply Processing** Panel for STOPREGION*

	Pass/Selection	Retry Count	Reply Text
required	`NORMAL`	*n*	`/STOP REG &EHKVAR1`
required	`ABEND`	*n*	`/STOP REG &EHKVAR1 ABDUMP &EHKVAR2`
optional	`CANCEL`	*n*	`/STOP REG &EHKVAR1 ABDUMP &EHKVAR2`
required	`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.

Use action `rep` on the ***Message Processing*** panel.

*Table 49. Data to Be Entered in the **Reply Processing** Panel for TCO*

Pass/Selection	Retry Count	Reply Text
INIT	—	DFSTCF LOAD DFSTCF .
SPEC	—	DFSTCF LOAD &EHKVAR1 .
START	—	/START LTERM DFSTCF1 .
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.

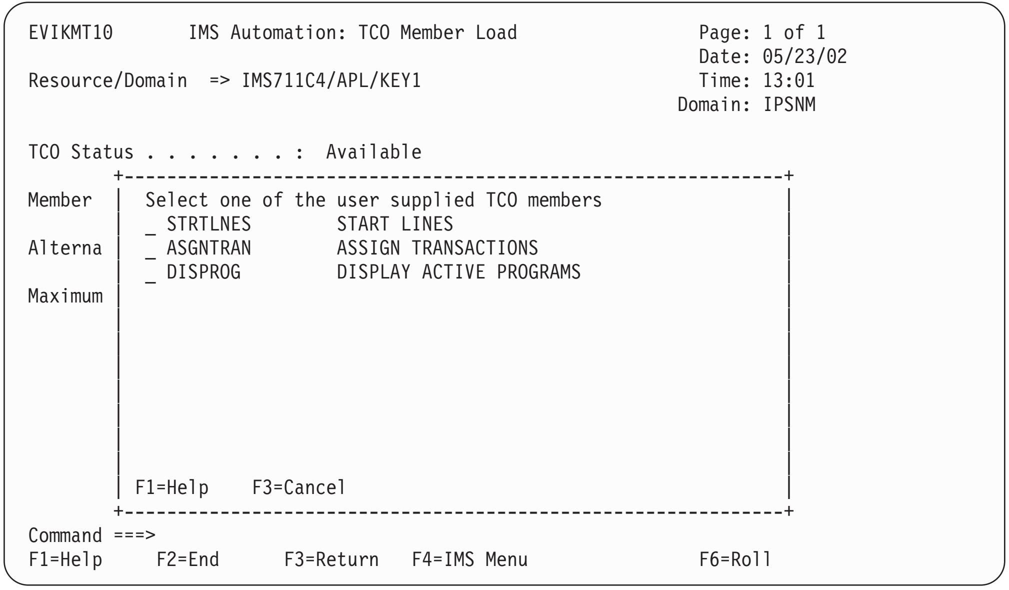
STOP
This entry specifies the IMS command required to stop TCO.

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
```

Use action `user` on the ***Message Processing*** panel.

*Table 50. Data to Be Entered in the **User Defined Data** Panel for TCOMEMBERS*

Keyword	Data
NAME	(*membername*,'*comment*')

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 123.

TPABEND—Dependent Region Abend

An entry with this keyword is required for a dependent region abend with a U0002 return code.

Use action `code` on the ***Message Processing*** panel.

*Table 51. Data to Be Entered in the **Code Processing** Panel for TPABEND*

Code 1	Code 2	Code 3	Value Returned
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.

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:

BUILDQCKPT, DBRCRESRV, DCSTATUS, DEPREGID, EDNDT, LASTABENDCODE, GENAPPLID, RUNAPPLID, RUNSTARTYPE, STARTDT, STARTTYPE, VER.

For more information about the keywords, refer to common routine "EVIEX003—Update Status File Data" on page 92.

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

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:

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.

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.

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.

user
The IMS system will be started using user defined entries in the STARTUP policy item.

RUNAPPLID=
Specifies the specific applid. The value is taken from the IMSCNTL APPLID ACF entry when it is loaded.

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.

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.

user
The IMS system will be started using user defined entries in the STARTUP policy item.

VER=
Specifies the IMS product version and release level executing in the particular IMS subsystem. It has the format “version.release”.

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.

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.

```
►►──IMSBMSG──┬─START──,──domainid──,──taskid─┬──────────────────────►◄
             ├─DATA──,──linetype──,──data────┤
             ├─END───────────────────────────┤
             └─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

IMSCMD now acts as a stub for INGIMS for compatibility. Refer to *IBM Tivoli System Automation for z/OS Operator's Commands* for information on the INGIMS command.

```
►►──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

Refer to *IBM Tivoli System Automation for z/OS Operator's Commands* for comments and usage on the INGIMS command.

Note: The IMS command prefix character "/" is now optional for IMSCMD. Also the IMS subsystem can be represented by a fully defined SA resource (e.g. IMS811C4/APL/SYS1).

IMSFWM—Add Messages to SDF

This routine adds messages to SDF critical message panels.

```
►►──IMSFWM──┬─msgtext───────────────┬────────────────────────────────────►◄
            └─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.

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.

```
►►──IMSQRY──REQ=──┬─VALIDATE─┬──┬───────────────────────────┬──┬─────────────────────┬──────────►◄
                  └─GET──────┘  └─,─NAME=──┬─subsystem─────┬─┘  └─,─TYPE=──┬─IMS─────┬─┘
                                           ├─resource_name─┤               ├─GROUP───┤
                                           └─jobname───────┘               ├─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 z/OS

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 z/OS.

TYPE=
Used to provide a specific type. The types are:

IMS
Search for a specific IMS subsystem name, as it is known to SA z/OS. 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.

Note: For NetView 5.1 and above DSIDMNK entries have been moved to CNMSTYLE.

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
32	Unsupported function
36	Resource name is ambiguous (more than one resource of the same name exists within the sysplex but none are defined on the local system)
40	System name where the IMS resource resides is not unique within the enterprise
44	IMS resource tree contains 0 or more than 1 MOVE groups

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 z/OS 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.

```
►►──IMSRCMD──NAME=──subsystem──,──┬─────────────────────┬──┬────────────────────┬──CMD=──cmd──────►◄
                                  └─RESP=──┬─ACK─┬──,─┘  └─OPER=──operator──,─┘
                                           ├─YES─┤
                                           └─NO──┘
```

Keyword and Parameter Definitions

NAME=
The name by which the target IMS subsystem is known to SA z/OS, 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, AUTBASE 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

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 *IBM Tivoli System Automation for z/OS 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 z/OS 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 z/OS.

Using IMS Automation Panels

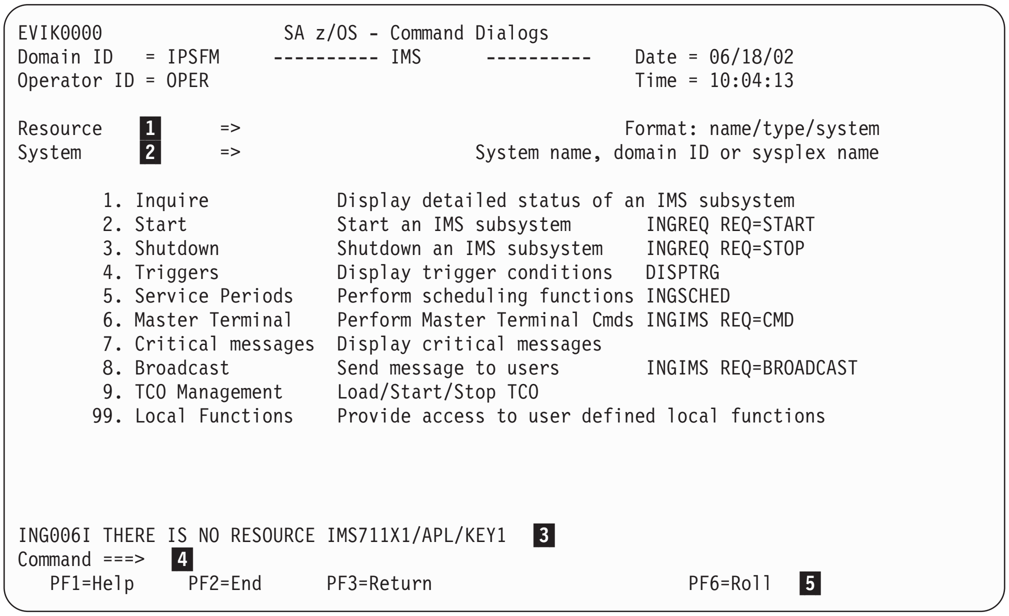
This section explains:

- 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 17.



```
EVIK0000                 SA z/OS - 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 17. 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 z/OS. 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.

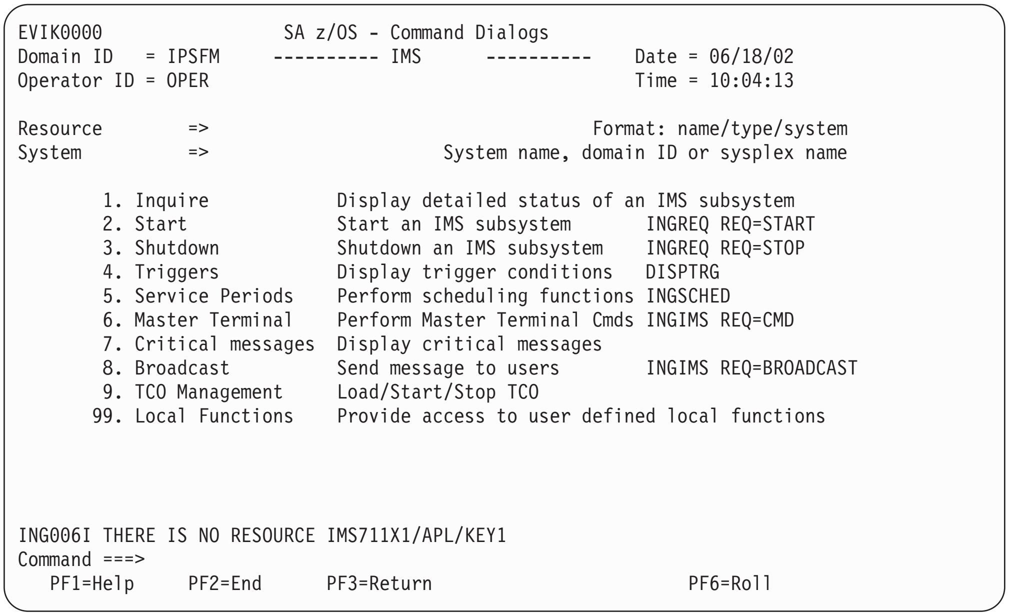
3 IMS Automation messages display here.

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 z/OS - 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 18. 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 111.

Start
Use this option to initiate the startup process of a resource. By choosing this option you call the INGREQ command of SA z/OS. See "Start" on page 117.

Shutdown
Use this option to initiate the shutdown process of a resource. By choosing this option you call the INGREQ command of SA z/OS. See Chapter 8, "Starting and Stopping Resources," on page 117.

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 z/OS. See *IBM Tivoli System Automation for z/OS 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 z/OS. See *IBM Tivoli System Automation for z/OS 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 *IBM Tivoli System Automation for z/OS 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 131.

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 z/OS. See *IBM Tivoli System Automation for z/OS 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 123 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.

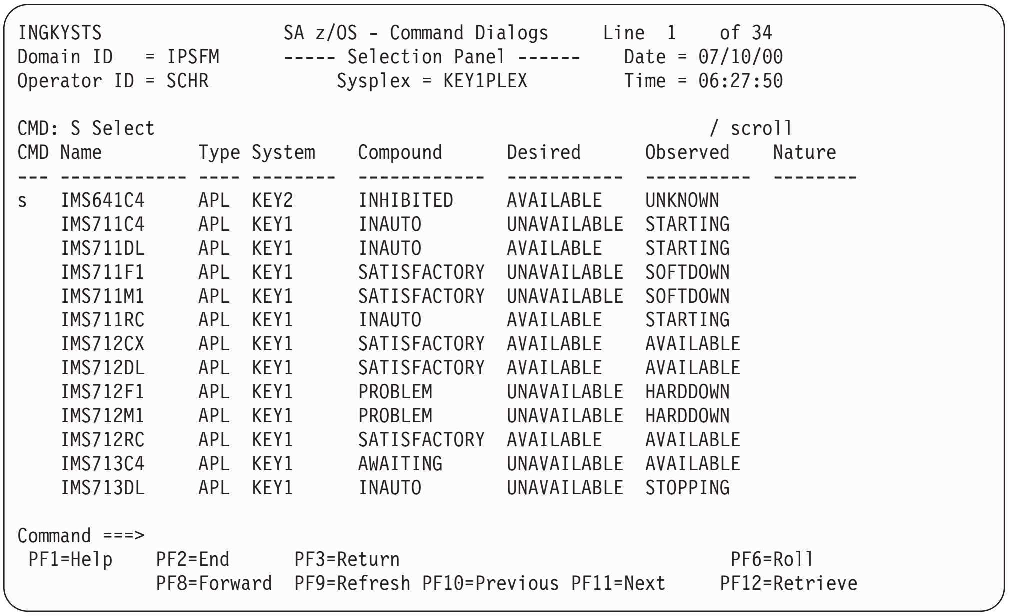
Selecting and Viewing Subsystems

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 z/OS commands (INGREQ, INGSCHED, DISPTRG), you can use an asterisk (*) as a wildcard.

Figure 19 on page 111 shows a list of IMS control regions generated with the question mark function.



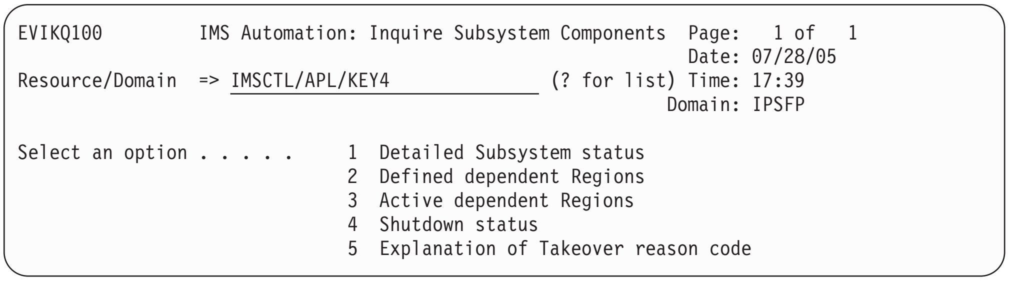
```
INGKYSTS                 SA z/OS - Command Dialogs     Line  1    of 34
Domain ID   = IPSFM      ----- Selection Panel ------    Date = 07/10/00
Operator ID = SCHR            Sysplex = KEY1PLEX         Time = 06:27:50

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 19. 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 z/OS. 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 5 through 7, for example, contain status information for the subsystems; for more details on the different status types, see *IBM Tivoli System Automation for z/OS 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: 07/28/05
Resource/Domain  => IMSCTL/APL/KEY4              (? for list) Time: 17:39
                                                             Domain: IPSFP

Select an option . . . . .     1  Detailed Subsystem status
                               2  Defined dependent Regions
                               3  Active dependent Regions
                               4  Shutdown status
                               5  Explanation of Takeover reason code
```

Figure 20. Inquire Subsystem Components Panel

From this screen, you can select the type of status display desired. The options are:

Detailed Subsystem status
Displays detailed status of the specified resource and domain. See Figure 21 on page 112 for an IMS control region example, or Figure 22 on page 113 for a DB control region example.

Defined dependent Regions
Displays the dependent regions as defined in the policy database for the specified subsystem and the status of each.

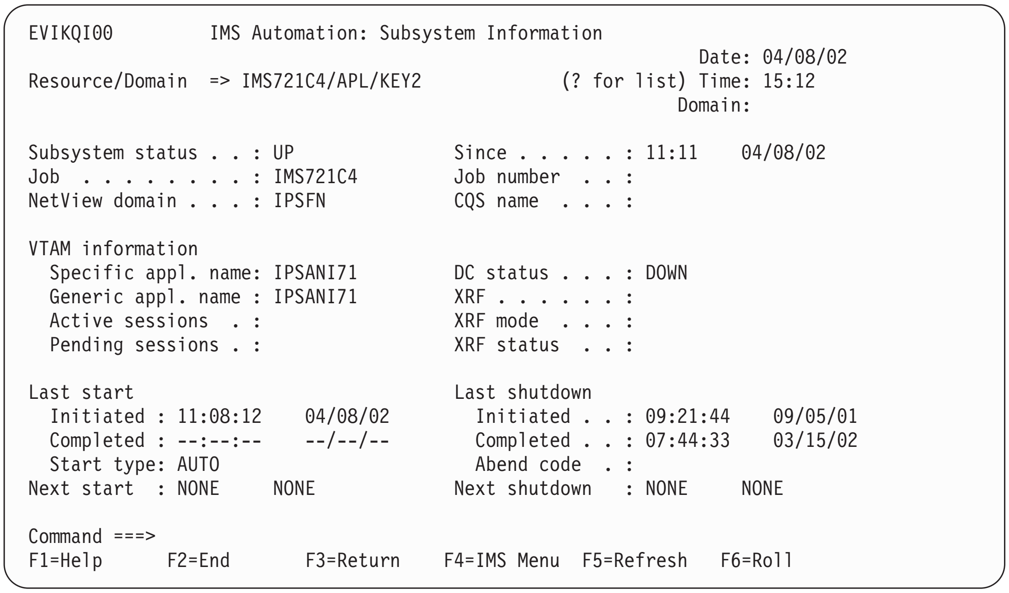
Active dependent Regions
Invokes the Subsystem/Active Regions Display panel, which lists the subsystem status, any active regions, jobnames, and tasknames.

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 status
Invokes the **Shutdown Status Display** panel, which lists the active elements, either nodes (terminals), lines, or links during shutdown processing.

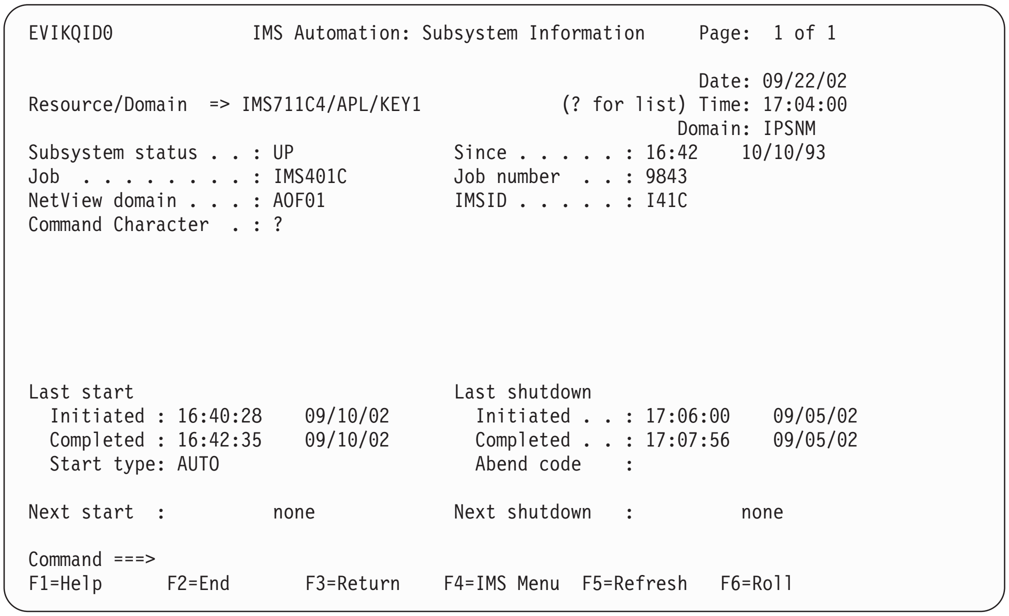
Explanation of Takeover reason code
Displays the reason code and explanation for a takeover.

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
```

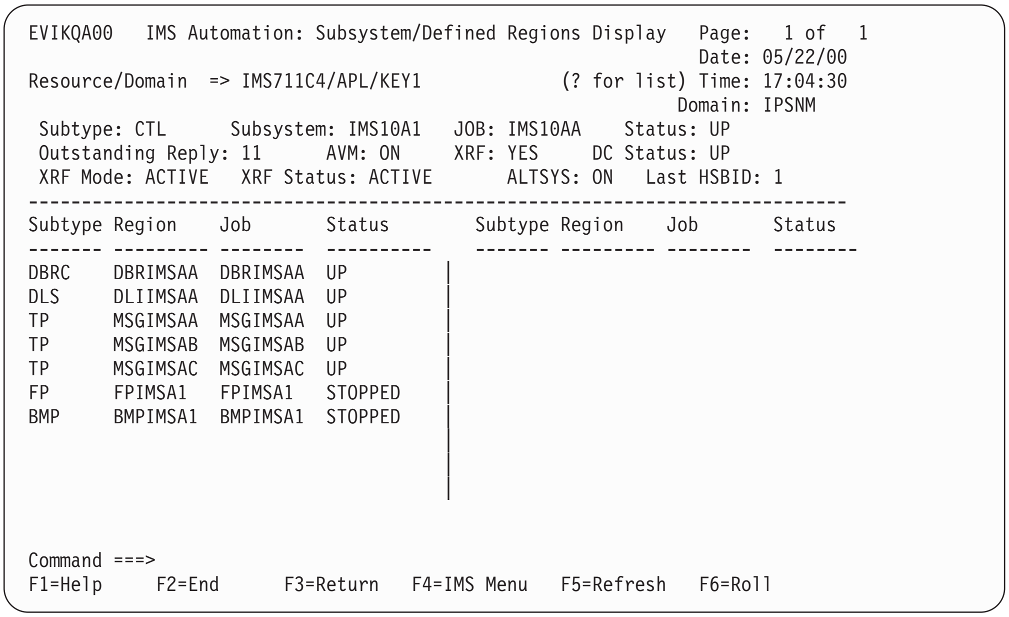
Figure 21. Detailed Subsystem Information Panel for an IMS Control Region. If the startup or shutdown occurred while NetView was up and running, the “Last start” and “Last shutdown” data will be accurate. If NetView was not running when the startup/shutdown occurred, the data may be inaccurate.



```
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 22. 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 23 shows the panel for an IMS control region, and Figure 24 on page 114 shows the panel for a DB control region.

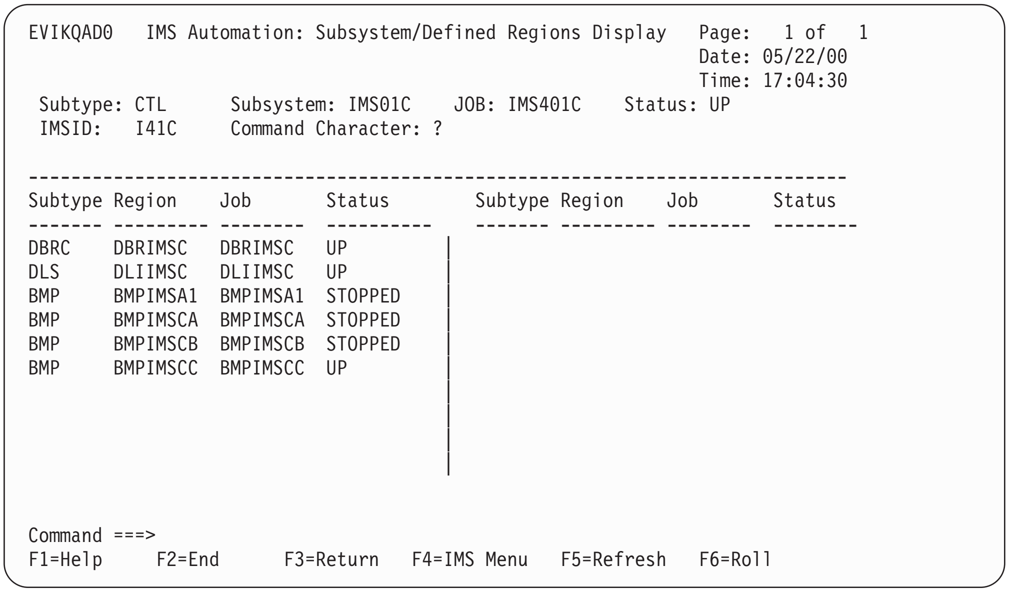


```
EVIKQA00   IMS Automation: Subsystem/Defined Regions Display   Page:   1 of   1
                                                               Date: 05/22/00
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 17:04:30
                                                             Domain: 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 23. Subsystem/Defined Regions Display Panel for an IMS Control Region

All defined regions display whether or not they are currently active. In Figure 23, the subsystem is IMS10A1 and the job is IMS10AA. For each region, the panel displays the subtype, region, job name, and status. Subtypes include:

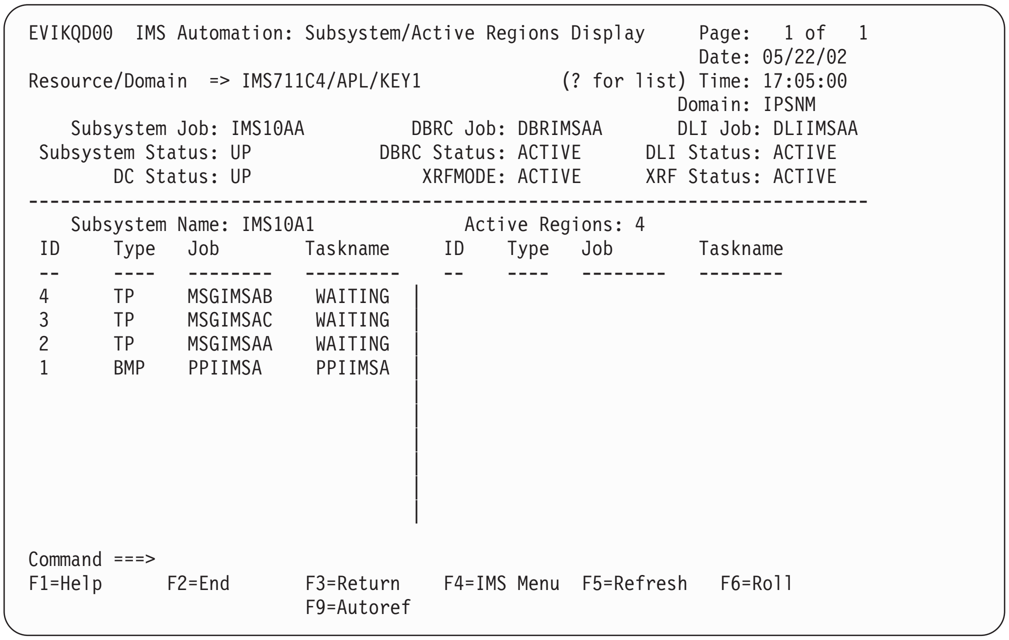
DBRC	Database recovery control region
DLS	Data language interface control region
TP	Normal online message region (transaction processing)
FP	Fast path message region
BMP	Batch message processing region



```
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
```

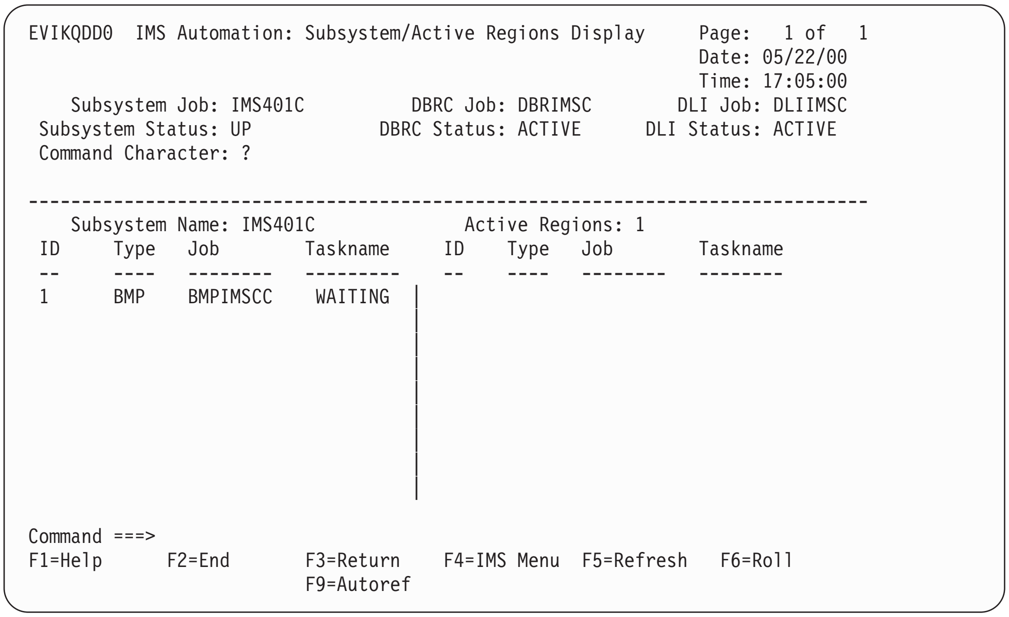
Figure 24. Subsystem/Defined Regions Display Panel for a DB Control Region

To display only the dependent regions that are currently active, select option 3 from the Inquire Subsystems Components panel. Figure 25 on page 115 shows the panel for an IMS control region, and Figure 26 on page 115 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 25. 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 26. Subsystem/Active Regions Display for a DB Control Region

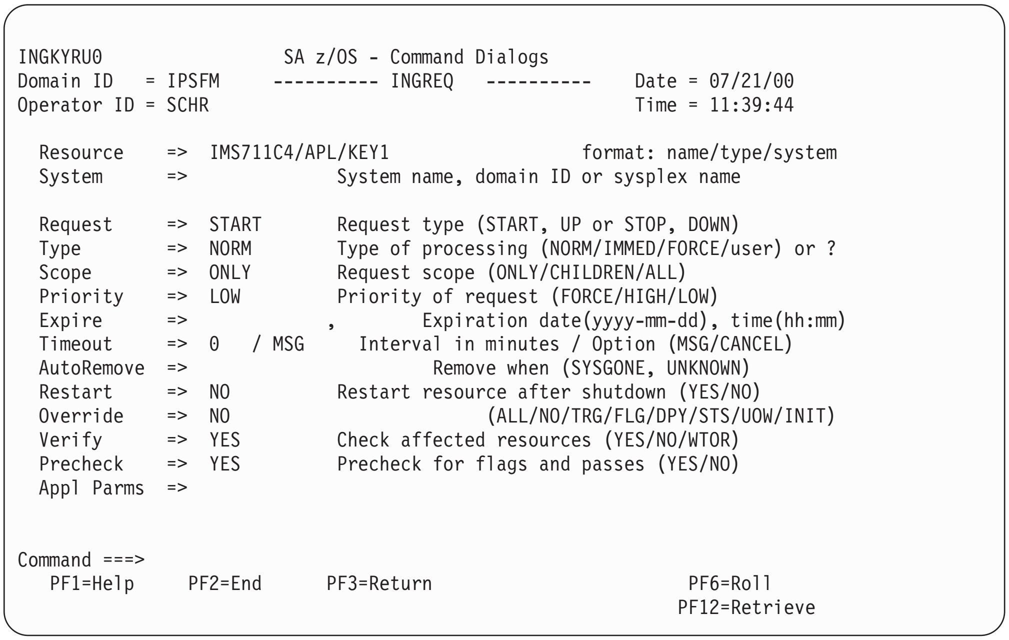
Chapter 8. Starting and Stopping Resources

IMS Automation uses the INGREQ command of SA z/OS for starting and stopping resources. For information on INGREQ, see *IBM Tivoli System Automation for z/OS 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 110.

Start

When you select option 2, Start, from the main menu panel, the INGREQ command dialog of SA z/OS is displayed.



```
INGKYRU0                 SA z/OS - 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 27. Input Panel for the INGREQ Command

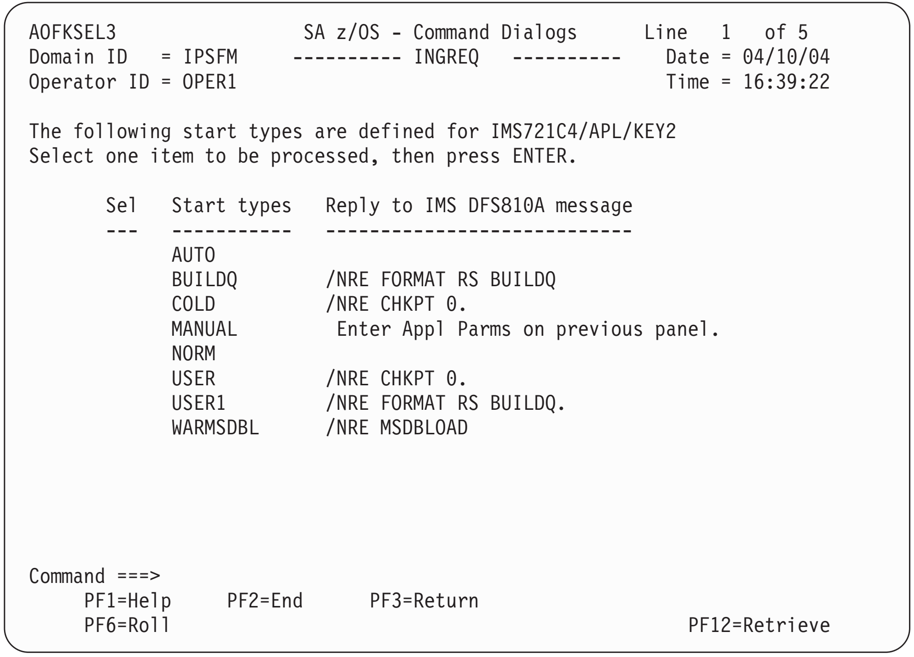
The IMS-specific features concern the **Type** and the **Appl Parms** fields:

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.

Startup Type	Explanation
MANUAL	Uses the data stored in the Apply Parms field of the INGREQ command to determine the IMS startup type.
User	Uses user-defined entries in the STARTUP policy item to determine the startup type.

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 z/OS - Command Dialogs      Line   1   of 5
Domain ID   = IPSFM     ---------- INGREQ   ----------    Date = 04/10/04
Operator ID = OPER1                                       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   Reply to IMS DFS810A message
       ---   -----------   ----------------------------
             AUTO
             BUILDQ        /NRE FORMAT RS BUILDQ
             COLD          /NRE CHKPT 0.
             MANUAL         Enter Appl Parms on previous panel.
             NORM
             USER          /NRE CHKPT 0.
             USER1         /NRE FORMAT RS BUILDQ.
             WARMSDBL      /NRE MSDBLOAD





Command ===>
     PF1=Help     PF2=End      PF3=Return
     PF6=Roll                                        PF12=Retrieve
```

Enter s in the **Sel** column to select the desired type.

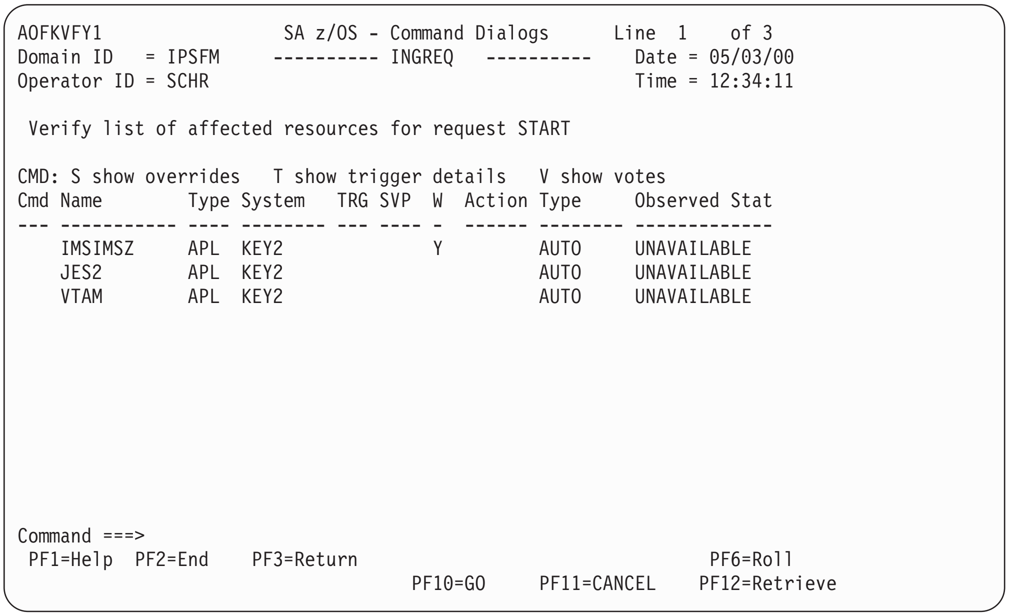
Note: When you select a startup type that is valid for IMS, but has not been defined in the STARTUP policy item of the target resource, INGREQ issues the command defined for the NORM startup type in the STARTUP item. If that entry does not exist either, the command MVS START *jobname* is issued.

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 28 on page 119) after you have pressed ENTER. This panel displays the target resource and in addition all the resources which SA z/OS will try to start because the startability of the selected resource directly or indirectly depends on them.



```
AOFKVFY1                 SA z/OS - 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
```

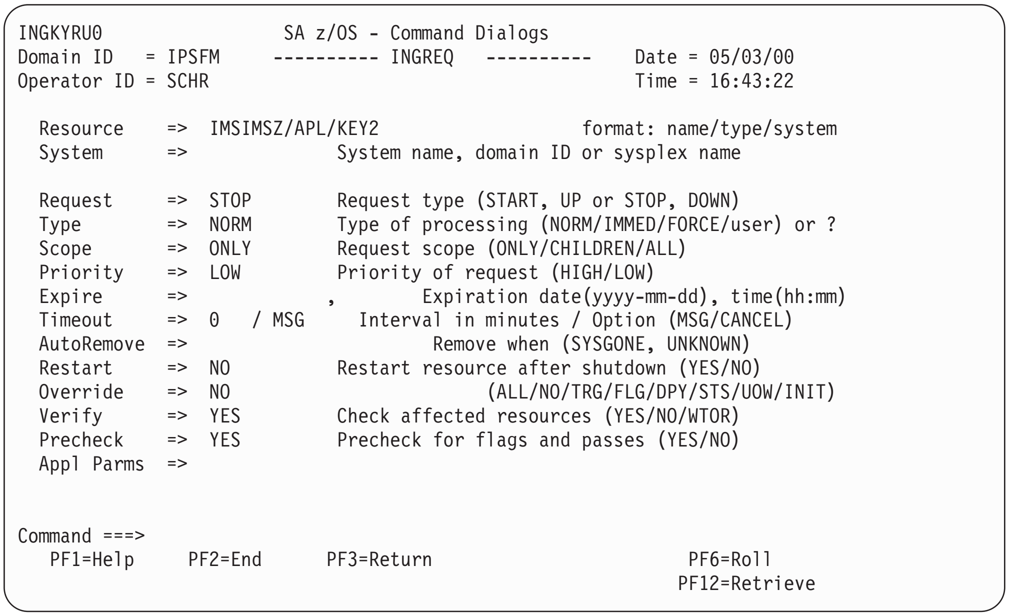
Figure 28. Verification Panel for INGREQ

For more information on the verification panel of INGREQ, see *IBM Tivoli System Automation for z/OS Operator's Commands*.

Note: For starting message regions, use the INGLIST command of SA z/OS.

Shutdown

When you select option 3, Shutdown, from the main menu panel, the INGREQ command dialog of SA z/OS is displayed.



```
INGKYRU0                SA z/OS - 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 29. 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 77), 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 34). 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}
```

When you do a NORM or IMMED shutdown, the only options allowed are:

- For a DBCTL region: PURGE, FREEZE
- For an XRF CTL region in BACKUP mode: BACKUP
- For a CTL region: PURGE, FREEZE or DUMPQ

When you do a FORCE the only options allowed are DUMP or 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

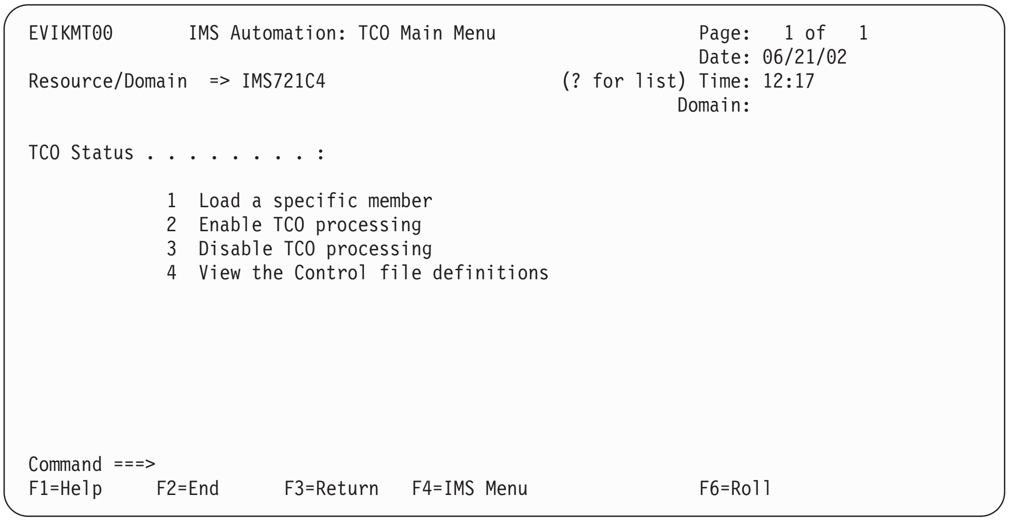
```
BROADCAST={YES|NO|CTL}
```

NO is the default.

Note: For stopping message regions, use the INGLIST command of SA z/OS.

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 30. TCO Main Menu

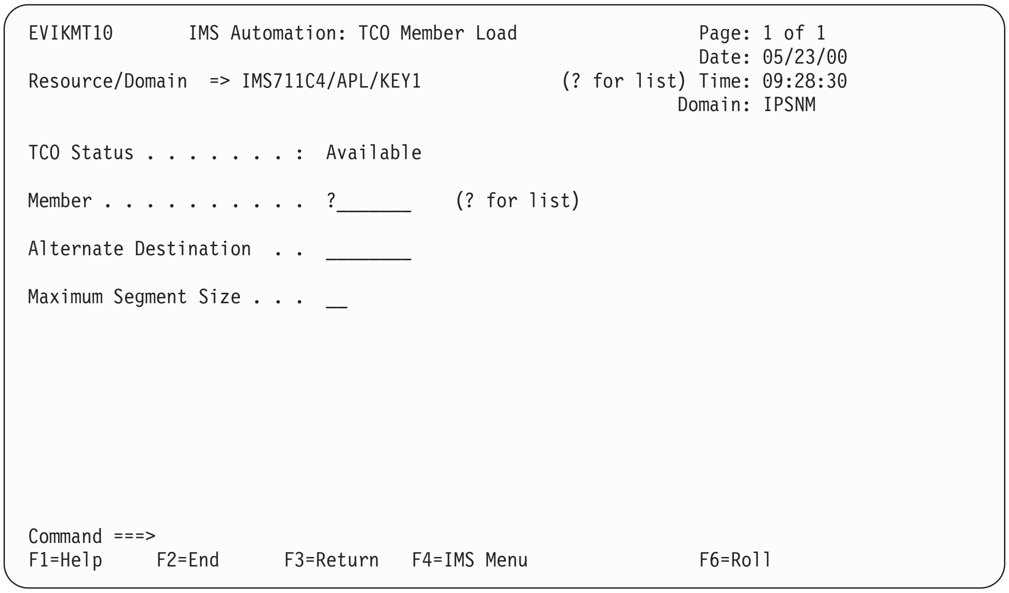
The Timer-Controlled Operations (TCO) interface lets you issue TCO commands from the IMS Automation interface.

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

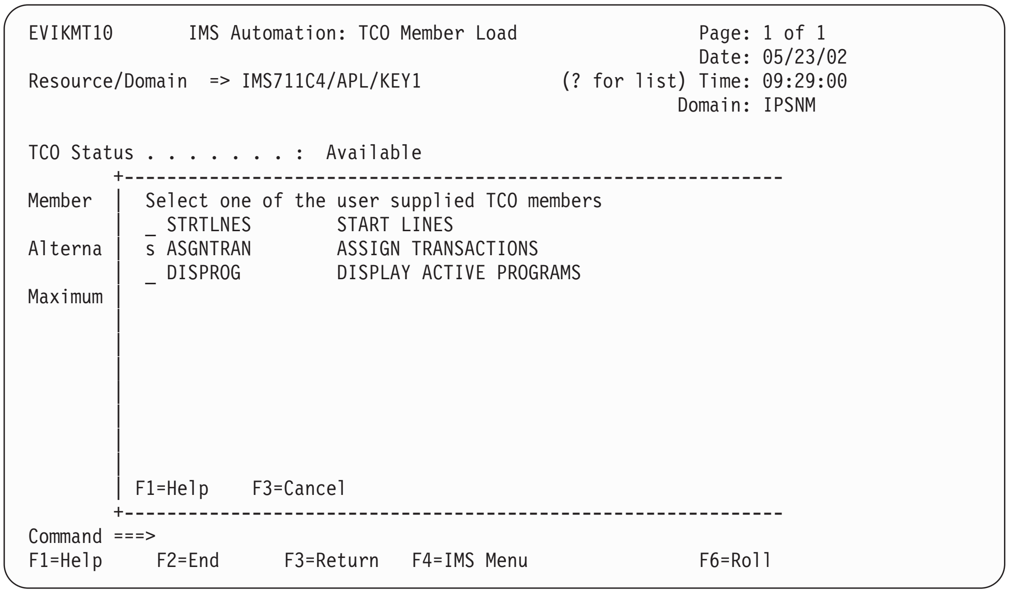
Select Option 1 from the TCO Main Menu to load a specific member. Panel EVIKMT10 is displayed. Enter a ‘?’ to see a member list:



```
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 31. TCO Member Load Panel

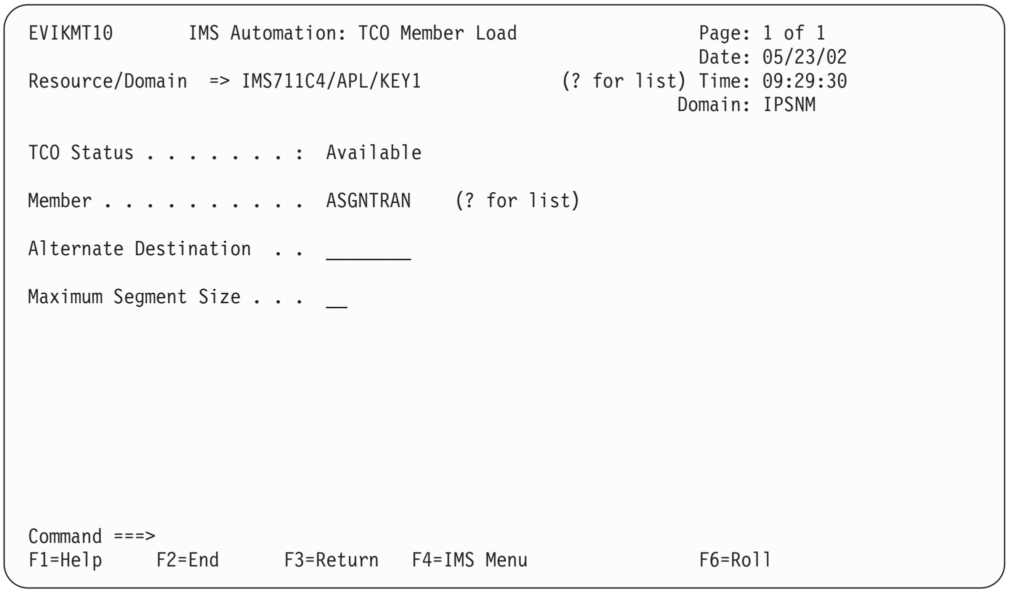
Select a member:



```
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 32. TCO Member Load Panel

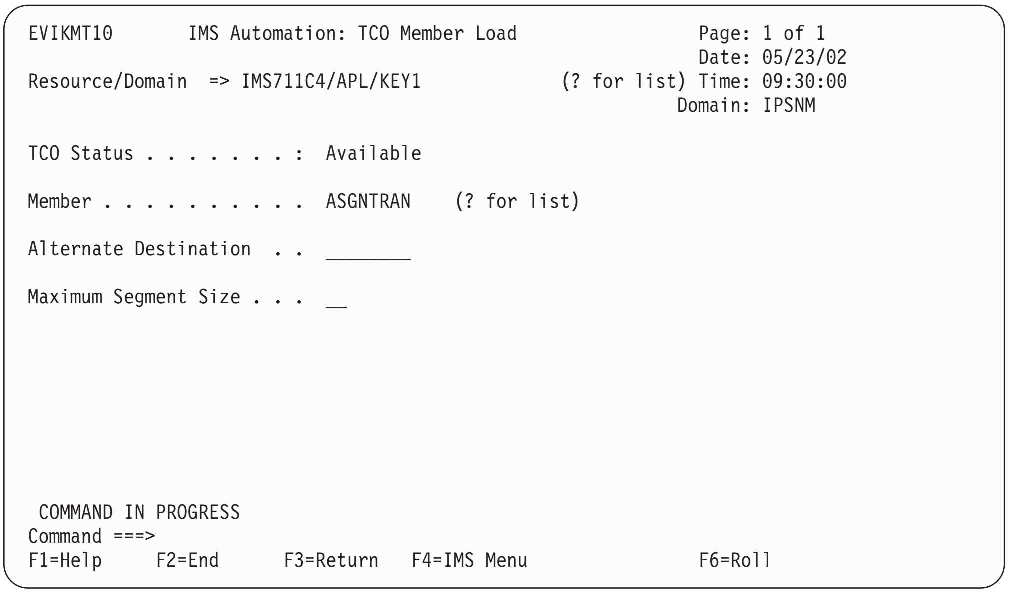
The member name is inserted. Press ENTER to load:



```
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 33. TCO Member Load Panel

You will receive a message that the command is in progress:

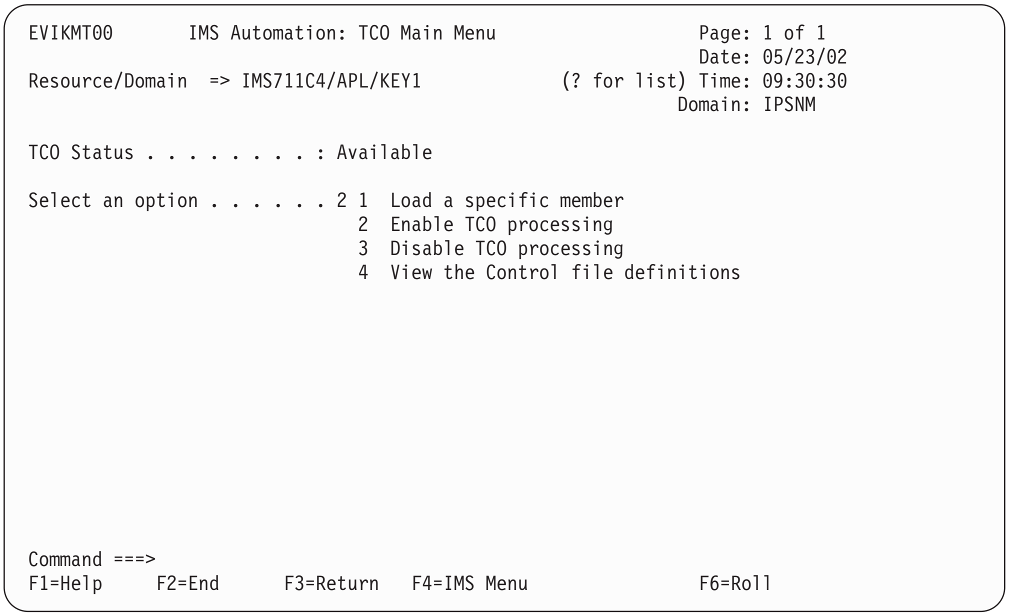


```
EVIKMT10       IMS Automation: TCO Member Load                Page: 1 of 1
                                                               Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1             (? for list) Time: 09:30:00
                                                             Domain: 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 34. TCO Member Load Panel

TCO Main Menu Option 2 — Enable TCO Processing

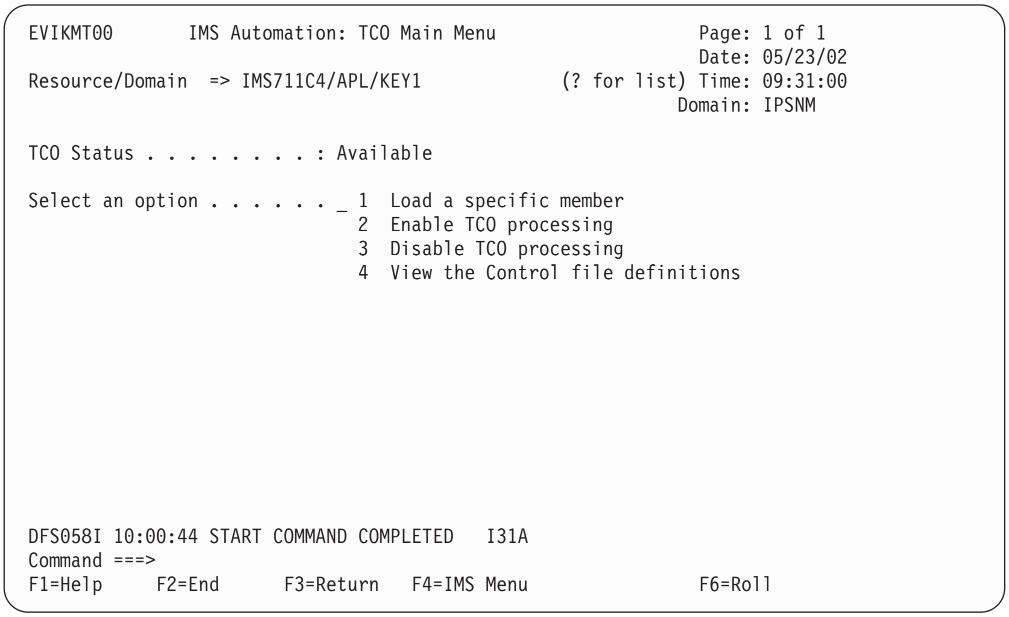
Select Option 2 from the TCO Main Menu to enable TCO processing:



```
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 35. TCO Main Menu

You will receive a message when the START command has completed:

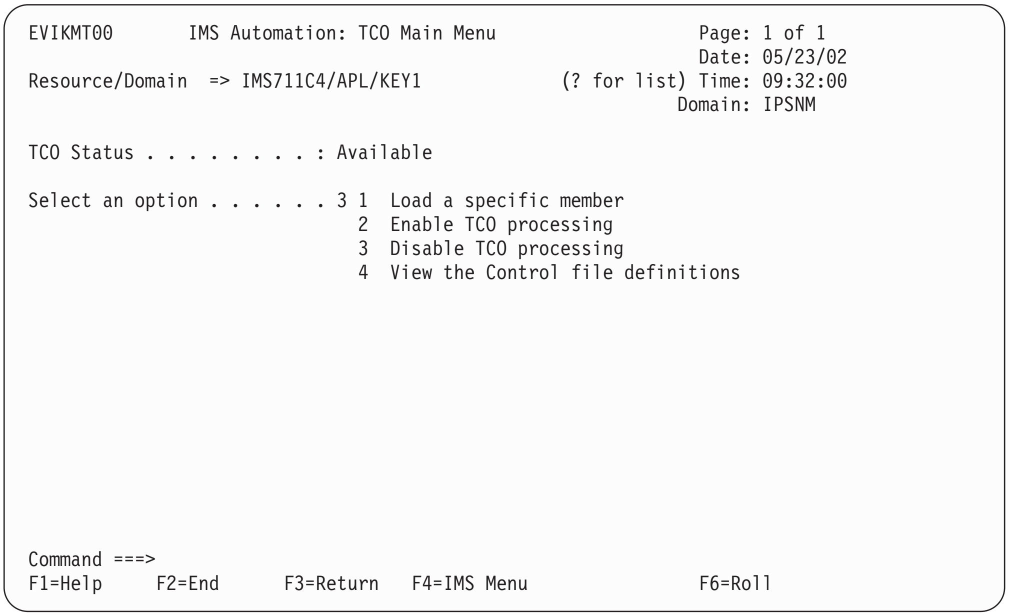


```
EVIKMT00       IMS Automation: TCO Main Menu                  Page: 1 of 1
                                                              Date: 05/23/02
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 09:31:00
                                                            Domain: 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 36. TCO Main Menu - Enable TCO Processing

TCO Main Menu Option 3 — Disable TCO Processing

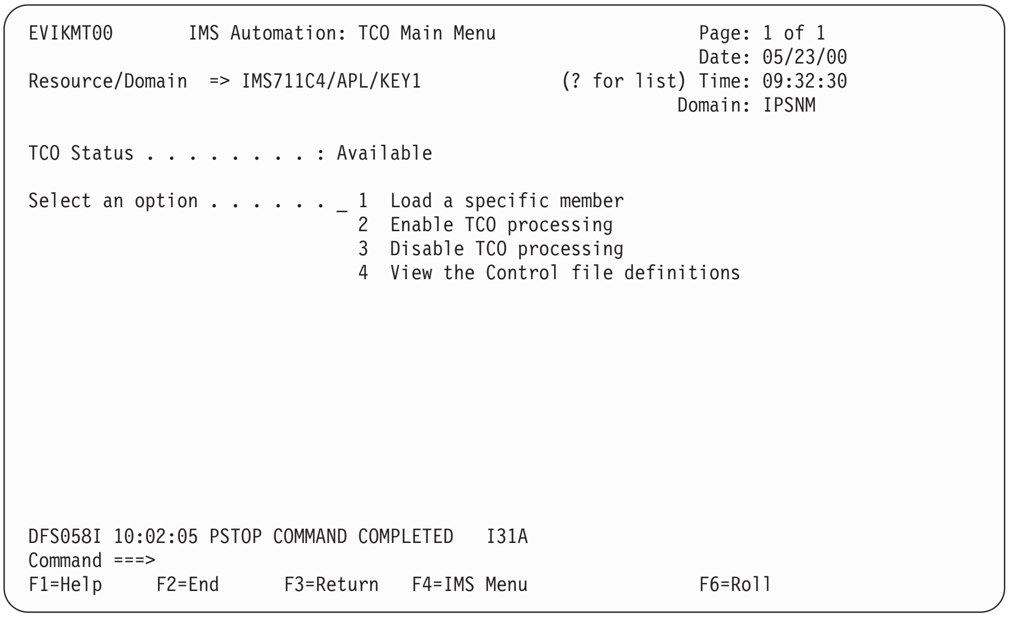
Select Option 3 from the TCO Main Menu to disable TCO processing:



```
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 37. TCO Main Menu

You will receive a message when the PSTOP command has completed:

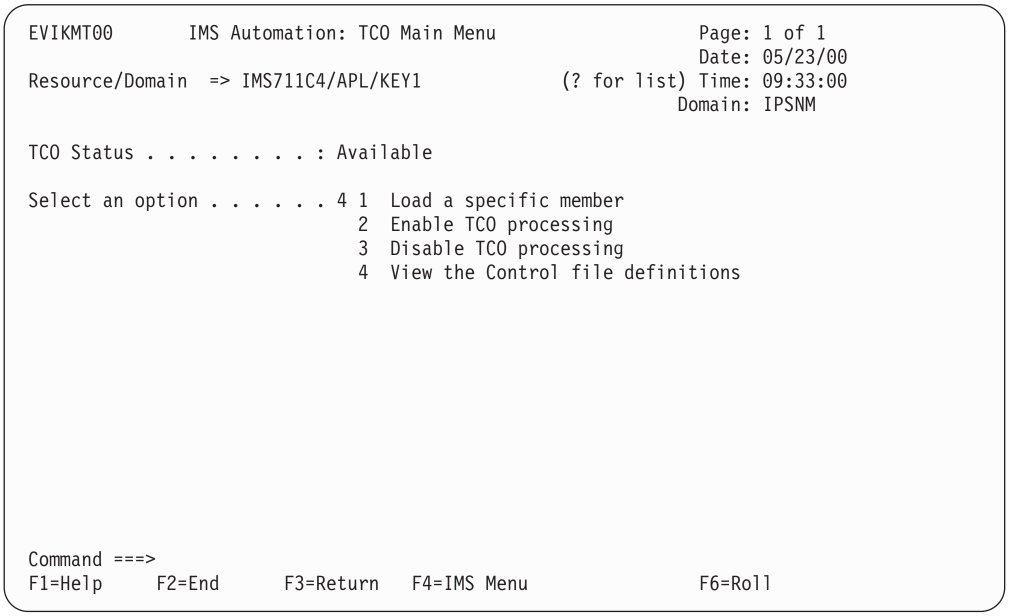


```
EVIKMT00       IMS Automation: TCO Main Menu                  Page: 1 of 1
                                                              Date: 05/23/00
Resource/Domain  => IMS711C4/APL/KEY1            (? for list) Time: 09:32:30
                                                            Domain: 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 38. TCO Main Menu - Disable TCO processing

TCO Main Menu Option 4 — View TCO related Message IDs

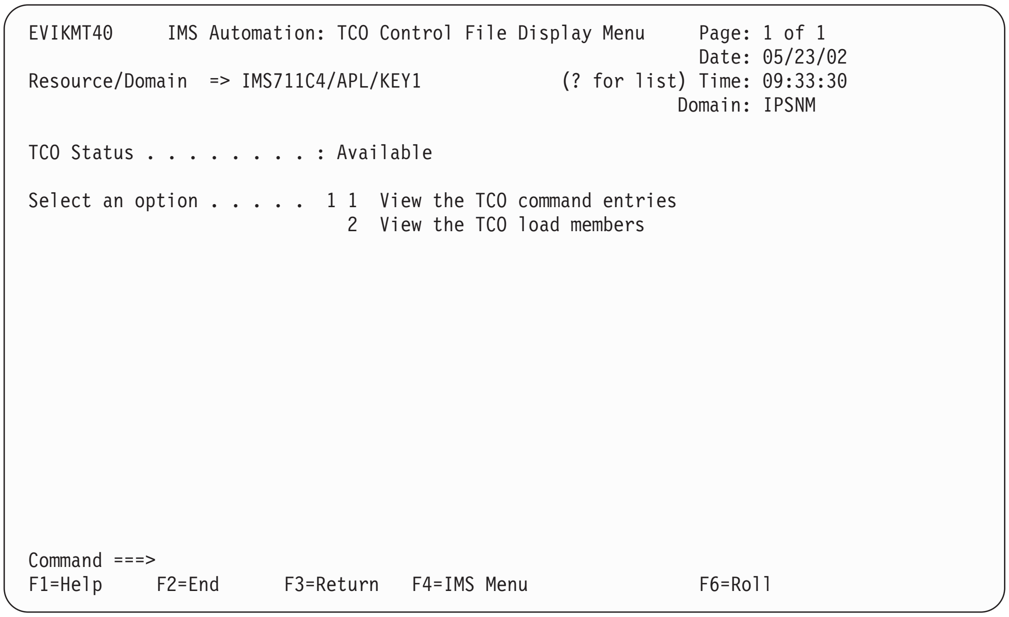
Select Option 4 from the TCO Main Menu to view the definitions of the TCO and TCOMEMBERS reserved message IDs (see "TCO—Issue Commands for Time-Driven Procedures" on page 85 and "TCOMEMBERS—Define TCO Members" on page 86):



```
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 39. TCO Main Menu

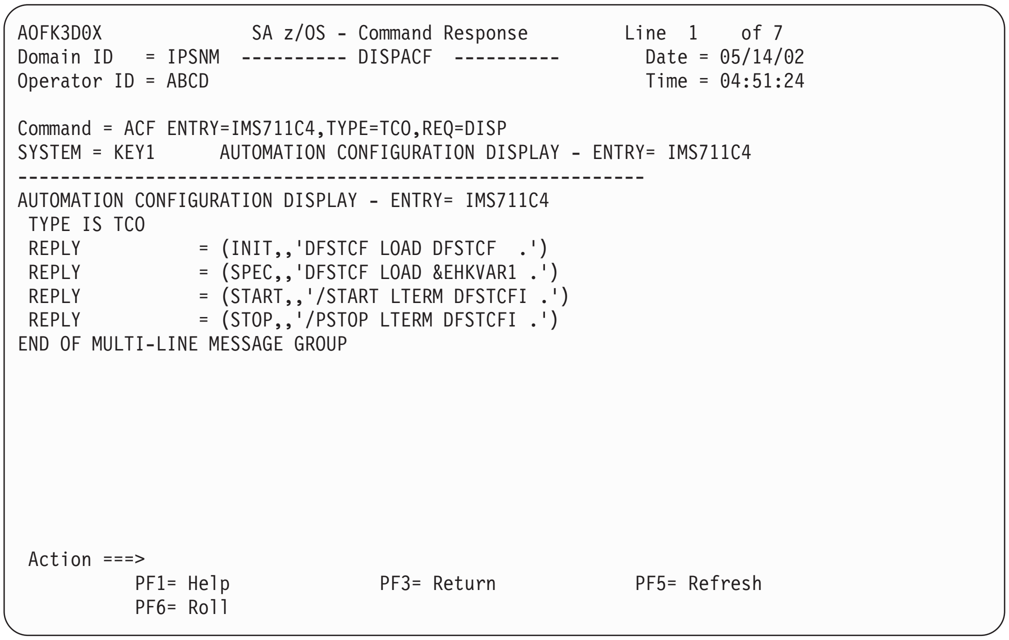
The following panel will be displayed. Select Option 1 to view the TCO command entries:



```
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 40. TCO Control File Display Menu

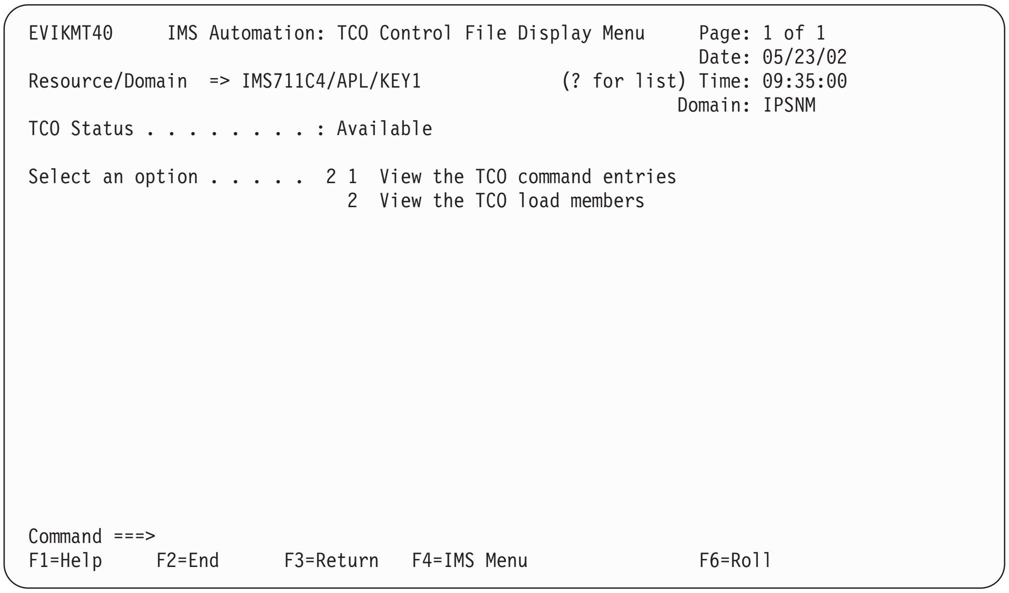
The following panel will be displayed:



```
AOFK3D0X              SA z/OS - 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 41. TCO Configuration Display Panel

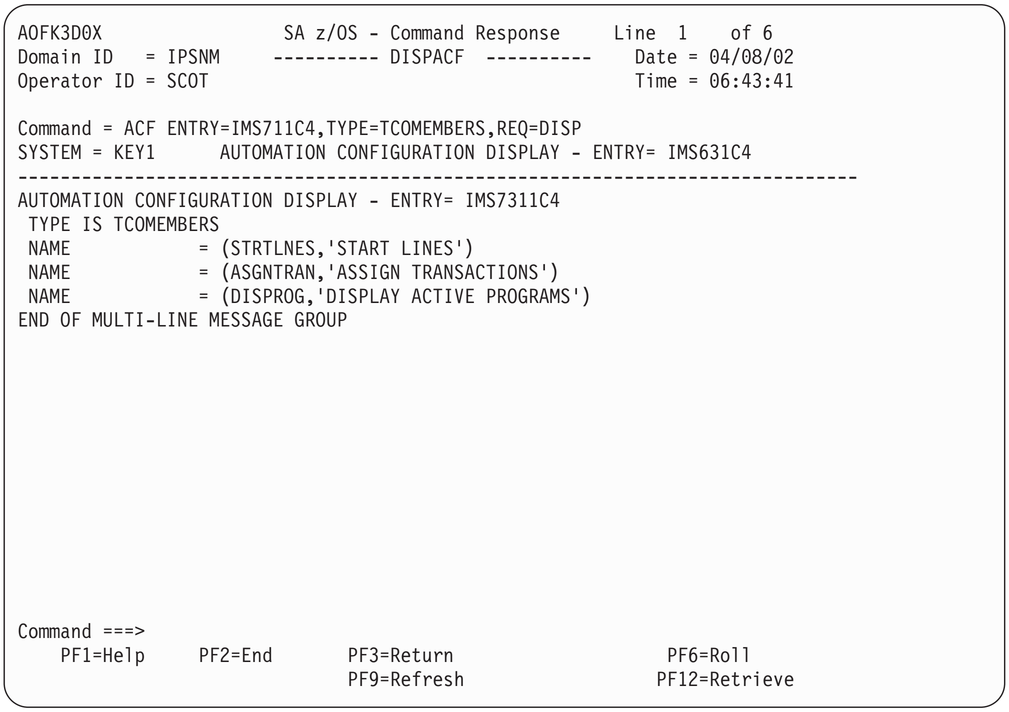
Select Option 2 from the TCO Control File Display Menu to view the TCO load members:



```
 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 42. TCO Control File Display Menu

The following panel will be displayed:



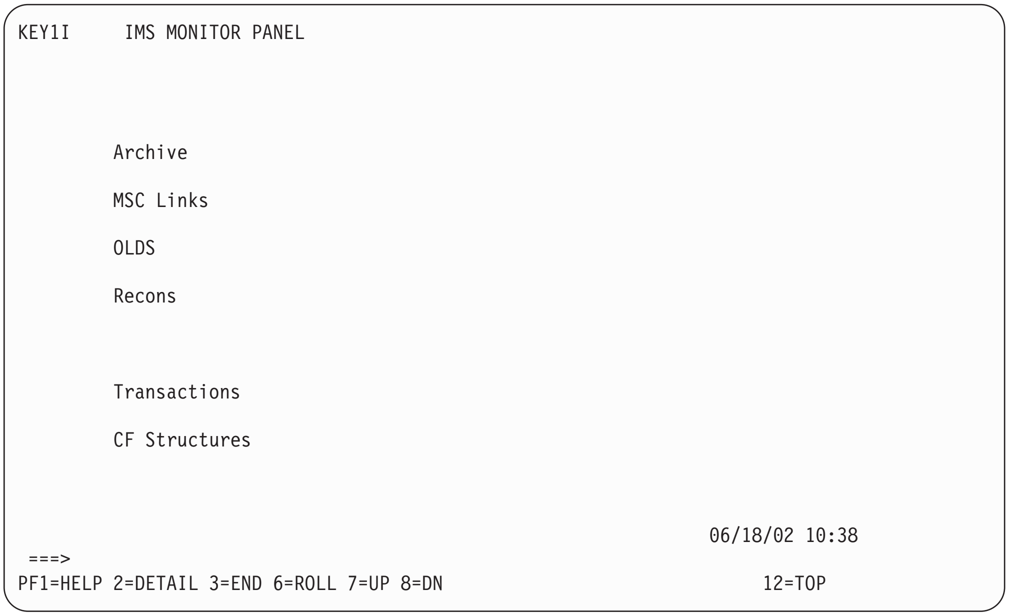
```
AOFK3D0X                 SA z/OS - Command Response     Line  1    of 6
Domain ID   = IPSNM     ---------- DISPACF  ----------    Date = 04/08/02
Operator ID = SCOT                                        Time = 06:43:41

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 43. TCO Configuration Display Panel

Chapter 10. Displaying Critical Messages

Through the Critical Message Manager, IMS Automation utilizes the SA z/OS Status Display Facility (SDF). The Critical Message Manager displays critical messages in a prioritized, scrollable format. Messages are defined as critical in the AT.

Select option 7 from the main menu to display Figure 44.



```
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
```

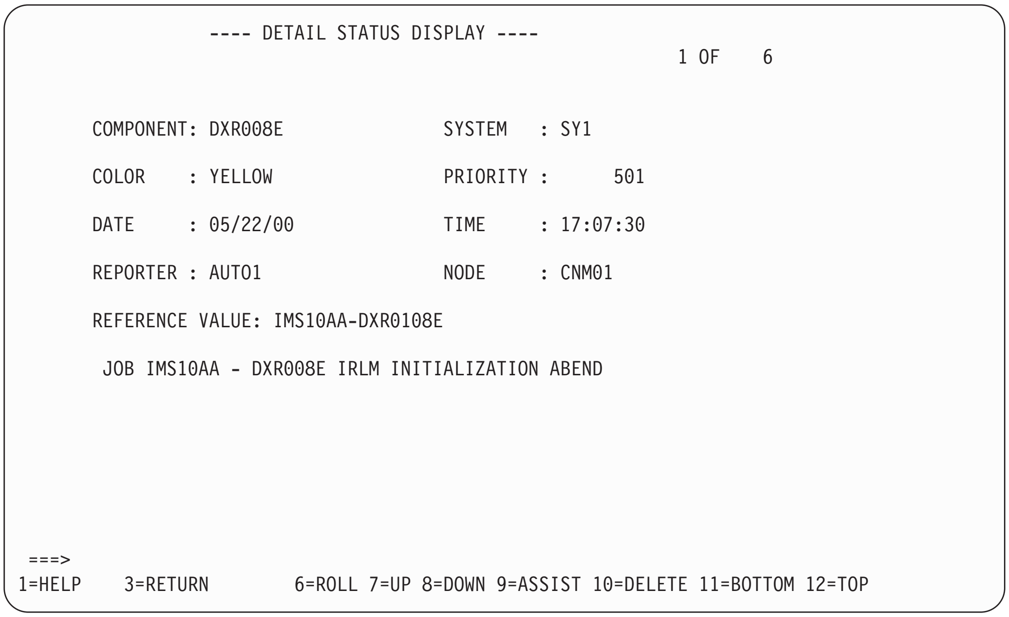
Figure 44. Critical Messages Manager Panel. If critical messages have accumulated for a category, the category will be highlighted.

SDF's IMS Monitor Panel, shown in Figure 44, 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 45 shows a sample critical message display.



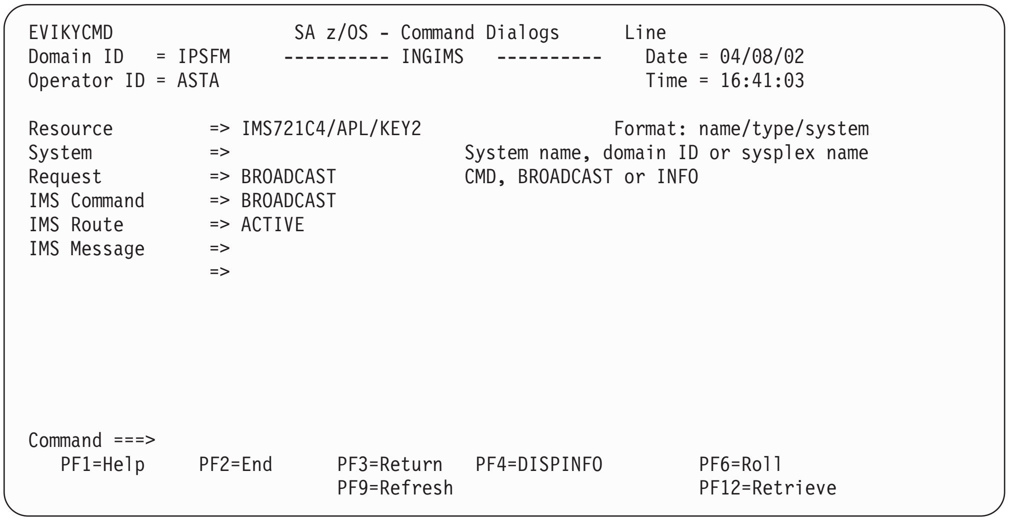
```
                 ---- DETAIL STATUS DISPLAY ----
                                                        1 OF    6


        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 45. 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 *IBM Tivoli System Automation for z/OS Operator's Commands*.



```
EVIKYCMD                 SA z/OS - 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 46. 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

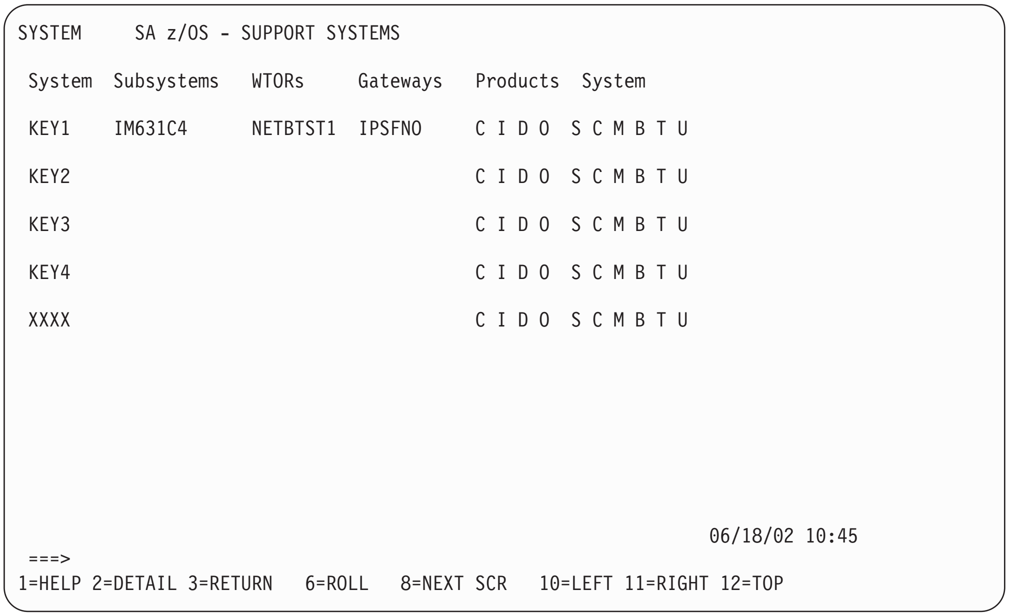
OLDS Shows the error status of OLDS datasets.

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:



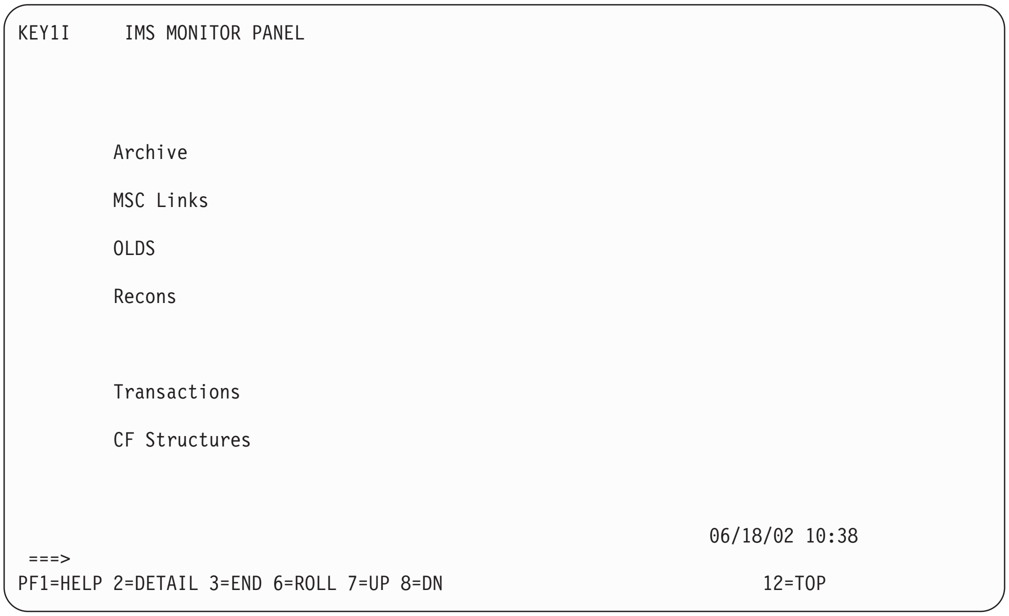
```
SYSTEM     SA z/OS - SUPPORT SYSTEMS

 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 47. 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.

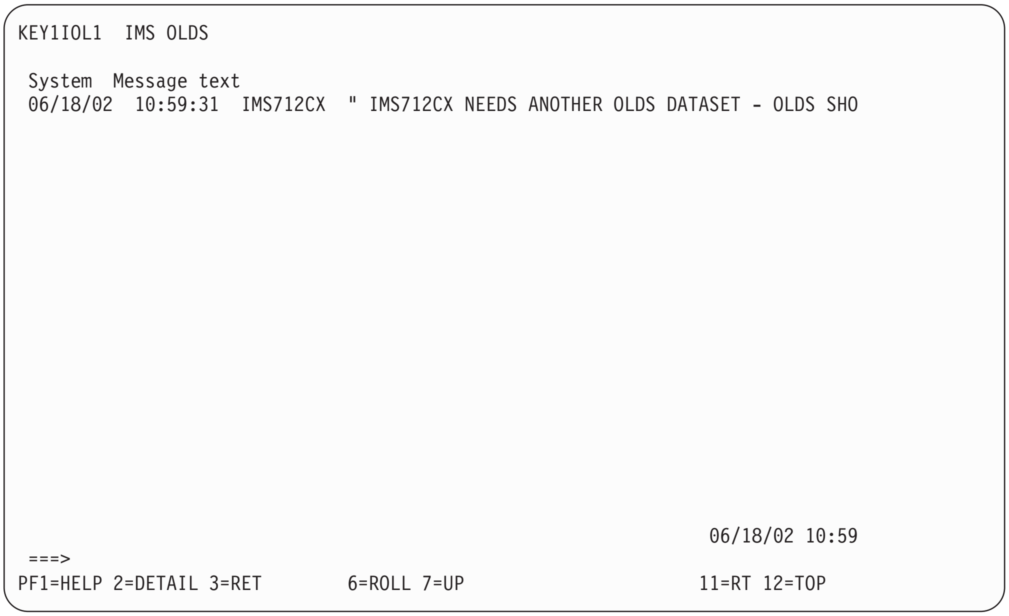
If you place the cursor under the letter C on the panel displayed in Figure 47 and press PF8, the following panel displays (assuming you are using the default sample panels):



```
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 48. 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:

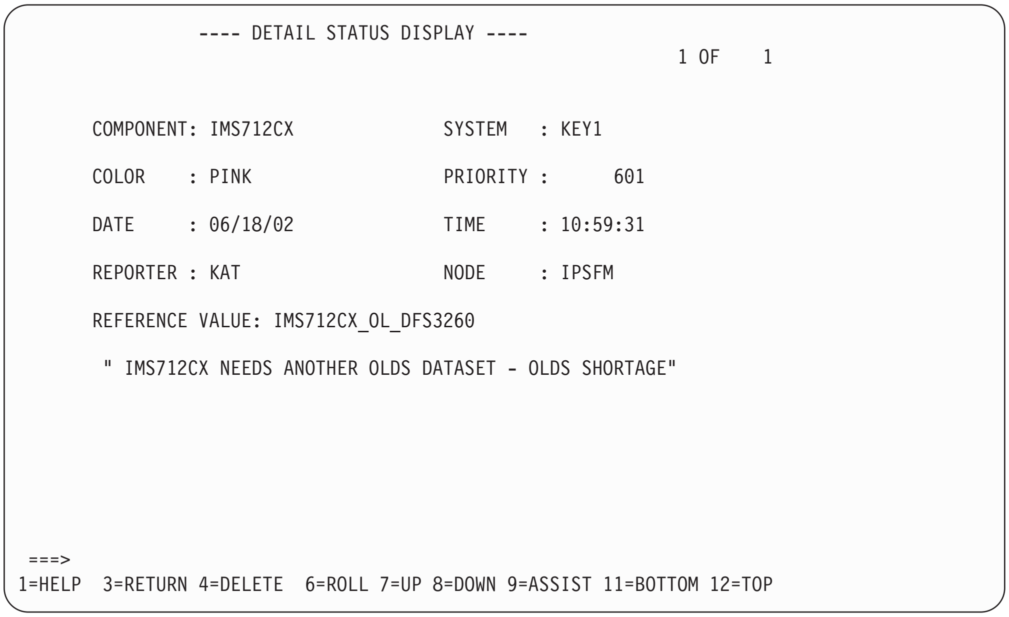


```
KEY1IOL1  IMS OLDS

 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 49. 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 50. 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 z/OS 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 z/OS.

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

Readers' Comments — We'd Like to Hear from You

IBM Tivoli System Automation for z/OS
IMS Automation
Programmer's Reference
and Operator's Guide
Version 3 Release 1

Publication No. SC33-8268-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

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

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You

SC33-8268-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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

Cut or Fold
Along Line

Program Number: 5698-SA3

Printed in USA

SC33-8268-00