CA SOLVE:Operations® Automation for CICS

Administration Guide

Release 11.9

CA Technologies Product References

This document references the following CA Technologies products:

- CA Automation Point™
- CA NetMaster® Network Management for SNA (CA NetMaster NM for SNA)
- CA Network and Systems Management (CA NSM)
- CA OPS/MVS® Event Management and Automation (CA OPS/MVS EMA)
- CA Service Desk
- CA SOLVE:Central™ Service Desk for z/OS (CA SOLVE:Central) (It includes SOLVE:Problem.)
- CA SOLVE:Operations® Automation
- CA SOLVE:Operations® Automation for CICS
- CA SYSVIEW® Performance Management (CA SYSVIEW)
- CA TCPaccess™ Communications Server (CA TCPaccess CS)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 13: Implementing Services 135

Chapter 14: Implementing the Graphical Monitor 149

Chapter 15: Setting Up the Alert Monitor 167

Chapter 1: Introduction

This section contains the following topics:

Overview

You can define CICS resources in the knowledge base and bring those resources into your service-driven operations (SDO) strategy.

CA SOLVE:Operations Automation for CICS manages defined CICS started tasks and jobs by monitoring system messages and reacting to them. It manages defined CICS resources by monitoring CICS transient data messages and reacting to them.

Intended Audience

This guide is intended for technical personnel responsible for the planning, setup, and maintenance of your product's functions and services.

Typographic Conventions

This table explains the conventions used when referring to various types of commands and when indicating field attributes.

Convention	Description
Commands	Commands such as SYSPARM and SHUTDOWN are shown in uppercase.
User Entries	Information to enter onto panels is displayed in **bold** text.
Cross-References	Cross-reference links to other sections of the book are displayed as underlined blue text.

Convention	Description
Shortcuts	Shortcuts to menus or options are displayed in **bold**, for example, **/PARMS**.

System Services

System Services provides a central core of basic functions and services.

Operator Console

Operator Console Services (OCS) provides an operator environment for command entry to monitor and control your region.

OCS is used in conjunction with the following system services:

Activity Log

Allows you to access all the commands, messages, or errors that have been issued and logged in the region for any given day

Network Information Utility File

Provides descriptions of errors and codes that are displayed in OCS

Remote Operator Facility (ROF)

Allows you to monitor and control remote regions through OCS

Network Partitioning Facility (NPF)

Allows you to subdivide your network so that different parts are controlled by different operators

Event Distribution Services (EDS)

Allows you to filter out unwanted messages in OCS before they are passed to an application procedure

Multiple Access Interface-Operator Console (MAI-OC)

Allows you to log on to VTAM applications for monitoring and control

Security

Security for your system is provided by the User ID Access Maintenance Subsystem (UAMS). UAMS provides logon and password checking facilities, and the ability to control the authority and privileges of users. It can work together with your external security system.

Note: For more information, see the *Security Guide*.

Broadcast Services

Broadcast Services let you send broadcast messages to all users. Messages can be sent to terminals or can be sent to specific users based on selection criteria.

Print Management

The Print Services Manager (PSM) is a spooling facility that lets you control the physical printing of the reports your organization generates on JES or network printers.

Communications

Several facilities enable communication between regions and programs, and collect the following types of message flows:

Inter-Network Management Connection (INMC)

Lets you establish and monitor links between multiple regions.

Advanced Program-to-Program Communication (APPC)

Lets you use the APPC protocol to connect multiple regions.

Inter System Routing (ISR)

Lets you use INMC to provide centralized control at the system level.

Program-to-Program Interface (PPI)

Enables programs to communicate with each other.

Report Writer

Report Writer provides a facility for defining report layouts and generating reports to suit your particular site requirements.

Application Development

Using application development facilities, you can write your own menus, panels, and applications using the following facilities:

- Network Control Language
- REXX
- Managed Object Development Services

Network Control Language

Network Control Language (NCL) is the interpretive language that is used to develop procedures, which can be executed by your product.

Note: For more information about NCL and its features, see the *Network Control Language Programming Guide* and the *Network Control Language Reference Guide*.

REXX

Your product supports the REXX language. You can write REXX programs to perform various tasks in your product. However, there are differences in the use of the supported REXX when compared with IBM's REXX.

Note: For more information, see *NetMaster REXX Guide*.

Managed Object Development Services

Together with NCL, Managed Object Development Services (MODS) lets you create your own applications and develop panels to provide access to them. The following features are available:

Application Register

The definitions of all applications that are built in MODS must be defined in the application register.

Common Application Services (CAS)

A collection of high-quality, special-purpose NCL routines designed to facilitate program development.

Panel Services

A facility to create and maintain full-screen panel definitions.

Mapping Services

A facility that enables programmers to define complex data structures for use by NCL applications.

Administrative Functions

Maintains MODS control libraries, panel libraries, and object services support functions.

Note: For more information about MODS, see the *Managed Object Development Services Guide*.

Automation Services

Automation Services is a collection of facilities that let you manage resources in your system. Automation Services provides operational control and desired state automation, or a framework for performance monitoring.

Automation Engine

The automation engine reacts to events occurring in the region to ensure that the managed resources are in their desired states.

It relieves you of routine operational tasks so that you can concentrate on abnormal operational conditions.

It also minimizes the impact of a failure to the users of a service. If a failure occurs while the service is idle, the service is recovered automatically before a user requires that service. That is, the users are unaware of the failure and see the service as continuously available.

Information is stored in the knowledge base as definitions. The main types of information are ServiceView, ResourceView, and EventView definitions.

Knowledge Base

Automation Services uses a knowledge base to maintain resource information. System images, in which you define resources that are to be managed by a region, are part of the knowledge base.

ServiceView Definitions

ServiceView enables you to create a service from a group of resources that provide a business function. The region manages the availability of the service according to its definition. By creating services, you can directly monitor the health of the business functions you are providing.

ResourceView Definitions

ResourceView enables you to define the resources you want to manage. You define local resources to a system image. The image holds the information about how you want the region to manage the defined resources. A region manages the resources local to the system.

You can create different images to reflect the different ways you want to manage the resources on a system. During operation, you change the management methods by loading an appropriate image.

Each resource definition you create contains information about how you want to manage that resource. You can also define relationships between resources to control the startup and shutdown sequence.

EventView Definitions

EventView enables you to manage events. In particular, it helps you manage those events that are not resource based. You can define the following:

- Rule sets to perform the following functions:
  - Manage the message flow on the local system.
  - Execute event-based actions.
- Message profiles to consolidate message flow from different systems to a single console.

Automation Services Processes

An Automation Services process is a means of performing an action. You can define a process and have it executed on demand, or automatically on behalf of a resource.

Processes are defined by selecting one or more macros. A set of macros is distributed with your product. For example, there are macros to issue operating system commands, generate alerts, issue an SNMP TRAP, and issue a WTO. Macro parameters are supplied through full-screen panels, so that a process is easy to build and requires no programming knowledge.

Multisystem Support

Automation Services provides focal point management to support multisystem operation (that is, management at a focal point with subordinates feeding information to it) as follows:

Peer-to-peer architecture

Supports full connectivity between multiple regions. Regions linked in this way are known as focal point regions. (A stand-alone region is also regarded as a focal point region.)

When regions are communicating with each other, authorized users can monitor and control all managed resources from any terminal connected to a focal point region.

All focal point regions have the knowledge base synchronized.

Subordinate

Enables you to reduce the amount of traffic in your multisystem environment. You link subordinates to focal point regions that provide central monitoring and control. A subordinate has visibility and control of the locally managed resources only.

Independent Operation

Each region can run independently of other regions. If no communication links are available, each region still provides full monitoring, control, and automation of its own managed resources. If one region fails, automation continues for resources that are managed by the other regions.

You can also transmit various types of data from a stand-alone region to another region.

Communication Access Methods

Multisystem links support the following communication access methods: EndPoint Services (EPS), TCP/IP, and VTAM.

Multisystem Knowledge Base

Automation Services automatically maintains synchronization between linked knowledge bases, with automatic recovery in the event of link failure.

Automation Services Administration

Automation Services provides the following implementation and administration functions:

- The ability to update prompt lists. These are lists of valid values (from which you choose one value) that are displayed when you enter a question mark (?) in a prompted field. Prompted fields are identified by the presence of a plus sign (+) at the start of the input field.
- The ability to write your own NCL procedures or macros for use in Automation Services processes.
- The ability to define commonly used monitor commands—these commands are also NCL procedures.

Application Program Interface

The application program interface (API) is available to system programmers.

The API provides the following procedures:

- $RMCALL
- $RMDBAPI
- $RMEVENT
- $RMSTSET
- $RECALL
- $REDBAPI

These procedures enable external sources to call Automation Services functions and retrieve Automation Services data.

Note: For more information about the API, see the *Reference Guide*.

Monitors

The product provides the following monitors:

- Status monitor
- Graphical monitor
- Message monitor
- Alert monitor

These monitors perform the following functions:

- Warn you of any problems that might occur.
- Enable you to manually control the managed services and resources.

Status Monitor

The status monitor displays the status of defined services and the status of defined resources in the currently-active system images. The display is in the form of a list. You can customize the status monitor to display only the services and resources of interest.

You customize a status monitor by using status monitor filters. You can selectively view different groups of services and resources by swapping filters.

Graphical Monitor

The graphical monitor enables you to roll up the status of the managed services and resources in icons. You can thus obtain a higher level view of the items being managed (that is, fewer items to monitor directly).

To use the graphical monitor, you need to create the icon panels that present the view you require.

Message Monitor

The message monitor can display messages that originate from different systems on a single screen (the consolidated console).

You define and activate message profiles to consolidate messages.

Alert Monitor

The Alert Monitor provides an event notification system that tells you that something has happened that may require some action to be taken. It displays internally generated alerts and user-defined alerts. An alert is generated, for example, when a resource fails.

You can update, track, and delete alerts from the Alert Monitor. You can raise a problem ticket from an alert. You can also forward alerts to other applications and platforms.

The Alert Monitor provides a single point to view problems in your environment.

The Alert Monitor is also available through WebCenter.

Additional Features

Besides the Automation Services features, CA SOLVE:Operations Automation for CICS provides the following additional features:

- CICS agent
- Program-to-program interface (PPI) connection between the CA SOLVE:Operations Automation for CICS region and the CICS agent
- Visibility of CICS transient data messages
- CICS resource definitions
- Autopopulation of CICS resource definitions
- CICS resource control

CICS Agent

The CICS agent resides in a CICS region. It performs the following functions:

- Monitor CICS transient data messages, and send copies of selected messages to the CA SOLVE:Operations Automation for CICS region.
- Enable the CA SOLVE:Operations Automation for CICS region to issue commands to CICS and get the responses.

Note: For information about how to install the agent, see the *Getting Started* guide.

Program-to-program Interface Connection

The PPI connects the region to the agent that resides in the CICS region. The PPI connection enables the region to receive CICS transient data messages.

For each connection, the flow of unsolicited messages can be restricted by transient data queue and message ID.

Note: For information about how to implement the connection, see the *Getting Started* guide.

Message Visibility

With CICS agents installed and PPI connections established, a CA SOLVE:Operations Automation for CICS region can monitor the following messages:

- Messages sent to the system console (captured by the Automation Services subsystem interface)
- Messages sent to the CICS transient data queues (captured by the CICS agent)

By using these messages and the operations methods defined in the knowledge base, the region monitors defined CICS resources and reacts accordingly.

Besides being able to define resources that react to messages, you can also define the following:

- EventView rules to perform actions on the receipt of particular messages
- Consolidated console message profiles to enable you to monitor the messages from a single terminal

CICS Resource Definitions

CA SOLVE:Operations Automation for CICS supports the following classes of CICS resources:

- Started task
- Job
- Transaction
- File
- Link
- Database

By defining these CICS resources in the knowledge base, you can include them in services and manage them at the service level.

Autopopulation of CICS Resource Definitions

CA SOLVE:Operations Automation for CICS automatically discovers CICS started tasks and jobs, and the resources defined in your CICS regions. You can then select the resources you want to be defined in your current system image, and CA SOLVE:Operations Automation for CICS will create the definitions for you automatically.

CICS Resource Control

CA SOLVE:Operations Automation for CICS provides the following facilities by which you can control CICS resources:

- The command, CICSCMD, enables you to implement resource control in resource definitions.
- The CMD command enables you to issue CICS commands by using the MODIFY system command.
- The MTO command enables you to establish a session with a CICS region from the status monitor.

WebCenter

WebCenter is a web browser interface to the region. The interface provides web access to functions such as monitoring and history.

The same user ID and password for your region are used to access WebCenter.

The WebCenter web server runs in the region's address space. It is entirely z/OS hosted, and requires no third-party web servers or external components.

Problem resolution time is decreased and ease of use increased. Help desk users who are not familiar with z/OS mainframe 3270 interfaces can perform management functions with their standard web browser.

Every page of WebCenter has context-sensitive online help.

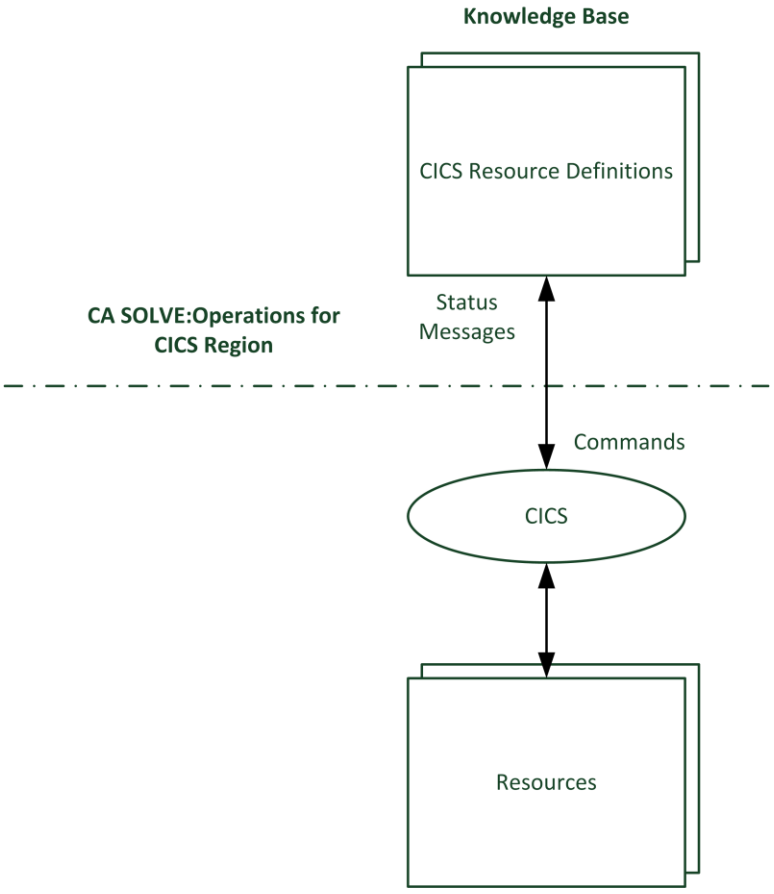
Chapter 2: CICS Resource Definitions

This section contains the following topics:

Overview

A CICS resource definition is based on the general ResourceView resource definition. It stores the operations information of a CICS resource in the knowledge base.

The following illustration shows the relationship between the definitions and the resources they represent.



Started tasks and jobs are CICS regions that own the transactions, files, links, and databases. Started task and jobs are owners. Transactions, files, links, and databases are CICS-owned resources.

Names of CICS-owned Resources

A CICS-owned resource can be defined in more than one CICS region. For example, a file can be shared between several CICS regions.

The CA SOLVE:Operations Automation for CICS region manages an owned resource in relation to the CICS region in which the resource is defined. The resource is known as *owner_name.resource_name*.

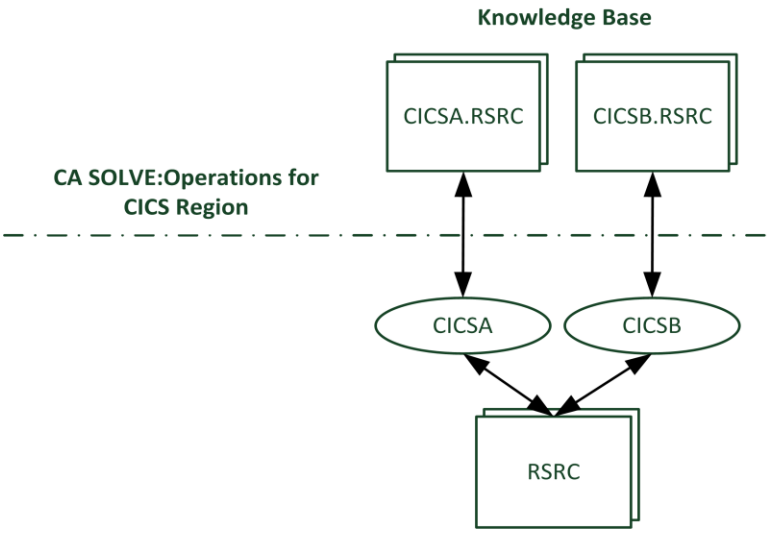
owner_name

Is the job or started task name of the CICS region in which the resource is defined.

resource_name

Identifies the resource.

The following illustration shows an example in which an RSRC resource is defined in two CICS regions, CICSA and CICSB.



CICS Resource Owner

The CICS resource owner is the CICS region in which the resource is defined. To enable the CA SOLVE:Operations Automation for CICS region to manage a defined CICS resource properly, you should ensure that the owner is defined in the same system image as the resource.

Depending on how a CICS region is established, it is defined either as a started task or a job.

Operational Relationship Between a CICS Resource and Its Owner

A CICS resource and its owner have the following operational relationships:

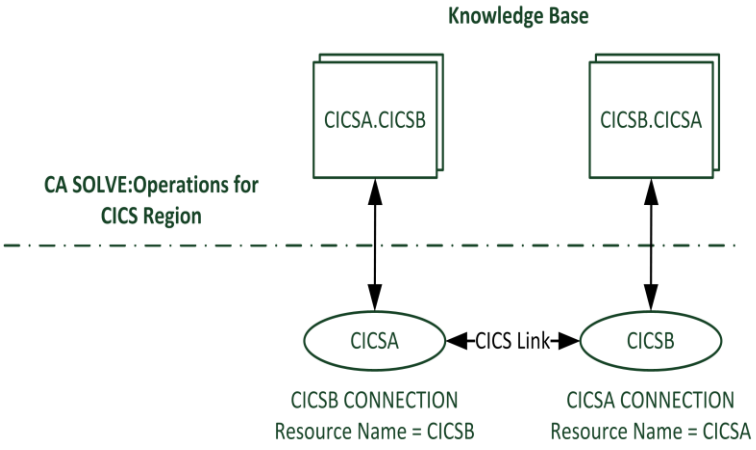
- A resource cannot be started until the CICS region that owns it has become active.
- The CA SOLVE:Operations Automation for CICS region does not inactivate an owned resource during shutdown. The CICS region that owns the resource inactivates the resource.

This resource-owner relationship is implicit. You do not need to define an explicit parent-child relationship between a CICS resource and its owner. However, you can define explicit relationships between the CICS resource and other resources.

CICS Link Resource Definitions

A CICS link resource definition enables CA SOLVE:Operations Automation for CICS to manage the link between two regions (for example, between two CICS regions, or between a CICS region and a DB2 region). This is a special CICS resource definition that has two owners: a local owner and a remote owner.

The following illustration shows examples of CICS link resource definitions. In CICSA, the CONNECTION resource to CICSB is defined as CICB. In CICSB, the CONNECTION resource to CICSA is defined as CICA. The definitions created in the knowledge base for these resources are CICSA.CICB and CICSB.CICA, representing the two ends of the link.



The owners in the CICSA.CICB and CICSB.CICA definitions are as follows:

CICS Link Resource Definition	Local Owner	Remote Owner
CICSA.CICB	CICSA	CICSB
CICSB.CICA	CICSB	CICSA

A link resource definition and its local owner should be defined in the same system image. In the example, CICSA.CICB and CICSA are defined in the same system image. Similarly, CICSB.CICA and CICSB are defined in the same system image.

ISC Link Owner Definitions

For an intersystem communication (ISC) link, you should ensure that the owners of the link are defined as follows:

- The local owner is defined in the same system image as the link resource definition (that is, in the same local region).
- The remote owner is defined in a region on the remote system.

You should also ensure that the two regions are connected.

MRO Link Owner Definitions

For a multiregion operation (MRO) link, you should ensure that the owners of the link are defined as follows:

- The local owner is defined in the same system image as the link resource definition (that is, in the same local region).
- The remote owner is defined in either one of the following regions:
  - Same region as the local owner and link resource definitions
  - Separate connected region on the same system

Operational Relationship Between a CICS Link Resource and Its Remote Owner

The remote owner of a CICS link resource does not have to be a CICS region. The remote owner can be, for example, a DB2 manager region. (To define resources other than the resources supported by CA SOLVE:Operations Automation for CICS, you require CA SOLVE:Operations Automation.)

A remote owner affects the activation of a remotely owned resource only. That is, it must be active before the owned resource can be started. The remote owner does not affect the inactivation of the resource.

Chapter 3: Starting and Stopping a Region

This section contains the following topics:

Start SOLVE SSI

To start the SOLVE SSI, issue the following command:

```
S ssiname,REUSASID=YES
```

For a region to connect to SOLVE SSI, it must first know the SSID to connect to. To identify the SSID, specify the SSID JCL parameter or use Customizer parameter groups. When this connection is complete, authorized region users can issue SOLVE SSI commands.

The region can use the SSID JCL parameter to establish an early connection to SOLVE SSI during initialization.

This parameter has the following format:

```
SSID={ NO | * | name }
```

NO

(Default) Does not attempt to connect to SOLVE SSI. The connection is only started (or attempted) after a SYSPARMS SSID command is issued.

*****

Starts or attempts a connection to an SSID with the first four characters of the region job name.

name

Starts or attempts a connection to the specified SSID.

If asterisk (*) or *name* is specified, an attempt to connect to the SSI is immediately made. If it fails, it retries every *n* seconds, depending on the default value of the SSI retry interval.

Note: To change the value of the SSID to connect at any time, update the SSI parameter group (enter **/PARMS**). You can use this parameter group to change the SSID value or to specify an SSI retry interval.

Stop SOLVE SSI

To stop SOLVE SSI, use *one* of the following methods:

- Enter the following command:

  ```
  SSI STOP
  ```

- Enter the following operating system STOP (P) command:

  ```
  P ssiname
  ```

Note: If you use cross memory services but have not specified REUSASID=YES when you start SOLVE SSI, the address space running SOLVE SSI terminates and is not available until after the next IPL.

Start a Region

To start a region, you run it as a job or a started task. A started task has been set up during the installation process.

To start a region, issue the following command:

```
S rname,REUSASID=YES
```

Users log on to a region by using the user IDs and passwords specified in their UAMS (or external security package) records.

WTOR Confirmation Message

If you have implemented region startup confirmation, the RMIWTO06 WTOR message is displayed and startup pauses.

The WTOR message enables you to change the startup parameters. If a reply to the message is not made in 120 seconds, startup continues.

Note: For information about startup confirmation, see the online help for the AUTOIDS parameter groups.

Continue Startup with No Change

To continue startup with no change to the parameters, reply as follows:

```
R n,U
```

n is the identification number of the WTOR message.

Continue Startup with Changes

To continue startup with changes to the parameters, reply as follows:

```
R n,parameter-1=value-1[,parameter-2=value-2[,…[,parameter-n=value-n]]]
```

You can use the following parameters in your reply. The parameters change the field values in the AUTOIDS parameter group specification panel that affects the loading of the system image.

SYSTEM

Corresponds to the System Image Name field.

VERSION

Corresponds to the Version field.

MODE

Corresponds to the Automation Mode field.

COLD

Corresponds to the Cold Start on Next Restart? field.

If you reply to change parameters, you are asked to confirm your changes. You can then make additional changes or accept the displayed values.

Example: Load a Different System Image

This example reply changes the system image to load to PROD version 2:

```
R n,SYSTEM=PROD,VERSION=2
```

Stop a Region

If you have the necessary authority, you can shut down the region.

To stop a region, issue the operating system STOP (P) command.

You can also stop a region by issuing *one* of the following commands: **SHUTDOWN** or **FSTOP**.

SHUTDOWN Command

The SHUTDOWN command stops the region when the last user logs off. When you issue the SHUTDOWN command, a broadcast is issued to all users. No further logons are accepted until the region is restarted, or the SHUTDOWN CANCEL command is issued.

You can issue the SHUTDOWN command from OCS or Command Entry. Alternatively, you can issue it as a z/OS MODIFY command.

Note: For more information about the SHUTDOWN command, see the online help.

FSTOP Command

The FSTOP command immediately disconnects user sessions and shuts down the region.

Restrict the use of the FSTOP command.

You can issue the FSTOP command from OCS or Command Entry. Alternatively, you can issue it as a z/OS MODIFY command.

Important! If you are running another product in the same region, it also stops if the FSTOP command is issued.

Note: For more information about the FSTOP command, see the online help.

How You Preserve Data When Region Stops and Restarts

You can preserve some data when a region stops so that this data is available when the region restarts. You can use global variables to preserve data. You can save global variables that the region reloads when it restarts. Saved global variables are known as persistent global variables.

To preserve data, create global variables with data you want to preserve and save them, for example:

- Use the Persistent Variables Administration option (access shortcut is /PVARS).
- Call the $CAGLBL procedure using the SAVE option.

  Note: For information about the $CAGLBL procedure, see the *Network Control Language Reference Guide*.

Create Persistent Global Variables Using the User Interface

You can create persistent global variables from the Persistent Variables List panel. The panel also lets you maintain those variables, for example, update, purge, or reload them.

To create a persistent global variable using the user interface

1. Enter the **/PVARS** panel shortcut.

   The Persistent Variables List panel appears.

2. Press F4 (Add).

   The Persistent Variable - Add panel appears.

3. Specify the name of the variable (without its global prefix) and its value. Press F3 (File).

The variable is saved so that it can be loaded the next time the region starts up.

Prevent the Reloading of Preserved Data

If problems occur during region startup because of invalid data being loaded, you can disable the reloading of the preserved data.

To prevent the reloading of preserved data, enter the following command when you start the region:

```
S rname,PARM='XOPT=NOPVLOAD'
```

The region starts without reloading the preserved data.

Chapter 4: Configuring a Region

This section contains the following topics:

Region Configuration

After you have completed installation and startup, your region is operational at a basic level; however, you must configure it to suit your requirements.

How You Use JCL Parameters to Configure a Region

JCL parameters enable you to configure a region. You use JCL parameters to set region information. This information includes, for example, the names of your INIT and READY procedures, and the types of security exit to use in your region.

This information is supplied by the PPREF statements in the RUNSYSIN member.

You can also pass this information in the START command using the JCL PARM field. If you specify multiple parameters, separate each with a comma.

Note: For more information, see the *Reference Guide*.

How You Display and Change JCL Parameter Settings

You can display the current settings of all the JCL parameters with the SHOW PARMS command from OCS or Command Entry. To change any of these parameters, specify their new values in the RUNSYSIN member and then restart the region.

Note: For more information about JCL parameters, see the *Reference Guide*.

How You Identify the Region to Users

If you have multiple regions or communicate with other regions, you can set the domain ID and put titles on the panels.

How You Identify Domains and Panels

The NMDID JCL parameter identifies the domain ID for each region. If you have multiple regions, specify a different domain ID for each one.

Note: For more information about the NMDID parameter, see the *Reference Guide*.

You can use the SYSTEMID (System Identifications) parameter group in Customizer to help identify your regions. This parameter group specifies a system identifier that is used when you link to other regions. Specify a different system identifier for each of your regions.

This parameter group also specifies the titles to display on the logon panel and the OCS console panel. These titles help users to identify the region that they have logged on to.

Note: The system ID parameter takes effect when the region is initialized.

Region Customizer

Customizer lets you review and update parameter groups.

You use Customizer to initialize and customize your region. Customizer is an initialization facility that lets you implement a region rapidly and easily. Also, Customizer enables you to customize parameters easily at a later stage.

When you first install a product, you set various parameters to get the product up and running. Customizer helps you set up these parameters. An initial dialog is supplied for the first time user, to walk you through the customization process. You are prompted to supply required and optional parameter values.

To access the parameter groups, enter **/PARMS**.

What Are Parameter Groups?

System parameters are grouped by category (such as Security) in logical parameter groups, to simplify the process of initializing and customizing a region.

Groups of individual parameters translate into one or more of the following:

- SYSPARMS that determine how your region functions
- Global variables that various NCL applications use to control their functions
- Local parameters that define how to implement actions associated with parameter groups

Print Parameter Group Settings

You can print the parameter group settings in a region for analysis. The output is in the same format as the initialization file (see page 53).

To print parameter group settings

1. Enter the **/PARMS** panel shortcut.

   The Parameter Groups panel appears.

2. Enter **PRINT** at the Command prompt.

   The Confirm Printer panel appears.

3. Specify your printing requirements, and press F6 (Confirm).

   The parameter group settings in the region are printed.

System Parameters

Most customization of your region is performed by using Customizer.

You can also use the SYSPARMS command to customize your region. Each operand of the SYSPARMS command lets you specify options to change and customize the way your region works. For ease of maintenance, you can use the Display/Update SYSPARMS panel, which is accessible by using the /SYSPARM panel shortcut.

Notes:

- SYSPARMS set by Customizer parameter groups can only be updated using Customizer.
- For SYSPARMS without a corresponding parameter group, set the SYSPARMS in the INIT and READY procedures so that they are applied when the region starts. You can update them dynamically using the SYSPARMS command.
- For more information about SYSPARMS operands, see the *Reference Guide*.

Use the SYSPARMS Command

To change a SYSPARMS operand with the SYSPARMS command, enter the following command at the OCS command line:

```
SYSPARMS operand=value operand=value ...
```

Example: Display Time on OCS Title Line

This example sets the time display at the beginning of the OCS title line using the following command:

```
SYSPARMS OCSTIME=YES
```

Initialization Operands

There are some SYSPARMS command operands that cannot be changed while the region is operational. These operands must be included in your INIT procedure so that they are executed during initialization.

Note: For a complete list of SYSPARMS commands, see the *Reference Guide*.

If you specify new values for these initialization operands, the new values do not take effect until the region is initialized. All other SYSPARMS can be changed during region operation by authorized users.

Audit Region Activities

Auditing lets you store selected region activities as System Management Facility (SMF) records. You can then extract those records using your own reporting tool. You can also write audit events as messages to your product region's activity log. The logged messages have IDs from NMAU0111 to NMAU0116. When auditing is enabled, the region automatically audits activities such as suppression of messages and updates to definitions in the knowledge base. You can add site-specific auditing using the $NMAUAPI Audit API.

Note: For information about the Audit API and SMF records, see the *Reference Guide*.

To audit region activities, you must enable it through the AUDIT parameter group. The group lets you specify if you want to store the activities as SMF records, log the audit events, or both. If you want to create SMF records, you must also ensure that an ID for SMF records with subtypes is specified in the SMF parameter group.

To enable auditing of region activities

1. Enter the **/PARMS** panel shortcut.

   The Parameter Groups panel appears.

2. If you want to generate audit events as SMF records, get a unique SMF record ID in the range 128 to 255 from your systems programmer and then enter **FIND SMF** to locate the SMF parameter group.

   a. Enter **U** beside the group.

      The group opens for updating.

   b. Specify the assigned ID in the SMF Record Identifier (Subtyped) field, and press F6 (Action).

      The ID is defined for audit event records.

   c. Press F3 (File).

      The group is updated with the changes.

3. Enter **FIND AUDIT**.

   The cursor locates the AUDIT parameter group.

   a. Enter **U** beside the group.

      The group opens for updating.

   b. Specify the auditing requirements for the following types of events:

      - ACCESS lets you audit security activities.
      - APPLICATION lets you audit activities generated by your applications.
      - CONFIGURATION lets you audit changes to definitions.
      - PROCEDURAL lets you audit actions performed at monitors.
      - SERVICEABILITY lets you audit changes in status of monitored resources.
      - UTILIZATION lets you gather statistics on the audited activities. You can customize the frequency and time by which the statistics are gathered.

      Your region does not generate audit events of the ACCESS and APPLICATION types. However, you can use the API to generate those events.

      Note: For more information about event types and audited objects, see the online help.

      Press F6 (Action).

      The region starts to audit the specified activities and create the specified records (SMF, log, or both).

   c. Press F3 (File).

      The group is updated with the changes.

Capture Messages Not Handled by Rules

If you want to capture certain messages missed by your resource definitions and message rules, use the Unmatched Message Alerting (UMA) feature. By capturing these messages, you can review them later to create rules for them.

To capture messages not being handled by your resource definitions and message rules

1. Enter the **/PARMS** panel shortcut.

   The Parameter Groups panel appears.

2. Enter **F MSGAWARENESS**.

   The cursor locates the MSGAWARENESS parameter group.

3. Enter **U** next to the group.

   The group opens for updating.

4. Specify **ACTIVE** in the Unmatched Message Alerting field, and customize the parameters to capture the type of messages you require.

   Unmatched Message Alerting Filter

   Specifies the type of messages you want to capture.

   Unmatched Message Alerting Options

   Specifies how you want to be notified of the captured messages. The notification can be by one or all of the following methods:

   - Raise alerts.
   - Log the occurrences of the messages.
   - Issue EDS events.

   Press F6 (Action).

   The region starts to capture the specified messages.

5. (Optional) Press F3 (File) if you want to make the changes permanent.

   The group is updated with the changes.

Transient Log Tuning

A *transient log* is a log of activities associated with a resource that is monitored. One transient log exists for each resource definition loaded in a region and exists as long as the definition remains loaded in the region. You can specify the age over which logged activities are deleted to keep their number down. When the default size parameters do not suit your requirements, you can customize them. You can also change the size of the transient logs for selected resources.

Customize Tuning Parameters

The AUTOTABLES parameter group contains the tuning parameters for transient logs. The parameters control the default and maximum sizes, and the deletion of logged activities that are over a specified age. For example, when overflows occur in the logs, you can lower the maximum size while you investigate the cause of the problem.

To customize the tuning parameters for transient logs

1. Enter the **/PARMS** panel shortcut.

   The Parameter Groups panel appears.

2. Enter **F AUTOTABLES**.

   The cursor locates the AUTOTABLES parameter group.

3. Enter **U** beside the group.

   The group opens for updating.

4. Customize the parameters for transient logs to suit your requirements. Press F6 (Action).

   The changes are applied in the region.

5. (Optional) Press F3 (File) if you want to make the changes permanent.

   The group is updated with the changes.

Resize Selected Transient Logs

After your region operates for a while, you may find that you need to tune the size of some transient logs. You may also find that you need to change the resource definition templates to suit your requirements.

Important! Resizing a transient log updates the resource definition. It is recommended that if a resource needs a large transient log size, it should be updated individually. If you have a large system image and you set all resource transient logs to the maximum size, there could be system performance degradation and storage issues.

To resize selected transient logs

1. Access the list of system images that contain the resources for which you want to resize logs. For example, enter /RADMIN.I.L to access the list of local system images.

   A System Image List panel appears.

2. Enter **STL** beside the required image.

   A Set TLog Size Specification panel appears.

3. Select the required resources using the Resource Class and Resource Name fields, specify the required size for their logs, and then press F6 (Action).

   A message appears, indicating the number of resource definitions affected.

4. Press F6 (Action).

   The resource definitions are updated with the specified size. If the image is active, the affected logs are also resized.

Note: For active system images, you can also resize the transient logs from the monitors using the SETTLOG command.

Resize Multiple Transient Logs in an Image

If the transient logs for certain resources become full, you can resize them from a resource monitor.

Important! Resizing a transient log updates the resource definition. It is recommended that if a resource needs a large transient log size, it should be updated individually. If you have a large system image and you set all resource transient logs to the maximum size, there could be system performance degradation and storage issues.

To resize multiple transient logs in an image from a resource monitor

1. Enter **SETTLOG** at the Command prompt.

   You are prompted to select the image that contains the resources whose logs you want to resize.

2. Enter **S** beside the required image.

   A Set TLog Size Specification panel appears.

3. Select the required resources using the Resource Class and Resource Name fields, specify the required size for their logs, and then press F6 (Action).

   A message appears, indicating the number of resource definitions affected.

4. Press F6 (Action).

   The resource definitions are updated with the specified size, and the affected logs are resized.

Chapter 5: Setting Up the Initialization File

This section contains the following topics:

Generate an Initialization File

If you are deploying multiple regions, each region must be configured for its local environment. When you have configured your first region, you can build an initialization file from that region and then configure it for use with your other regions. This removes the need to customize each region with Customizer.

The tasks outlined below show how to configure a region from an initialization file. The initialization file is produced from a running region for your product.

To generate an initialization file

1. From the Primary Menu, enter **/CUSTOM**.

   The Customizer panel appears.

2. Select option G - Generate INI Procedure.

   The Customizer : Generate INI Procedure panel appears.

3. Enter the data set name and the member name of the file in the Generate INI File Details section.

   Note: The data set must be in the commands concatenation of the RUNSYSIN member for the region in which it is used.

4. Ensure that the member name and data set name are correct. Enter **YES** in the Replace Member? field if you are replacing an existing member.

5. Press F6 (Action).

   The initialization file is generated.

6. Make a note of the data set and member names and press F6 (Confirm).

   The details are saved.

How You Configure the Initialization File

The initialization file must be configured before it can be used for other regions. You can perform this configuration as follows:

- Configure an individual initialization file for each region.
- Configure a common initialization file for multiple regions.

You can use system variables and static system variables with both of these methods. The variables substitute for the initialization parameters in the INI file.

Configure Individual Initialization Files

You can customize an initialization file generated from one region so that it can be used for another region.

To configure an individual initialization file for each region

1. Use your TSO editing tool to open the initialization file in edit mode.
2. Substitute the parameters in the initialization file with *one* of the following:
   - Hard-coded data set names for the region in which the file is used
   - System variables

   This enables the initialization file to work in regions with different data sets than the region in which it was generated.
3. Save the changes to the initialization file.
4. Copy the initialization file to the region's TESTEXEC or one of the other libraries in the COMMANDS concatenation.
5. Repeat steps 1 to 4 for each initialization file needed.

Note: The region from which the original initialization file was generated should have the same product sets as the destination regions that will use that initialization file.

Configure a Common Initialization File

You can customize an initialization file using variables so that it can be used for multiple regions.

To configure a common initialization file

1. Create a data set that is available to every region to be initialized from the common initialization file, for example, PROD.INIFILES.

2. Add the newly created data set to the COMMANDS concatenation of the RUNSYSIN member to every region to be initialized from the common initialization file.

   Note: RUNSYSIN is located in TESTEXEC.

3. Copy the initialization file generated into the new INIFILES data set.
4. Use your TSO editing tool to open the initialization file in edit mode.
5. Replace the relevant generated variables in the initialization file with the following system variables:

   &ZDSNQLCL

   The local VSAM data set qualifier.

   &ZDSNQSHR

   The shared VSAM data set qualifier.

   &ZACBNAME

   The primary VTAM ACB name used by the region.

   &ZDSNQLNV

   The local non-VSAM data set qualifier.

   &ZDSNQSNV

   The shared non-VSAM data set qualifier.

   &ZNMDID

   The domain identifier.

   &ZNMSUP

   The system user prefix.

6. Replace the relevant generated variables in the initialization file with the z/OS static system symbols as follows:

   &SYSCLONE

   The short name for the system.

   &SYSNAME

   The name of the system.

   &SYSPLEX

   The name of the sysplex.

   &SYSR1

   The IPL VOLSER.

7. Save the changes to the initialization file.

Start Your Region from an Initialization File

The name of the initialization file must be specified by the INIFILE parameter in the RUNSYSIN member.

Updating your RUNSYSIN member causes your region to set its initialization parameters from the initialization file. All Customizer parameter settings are overwritten.

To update your RUNSYSIN member

1. Use a text editor to open your RUNSYSIN member.
2. Insert the line PPREF='INIFILE=*membername*' into your RUNSYSIN member.
3. Save the member.

Chapter 6: Administering a Multisystem Environment

This section contains the following topics:

Multisystem Operation

Your product provides focal point management to support multisystem operation. Management is at a focal point with subordinates and other focal points feeding information to it, as follows:

Focal

Supports full connectivity between multiple regions. Regions linked in this way are known as focal point regions.

When regions are communicating with each other, authorized users can monitor and control all managed resources from any terminal connected to any region.

All focal point regions have the knowledge base synchronized.

Subordinate

Enables you to reduce the amount of traffic in your multisystem environment. You link subordinates to focal point regions that provide central monitoring and control. A subordinate has visibility and control of the resources that belong to the local system image only.

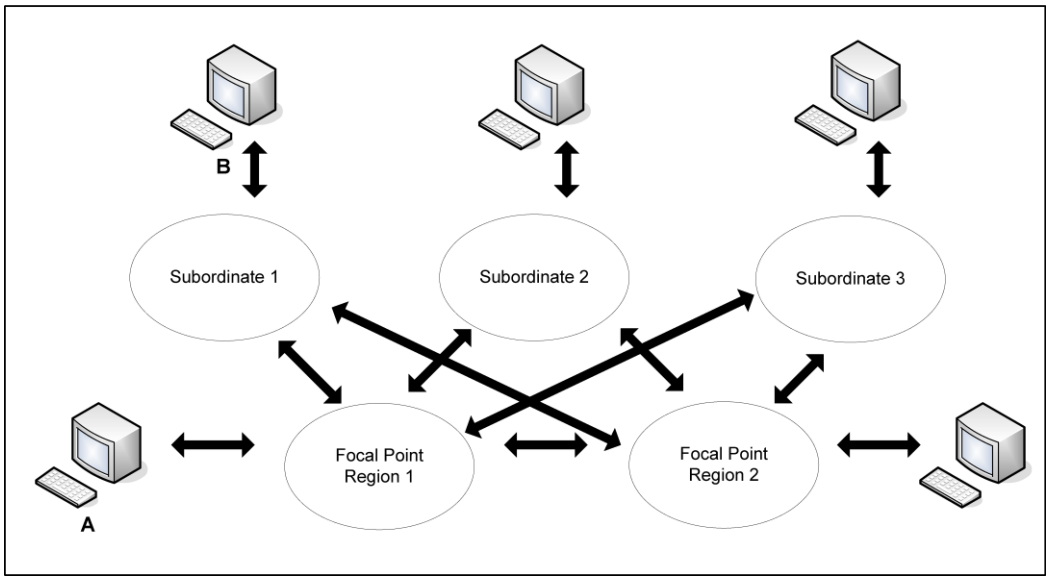
In a multisystem environment, each region runs independently of the other regions. If no communication links are available, each region still provides full monitoring, control, and automation of its own managed resources.

To link a focal point region to another focal point region, or to link a subordinate to a focal point region, you link and synchronize the regions.

Notes:

- You can link as focal points only those regions that are configured for the same products. For a subordinate-focal point link, the products configured in the subordinate region can be a subset of the products configured in the focal point region.
- Subordinate regions assume a system image name that cannot be used for any other region in the multisystem environment. We recommend that you use a unique system image name for subordinate regions running on the same LPAR. If you use express setup, the system image name defaults to the SMF ID.

The following diagram shows an example of a multisystem environment. Logging on to Console A allows visibility to all resources. Logging on to Console B allows visibility to the subordinate system image only.



Notes:

- A focal point region links to all other focal point regions and subordinates.
- A subordinate links to focal point regions but does not link to other subordinates.

Links in a Multisystem Environment

The link established between two regions in a multisystem environment is an INMC link. The link is used to pass knowledge base updates, status change notification, and other information between the two regions. The link can use any combination of the following communications protocols: VTAM, TCP/IP, and EPS. VTAM is the default.

For each region, the MULTISYS parameter group specifies the available communication access methods. If TCP/IP is used, ensure that the SOCKETS parameter group is activated.

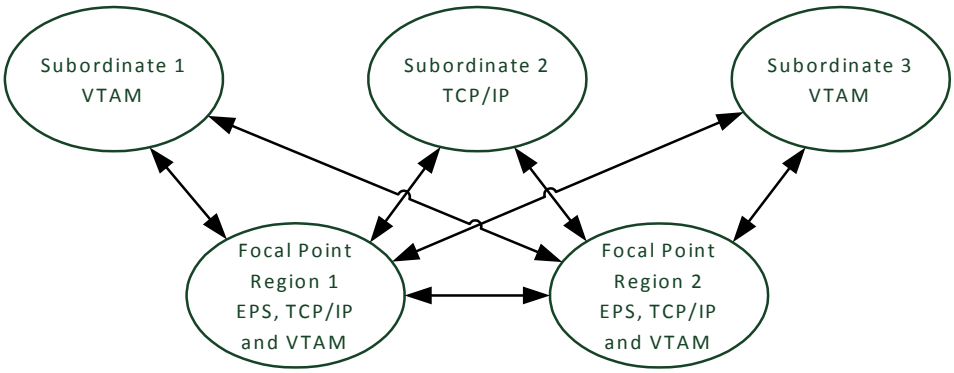
The INMC link between any two regions uses the access methods enabled by *both* regions (that is, the intersection of the two MULTISYS parameter groups). When multiple access methods are enabled, the link can use all these methods. This improves reliability because the link functions when one of the enabled methods is available.

When you plan your multisystem environment, ensure the following:

- All focal point regions must support at least one common type of access method.
- A subordinate region must support an access method that is also supported in all the focal point regions.

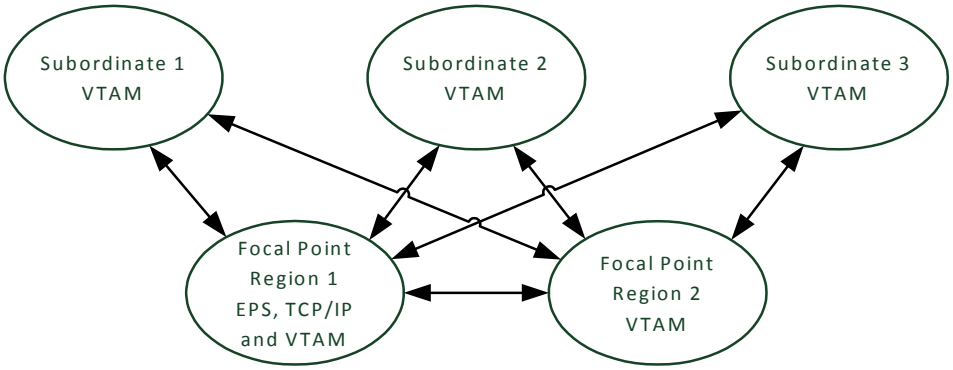
Example: Focal Point Regions Support All Access Methods

This example shows a multisystem link configuration when the focal point regions support ESP, TCP/IP, and VTAM. The subordinate regions can support any one of these access methods.



Example: One Focal Point Region Supports VTAM Only

This example shows a multisystem link configuration when a focal point region supports VTAM only. The subordinate regions must support VTAM.



Multisystem Support in a Sysplex

With the EPS access method, you can use the sysplex cross-system coupling facility (XCF) to implement your multisystem environment.

Notes:

- To support the EPS access method, a SOLVE SSI region must be active in each of the co-operating systems and must be registered to XCF.
- To register the SSI region to XCF, ensure that XCF=YES is set in the SSI parameters member of the SSIPARM data set. This is the default setting at installation.

Multisystem Implementation Considerations

When you implement your multisystem environment, consider the following:

- Ensure that the link requirements (see page 59) are satisfied for the planned multisystem environment.
- When you link two regions, the knowledge base in one region overwrites the knowledge base in the other region. *You must transmit all system images used by the local region to the target focal point region prior to synchronization.*
- You can only link a region to a focal point region. The focal point region can be a stand-alone region or part of a multisystem environment.
- You can only link a stand-alone region into a multisystem environment.

How a Multisystem Environment Is Established

When you install your product, two databases are downloaded. These databases, which can be customized to suit your requirements, are:

- An icon panel database, where icon panel definitions are stored for the graphical monitor
- The RAMDB, where system image, resource, availability map, process, macro, command, and other definitions are stored

Together, these databases form the knowledge base.

Populate these databases with definitions specific to your environment. These definitions can include the system image definitions for any other regions that you want to install in your environment in the future.

As you establish regions, link the new regions to the first region by using the Link Region and Synchronize Database (see page 61) option. When databases are linked, future synchronization is automatic. Changes to the database in one region are sent to the databases in the linked regions that have visibility to those resources and system images.

Note: Synchronization does not apply to the NCL procedures represented by the registered commands and macros. Changes to these NCL procedures are not automatically reflected in the linked regions.

In a multisystem environment, you can monitor and control the resources in all linked regions from a single focal point.

Linked Regions and Database Synchronization

When the first region is created in your environment, two databases are downloaded and can be customized for your environment. Together, these two databases (the Automation Services database and the icon panel library) form the knowledge base.

To build a multisystem environment, you start by linking two regions, and then continue to link in any other regions. The linking process also synchronizes the knowledge bases of these regions.

Notes

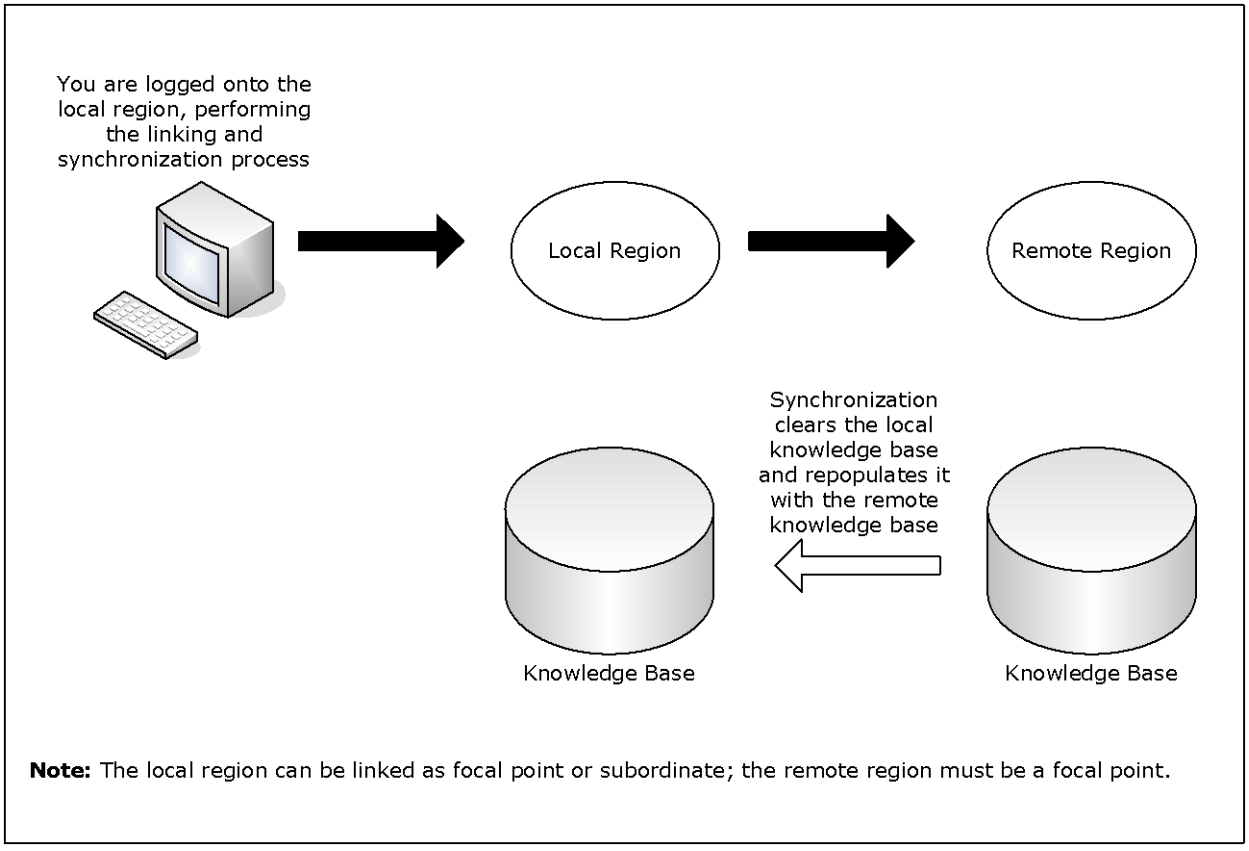
- For linked focal point regions, synchronization is complete and the focal point knowledge bases are identical.
- For linked subordinates, synchronization is complete only to the extent of the relevant definitions in the knowledge base. For example, a subordinate knowledge base does not contain all system images. A subordinate knowledge base contains only those images that represent the environment the subordinate is managing.

When you link two regions, the local region in which you perform the link operation receives the knowledge base from the remote region. This remote region must be a focal point region. When you link a region into an existing multisystem environment, that region must be a stand-alone region.

Important! During the linking and synchronization process, the knowledge base in the local region is overwritten by the knowledge base from the remote focal region. If the local knowledge base has customized definitions that you want to retain, transmit these definitions to the remote knowledge base before you link the regions. Otherwise, the local knowledge base definitions are overwritten and lost.

Note: If the local region terminates during the linking and synchronization process, the local knowledge base can become corrupted and you cannot restart the region. Replace the corrupted knowledge base with your backup, restart the region, and resynchronize the knowledge base. For more information about backups, see the *Reference Guide*.

The following illustration shows the link and synchronization operation.



Note: The local region can be linked as focal point or subordinate; the remote region must be a focal point.

After you link the regions, the knowledge bases are synchronized and remain synchronized. If you change the knowledge base in one region, the changes are propagated to the other regions.

Background User Considerations

When you establish a region, a UAMS background system (BSYS) user ID for that region is automatically defined. The background user ID comprises the four-byte region domain ID, followed by the characters BSYS. To establish fully-functioning communication links between regions, the BSYS user ID of each region must be duplicated in each linked region.

During a link and synchronize procedure, any required BSYS user IDs are defined automatically to UAMS, provided that the following conditions apply:

- You have UAMS maintenance authority on *all* the linked regions.
- The existing multisystem linked regions are active when the request is made.

If either of these conditions does not apply, then any required BSYS user IDs must be defined manually to UAMS. The simplest way to do this is to copy the BSYS user ID for the current region from the UAMS User Definition List and update the user ID. To access the UAMS maintenance functions, enter the **/UAMS** shortcut.

The link and synchronize request is rejected if *both* of the following apply:

- You do not have UAMS maintenance authority in the local or the remote region. (The user ID of the person who requests the link and synchronize procedure must be defined in the local and remote regions.)
- The required BSYS user IDs are not defined in the local or the remote region.

Important! If you use an external security system, you must manually define the BSYS user IDs of the remote systems to your external security system.

Link and Synchronize Regions

Important! Do not add, update, or delete knowledge base records in any linked regions while synchronization is in progress. These changes may not be propagated to the new region. Before you perform synchronization, ensure that you back up the knowledge base.

To link and synchronize regions

1. Log on to the region to synchronize with the source (remote) region.

   The source region contains the knowledge base you want.

2. Enter **/MADMIN** at the prompt.

   The Multi-System Support Menu appears.

3. Select option **SD**.

   This establishes a link between the local region and another region, and updates the knowledge base of the current region.

   The Remote System Identification panel appears.

4. Complete the following fields:

   Primary Name

   Specifies the ACB name of the remote focal point region to which you want to link this region.

   Role in Multi-System Operation

   Specifies whether this region is a focal point region or a subordinate region. A focal point region must satisfy the following conditions:

   - The product sets in all focal point regions match.
   - At least one access method must be available.

   Subordinate System Image Name

   (Optional) If you specified subordinate, specify the name of the system image that is to be used by it.

   Important! Each subordinate is assigned a unique system image name, and it can use an image by that system image name only. When you build your environment for a subordinate, you must build the environment under the system image name specified during the linking operation.

   Subordinate regions are restricted to loading only system images with the name specified here. Different system image versions can be maintained under the system image name.

   Work Dataset

   (Optional) Specifies the VSAM data set to use to reduce the time for synchronization.

   The following fields specify the communication access methods to be used during synchronization. You can select any combination of the access methods; however, you can only select an access method if it is enabled in the MULTISYS parameter group.

   Use VTAM?

   (Optional) Specifies whether to use VTAM for communication.

   Use EPS?

   (Optional) Specifies whether to use EPS for communication.

   TCP/IP Host Name/Addr

   (Optional) Specifies the TCP/IP host name and address of the remote region.

Port Number

(Optional) Specifies the TCP/IP port number of the remote region.

5. Press F6 (Action) to initiate the linking process.

   A confirmation panel appears.

6. Press F6 (Confirm) to initiate region linking and knowledge base synchronization.

   A status panel appears.

   Note: Press F3 (Exit) to exit the status panel at any time without affecting the link and synchronize procedure. If you exit early, note the task number for later reference.

Monitor the Synchronization Procedure

While the synchronization procedure is in progress, the Synchronize Database Status panel is refreshed automatically every 10 seconds. This panel can be refreshed manually at any time by pressing the Enter key.

To check the status of the synchronization

1. From the Multi-System Support Menu, select option L to view the administration task log.
2. Enter S beside the appropriate entry from the log to view the status of the task.

The administration task log may contain up to 50 entries at any given time. Each task is allocated a sequential task number (between 1 and 50) as it commences. When the maximum task number is reached, allocation restarts from one and the oldest status records are overwritten. To delete a completed or failed task from the log, apply the D (Delete) action.

Knowledge Base Synchronization Maintenance

Automation Services maintains synchronization between linked knowledge bases by using a staging file.

When a knowledge base update occurs, information about the update is stored in the staging file as follows:

- For an update in a focal point region, a separate update record is written for each affected linked region.
- For an update in a subordinate region, a single update record is written for a linked focal point region.

A record stays in the staging file until the update is performed successfully in the destined region. If the region is inactive, the record stays in the staging file until the region is started.

Important! If the staging file becomes full, knowledge base synchronization cannot be maintained and the local region is unlinked automatically. A staging file can become full if a remote linked region remains inactive for an extended period of time. If an extended downtime is planned for a linked region, unlink the remote region before inactivation.

Display Linked Regions

To list the linked regions in your multisystem environment, enter **/LISTREG** at the prompt.

The Linked Regions panel displays the ACB names, the mode these regions are linked in, and a brief description of the linked regions. The panel also displays the status of the data flow traffic managers.

Press F11 (Right) to scroll right to display more information.

Unlink Regions

You may want to unlink a region from the other regions in a multisystem environment (for example, for maintenance purposes). If a region is no longer of use and you want to remove it, ensure that you unlink it first. An unlinked region is a stand-alone region.

To unlink a region

1. Log on to the region you want to unlink and enter **/MADMIN.U** at the prompt.

   The Confirm Unlink Panel appears.

   Note: To cancel the unlinking procedure, press F12 (Cancel) now.

2. Press Enter to proceed with the unlinking procedure.

To relink a region, link that region with one of the regions in the multisystem environment.

Transmission of Records

You can transmit, that is, copy knowledge base records from the local region to a remote region that is not linked to it.

You cannot transmit a system image to a region in which the image is currently loaded. You cannot transmit and replace a rule set when the rule set is currently loaded in the remote (target) region.

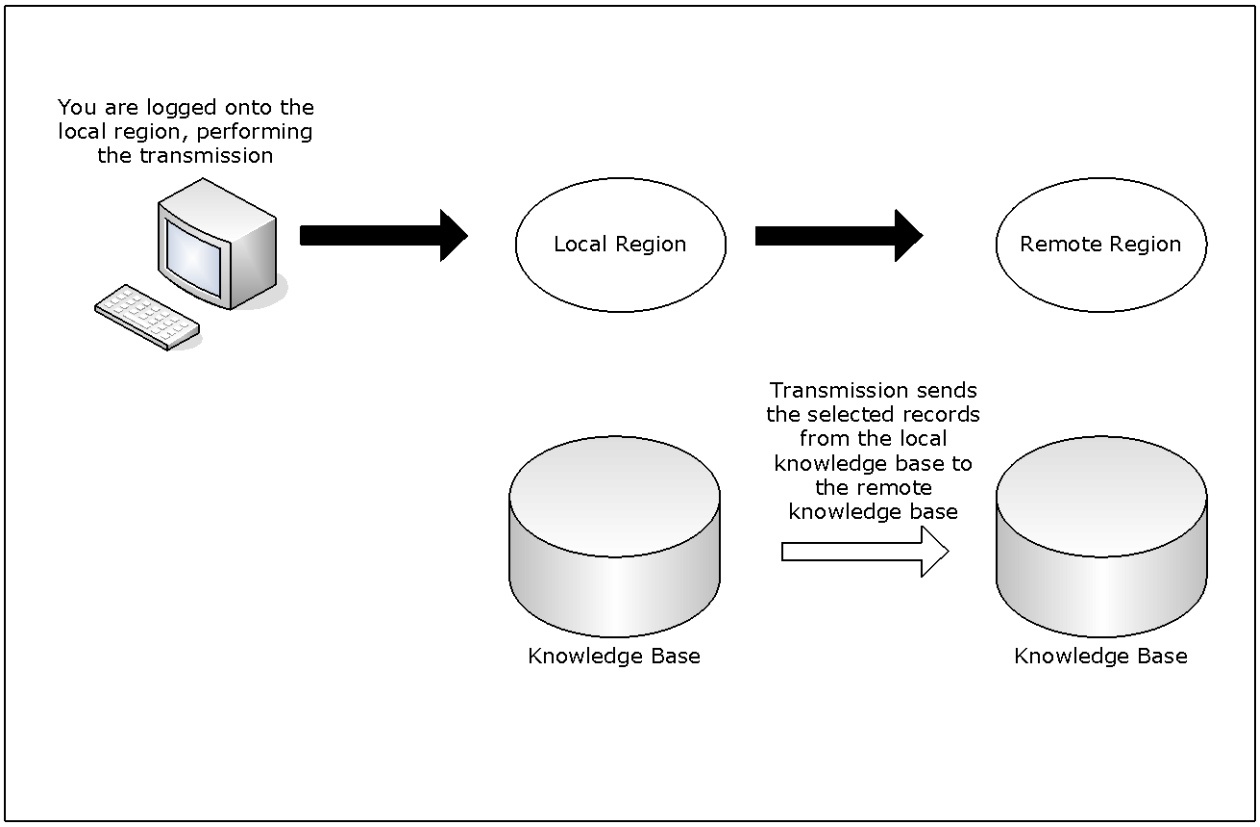
By specifying the appropriate transmission mode on the Remote System Identification panel, you can specify how to update the records in the remote region.

The following transmission modes are available:

- Replace (R) deletes any existing remote records, and then transmits the local records.
- Overlay (O) replaces existing remote records with the same name, adds records that do not exist, but does not delete any remote records.
- Merge (M) adds records that do not exist, but does not affect existing records in the remote knowledge base.

Transmit Records

The following illustration shows the transmit operation.



To transmit knowledge base records

1. Log on to the region from which you want to transmit the records.
2. Enter **/MADMIN** at the prompt.

   The Multi-System Support Menu appears.
3. Specify the option you want at the prompt and press Enter.

   A Remote System Identification panel appears.
4. Specify the ACB name (primary name) of the region to which you want to transmit records.

   If you specified the TI, TS, or TR option, go to Step 5. If you specified any other transmission options, go to Step 6.

5. Complete the following fields:

   System Name

   Specifies the name of the system to transmit. Applies to option TI only.

   Version

   Specifies the version of the system to transmit. Applies to options TI and TS only.

   Ruleset Name

   Specifies the name of the rule set to transmit. Applies to option TR only.

6. Do *one* of the following:

   - If you want to replace a set of records or all elements of a component, enter REPLACE in the Transmission Mode field.
   - If you want to update a region by adding new records without updating existing records, enter MERGE in the Transmission Mode field.
   - If you want to update a region by adding new records and updating existing records, enter OVERLAY in the Transmission Mode field.

7. Specify the communication access methods to use for transmitting the selected records. You can enable any combination of the access methods.

8. Press F6 (Action) to select the specified option.

   If a selection list appears, go to Step 9. If the Confirm Transmit panel appears, go to Step 11.

9. Do *one* of the following:

   - If you selected option TC with a transmission mode of REPLACE, enter **S** next to the categories that you want to transmit.
   - If you selected option TC with a transmission mode of MERGE or OVERLAY, enter **S** next to the categories that you want to transmit.
     To select specific definitions in a category for transmission, perform the following steps:

     a. Enter **L** (List) next to the category to list the definitions.

     b. Enter **S** next to the definitions to transmit.

   - If you selected other transmission options with a transmission mode of MERGE or OVERLAY, take *one* of the following actions:

     - To transmit all definitions, press F4 (All).
     - To transmit specific definitions, enter **S** next to the definitions that you want to transmit.

10. Press F6 (Transmit).

    A Confirm Transmit panel appears.

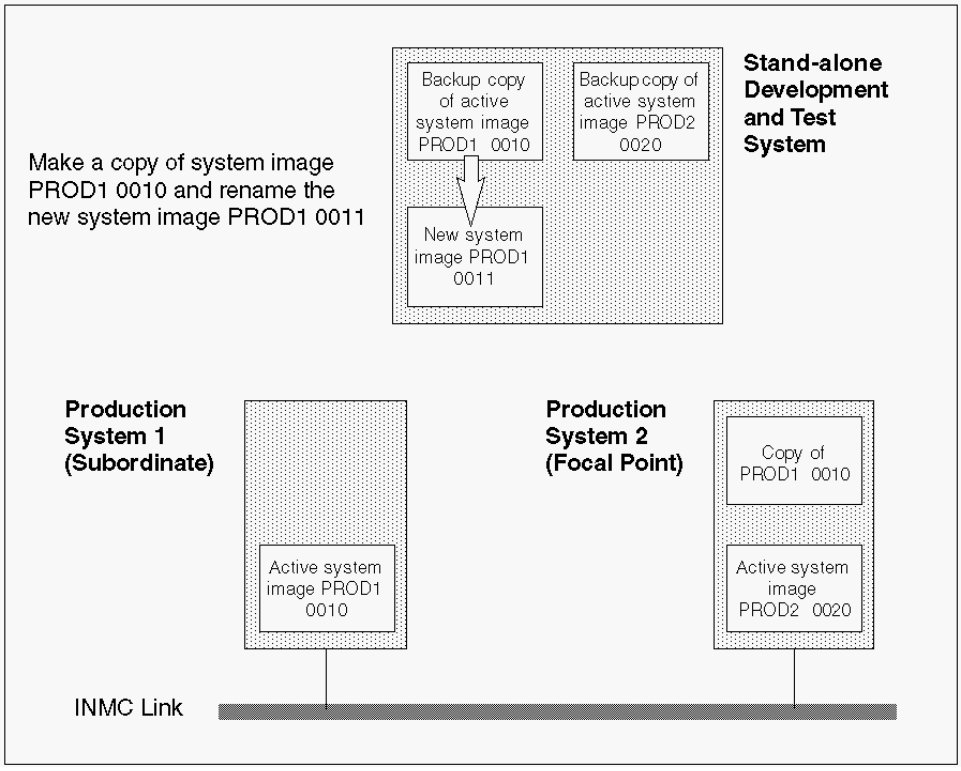
11. Press Enter to confirm transmission.

    A status panel appears, showing the progress of the transmission.

    Note: If you exit the status panel, you can check the status of the task by viewing the administration task log. Before you exit, note the task number for future reference.

Update and Test a System Image in Isolation

A typical multisystem setup consists of a group of linked production systems (or regions), and a stand-alone development and test system, as shown in the following illustration:



The development and test system has copies of the active system images of all your production systems.

To update and test a current system image in a stand-alone development region

1. Transmit a copy of the active system image that you want to update to the stand-alone development region.

   The development region has a copy of the image you want to update

2. Copy this system image to a new version.

   The region creates another copy but at a new version.

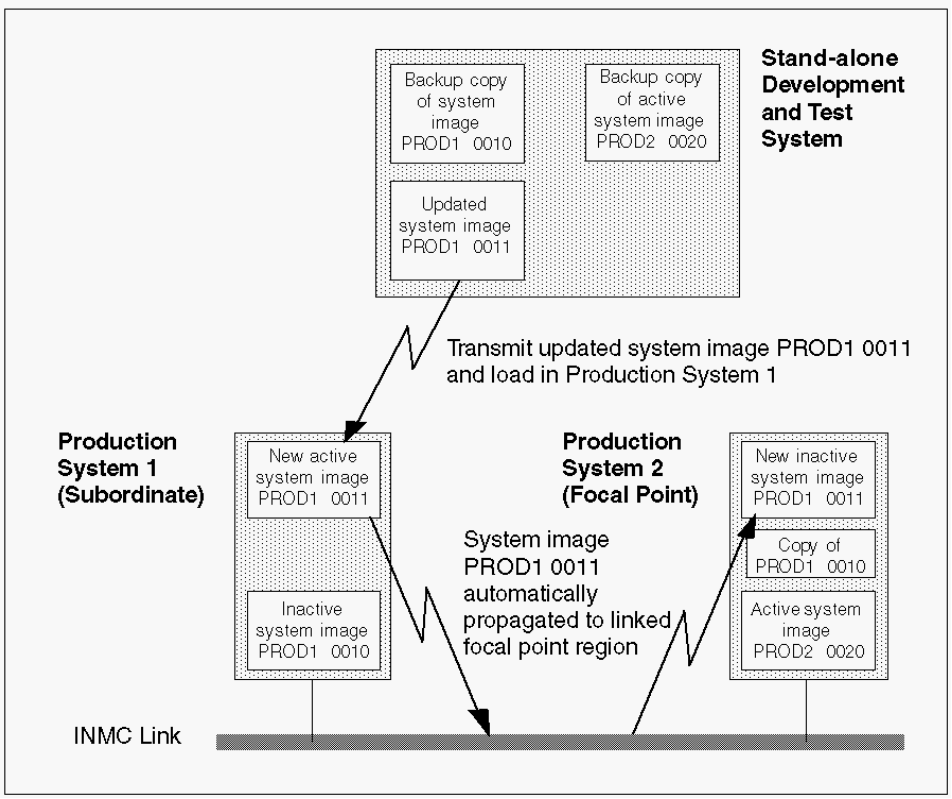
3. Make the required changes to the new version of the system image and test these changes.

4. Transmit the updated system image to the production region where it originated when you are satisfied with the updates you have made.

   The new system image is automatically copied to all affected linked regions.

5. Load the new system image.

   The currently active system image is replaced with the updated version.



Chapter 7: Management of CICS Resource Operations

This section contains the following topics:

How CICS Resource Operations Can Be Managed

You can use the following methods to help you manage the operation of CICS resources:

- Create resource and service definitions to provide automated operation for, and visibility of, important resources.
- Create message rules that generate alerts to warn you of problems with resource operation.
- Create message profiles to monitor the CICS message flow on your systems.

CICS Started Tasks and Jobs

CICS transactions, files, links, and databases are owned by CICS regions. To enable the CA SOLVE:Operations Automation for CICS region to manage these CICS-owned resources, you should create resource definitions for their owners.

Depending on whether a CICS region is established by a started task or a job, you create a started task or job resource definition to manage it.

CICS Started Task Templates

Templates are provided to help you define CICS started tasks in the knowledge base. The templates use the following facilities to help you manage a CICS started task:

- S *cics_region_name* system command that activates the CICS started task.
- F *cics_region_name*,CEMT PER SHUT system command that inactivates the CICS started task.
- C *cics_region_name* system command that cancels the CICS started task.
- A default display method that checks the existence of the CICS started task address space ID, and sets the actual state to either ACTIVE or INACTIVE.
- System and CICS messages that change the actual state of the task. For messages that indicate a problem, you can include a process that takes certain actions (for example, to record the problem in your problem management application).

CICS Job Templates

No templates are provided for CICS job resource definitions. However, you can copy the CICS started task templates to the JOB class and update them to satisfy your CICS job operations requirements.

CICS Transactions

You can create a CICS transaction resource definition to manage a transaction.

For a transaction that is owned by several CICS regions, you can create a definition for each of the regions.

For a remote transaction, you can create the following definitions:

- CICS link resource definitions that manage the link between the CICS region and the CICS region that owns the transaction
- A CICS transaction resource definition for the CICS region that owns the transaction

Transaction Templates

Templates are provided to help you define CICS transaction resources in the knowledge base. The templates contain the following information to help you manage a transaction:

- CICSCMD *cics_region_name* SET TRAN(*transaction_name*) ENA command that enables the transaction when the resource is activated
- CICSCMD *cics_region_name* SET TRAN(*transaction_name*) DIS command that disables the transaction when the resource is inactivated
- CICSCMD *cics_region_name* INQ TRAN(*transaction_name*) command that inquires about the status of the transaction
- Transaction abnormal termination messages that change the actual state of the resource to indicate a problem—you can include a process that takes certain actions when an abnormal termination occurs (for example, to record the problem in your problem management application)

CICS Files

CICS enables online access to files.

You can create a CICS file resource definition to manage a file.

For a file that is shared between several CICS regions, you can create a definition for each of the regions.

For a remote file, you can create the following definitions:

- CICS link resource definitions that manage the link between the CICS region and the CICS region that owns the file
- A CICS file resource definition for the CICS region that owns the file

File Templates

Templates are provided to help you define CICS BDAM and VSAM file resources in the knowledge base. The templates contain the following information to help you manage a file:

- CICSCMD *cics_region_name* SET FILE(*file_name*) ENA command that enables the file when the resource is activated
- CICSCMD *cics_region_name* SET FILE(*file_name*) DIS command that disables the file when the resource is inactivated
- CICSCMD *cics_region_name* INQ FILE(*file_name*) command that inquires about the status of the file
- File error messages that change the actual state of the resource to indicate a problem—you can include a process that takes certain actions when an error occurs (for example, to record the problem in your problem management application)

CICS Links

For each CICS link, you can create two definitions, one for each end of the link.

You can combine link resource definitions with database, file, and transaction resource definitions to manage remote databases, files, and transactions.

Templates are provided to help you define the following types of CICS link resources in the knowledge base:

- ISC links
- MRO links

ISC Link Templates

The ISC link templates contain the following information to help you manage an ISC link:

- ACTVTLNK process that acquires a session on the link when the resource is activated
- INAVTLNK process that releases the connection when the resource is inactivated
- CICSCMD *cics_region_name* INQ CONN(*connection_name*) command that inquires about the status of the connection
- Link messages that change the actual state of the resource—you can include a process that takes certain actions when an error occurs (for example, to record the problem in your problem management application)

MRO Link Templates

The MRO templates contain the following information to help you manage an MRO link:

- ACTLINK process that puts the link into service when the resource is activated
- CICSCMD *cics_region_name* SET CONN(*connection_name*) OUT command that puts the link out of service when the resource is inactivated
- CICSCMD *cics_region_name* INQ CONN(*connection_name*) command that inquires about the status of the connection
- Link messages that change the actual state of the resource—you can include a process that takes certain actions when an error occurs (for example, to record the problem in your problem management application)

CICS Databases

CICS enables online access to databases. The commonly used databases are ADABAS, DB2, and Information Management System (IMS) DL/I. CICSCMD commands are provided to enable you to inquire about these databases.

You can define resource definitions in the knowledge base to manage these databases.

Templates are provided to help you define CICS DL/I database resources in the knowledge base. The templates contain messages that reflect the actual state of the resource. As supplied, the templates provide monitoring functions only.

Management of CICS Resources by Using an SDO Approach

The service-driven operations (SDO) approach to the management of your information system (IS) resources enables you to manage directly the business functions you are providing to your users. You use services to represent these business functions.

After you have defined the resources you want to manage, you can group them by business functions in ServiceView service definitions. These definitions enable you to specify the availability requirements of the defined services, and enable the services to be monitored and controlled.

Note: You can define and manage services from focal point regions only. Services are not visible in subordinate regions, but you can include resources managed by a subordinate region in a service.

Management of CICS Resources by EventView Message Rules and Alerts

By using EventView message rules, you can generate alerts for CICS resources that have problems.

By using message rules, you can extend the class of CICS resources that are managed. During the setup of your CA SOLVE:Operations Automation for CICS region, a filter is defined through the CICSCNTL parameter group to determine which unsolicited CICS messages are visible to the region. For example, if the filter passes CICS terminal messages, you can use message rules to monitor these messages and generate alerts when problems occur.

Note: For information about how to specify the filter criteria, see the *Getting Started* guide.

Management of CICS Resources by Messages

By using consolidated console message profiles, you can monitor CICS messages that are visible to connected regions from a single terminal.

Note: The consolidated console is fully functional in focal point regions only. In subordinate regions, only local message traffic is visible.

The consolidated console provides a purely monitoring facility. You should use resource definitions, message rules, and alerts to manage the operations of your CICS resources.

Chapter 8: Implementing a System Image of CICS Resources

This section contains the following topics:

Supported CICS Resource Definitions

The following table lists the CICS resource definition classes supported by CA SOLVE:Operations Automation for CICS. Some resource definition templates are supplied for different versions of CICS, as indicated by the *n* at the end of a template name.

CICS Resource Definition	Class Name	Class Number	Templates
Started task	STC	02	CICS
Job	JOB	19	
Database	CICDB	06	DLIDBCTL (Database Control) DLILOCAL (local DL/I)
File	CICFIL	05	BDAM (BDAM file) VSAM (VSAM file)
Link	CICLNK	07	ISC (ISC link) MROV*n* (MRO link)
Transaction	CICTRN	04	TRANV*n*

System Images

A system image specifies the operations environment to be managed by a region. Before you can define the resources to be managed, you must define a system image.

You define system images from the System Image List panel. To access the list, enter **/RADMIN.I** from any panel.

When you have a system image, you can define resources to and associate EventView rules with it.

Note: If you are running CA SOLVE:Operations Automation for CICS and CA SOLVE:Operations Automation in the same region, you can define a system image by using the AutoAssist Express Setup Facility. The facility creates a system image and adds resources on the system to the image. These resources include CICS started tasks and jobs.

You can define many system images, but you can make only one of them active in a region. Furthermore, an image can be active only on its home system. The region manages the operations environment specified by this active image. To make an image active, you load it in the region. To enable an image to be loaded every time the region starts up, identify the image in the AUTOIDS parameter group. (To access the list of parameter groups, enter **/PARMS** from any panel.)

Note: For detailed information about how to work with system images, see the *Reference Guide* and the online help.

How the Auto Populate Facility Works

The AutoAssist Auto Populate Facility helps you define selected resources to a system image quickly by using templates. You can use the facility to define CICS jobs, started tasks, files, links, and transactions. You need to define database resources accessible by CICS manually.

Also, CA SOLVE:Operations Automation for CICS does not provide templates for CICS job resource definitions. If you want to use this facility for jobs, you need to first define job templates. You can define them by basing them on the supplied CICS started task templates.

The facility does not overwrite existing definitions.

Important! Populating an active system image can degrade system response time. Try to populate an image before you load it in the region. If you must populate a large number of resource definitions in an active image, do it when the system is not busy.

The Auto Populate Facility operates as follows:

1. It uses the parameters specified on the CICS Auto Populate menu to build the list of resources.
2. From the list, you select the resources that are to be defined.
3. It creates the definitions by using the assigned templates.

Template System Selection

More than one template system image exists in the knowledge base. Only templates in the currently active template system are available to be used to define resources. The active template system is specified in the OPSYSIDS parameter group.

Populate a System Image with CICS Resources

You can use the Auto Populate Facility to add supported CICS resources to a system image.

To populate a system image with CICS resources that are defined in a CICS region

1. Enter **/RADMIN.CA** from any panel.

   The CICS Auto Populate Menu appears.

```
SOLVPROD---------- AutoAssist : CICS Auto Populate Menu ---------------$RM012
Select Option ===>

   J   - Build CICS Jobs
   S   - Build CICS Started Tasks
   F   - Build CICS Files
   L   - Build CICS Links
   T   - Build CICS Transactions
   X   - Exit

System Name .........+ SOLV____     ( Required )
Version .............+ 0001         ( Required )
CICS Jobname ......... ________     ( Required F L T )
CICS Resource Class ... _____       ( Required F L T, specify STC or JOB )
Template ............+ ___________ ( Required )
Resource Mask ......... ________________ ( Optional )
```

2. Type the option code for the class of resources you want to define at the Select Option prompt, and complete the following fields:

   - Identify the system image you want to populate in the System Name and Version fields.
   - (CICS-owned resources only) Identify the CICS job or started task that owns the resources in the CICS Jobname and the CICS Resource Class fields.
   - Identify the template you want to use in the Template field.
   - (Optional) Use the Resource Mask field to restrict the list of resources that will be displayed for which you can build definitions.

   Press Enter to display the list of resources.

3. Type **S** beside the resources you want to define to the system image. If you want to select all the displayed resources, enter **ALL S** at the Command prompt.

   (Optional) Adjust your template assignments for the individual resources if necessary. For example, you may want to use a special template for a particular resource.

   Press F6 (Action) to populate the system image.

   The selected resources are defined in the system image.

Identify Remote Owner of an Autopopulated CICS Link Resource

If you define a CICS link resource by using the Auto Populate Facility, you need to identify the remote owner in the definition.

To identify the remote owner of an autopopulated CICS link resource

1. Enter **/RADMIN.R.CICLNK** to list the defined link resources.

   The CICS Link List panel appears.

2. Enter **U** beside the name of the required resource definition.

   The definition opens for update.

3. Enter **1** at the Command prompt.

   The CICS Link General Description panel appears.

4. Identify the remote owner in the Remote Region Name, Class, and SMFID fields.

   Press F3 (File).

   The updated definition is saved.

Define a Resource Using the Resource Learning Feature

For a resource with no template (for example, a resource of the USRCLS class), use the AutoAssist resource learning feature. The feature helps you define the resource to the knowledge base. The feature guides you through the following:

- Filling in the fields in the resource definition
- Learning the messages for the resource
- Specifying relationships with other resources
- Creating a template from the resource definition if necessary

The feature defines the resource in the local active system image. If the image undergoes a loading or shutdown operation while resource learning is in progress, the learning session aborts.

The feature comprises a sequence of panels that prompt you for information. Some fields can also be primed with initial values. If you intend to build a template from the definition, supply only the common data initially. You can refine the definition later for the specific resource. The feature guides you through the following steps:

Note: You can use the F7 (Backward) function key to backtrack through the steps. If you change your mind and do not want to create the resource definition, press F12 (Cancel) before you press F8 (Continue) at Step 9.

To define a resource in the local active system image using the resource learning feature

1. Stop the resource you want to define.

Important! During the learning session, you actually start and stop the resource to learn about the relevant messages.

2. Enter **/RADMIN.AD.R** to access the resource learning feature.

   The Resource Learning panel appears.

3. Specify the class and name of the resource in the Resource Class and Resource Name fields. Press F8 (Continue).

4. Specify the resource type, and provide descriptions for the resource in the Description fields. Press F8 (Continue).

   Note: The operation mode is set to MANUAL for resource learning.

   A panel prompts you to specify the command to start the resource.

5. Define the activation details for the resource:

   a. Specify the command to start the resource in the Activation Command field. Press F8 (Continue) to start the resource.

      A Message List panel appears, showing the messages received during the starting process.

   b. Select the message that indicates that the resource has become active to proceed to the next step.

      A panel appears with the selected message.

   c. Replace any variable data such as the time of day by the asterisk (*) wildcard character. Each embedded * represents a single character. Specify only enough message text to identify the message positively. Press F8 (Continue).

      A panel prompts you to specify the command to display the status of the resource.

   d. Specify the command to display the status of the resource in the Status Command field. Press F8 (Continue) to issue the command.

      A Message List panel appears, showing the messages received during the displaying process.

   e. Select the message that indicates that the resource is active to proceed to the next step.

      A panel appears with the selected message.

   f. Replace any variable data such as the time of day by the asterisk (*) wildcard character. Each embedded * represents a single character. Specify only enough message text to identify the message positively. Press F8 (Continue).

      A panel prompts you to specify the command to stop the resource.

6. Define the inactivation details for the resource:

   a. Specify the command to stop the resource in the Inactivation Command field. Press F8 (Continue) to stop the resource.

      A Message List panel appears, displaying the messages received during the stopping process.

   b. Select the message that indicates that the resource has become inactive to proceed to the next step.

      A panel appears with the selected message.

   c. Replace any variable data such as the time of day by the asterisk (*) wildcard character. Each embedded * represents a single character. Specify only enough message text to identify the message positively. Press F8 (Continue).

      The display command specified in Step 5d is issued for the stopped resource. A Message List panel appears, displaying the responses.

   d. Select the message that indicates that the resource is inactive to proceed to the next step.

      A panel appears with the selected message.

   e. Replace any variable data such as the time of day by the asterisk (*) wildcard character. Each embedded * represents a single character. Specify only enough message text to identify the message positively. Press F8 (Continue).

      A panel prompts you to specify the command to force the resource to stop.

7. Specify the command that can force the resource to stop in the Force Inactivation Command field. Press F8 (Continue).

   A panel appears for you to enter messages that indicate the various resource states. The previously selected messages that indicate the ACTIVE and INACTIVE actual states are automatically displayed on the panel.

8. Specify the status display messages in the Message fields where applicable. Replace any variable data such as the time of day by the asterisk (*) wildcard character. Each embedded * represents a single character. Specify only enough message text to identify the message positively. Press F8 (Continue).

   A panel appears for you to define relationships between this resource and other resources.

9. Do *one* of the following:

   - Press F5 (Related), and then press F11 (Relate) to define the relationships (see page 91). After you define the relationships, Press F8 (Continue).
   - Press F8 (Continue) if you do not want to define relationships.

   The resource definition is saved. A panel appears for you to refine the created definition. For example, you want to change the operation mode.

10. Do *one* of the following:
    - Press F6 (Update) to update the definition. After you update the definition, press F8 (Continue).

      Note: If you intend to create a template, do not enter any data that is specific to this resource.
    - Press F8 (Continue) if you do not want to update the definition.

    A panel appears. The panel lets you create a template from the definition.
11. (Optional) Specify the name of the template in the Template Name field and describe the template in the Template Description field, and press F6 (Build).

    The template is created. The definition appears for you to review and update.
12. Press F3 (Exit) to finish the session.

Note: After you exit from the resource learning feature, you can use the Resources option on the Definition Menu to access the definition for further refinement.

Access CICS Resource Definitions for Maintenance

The CICS resource definition maintenance facility enables you to perform the following tasks on individual definitions:

- You can review and update the definitions created by autopopulation. For example, you might want to attach an availability map to the definition.
- You can create additional definitions in a system image.
- You can define explicit operational relationships between the resources described by the definitions.
- You can delete obsolete definitions.

Note: For general information about resource definitions, see the *Reference Guide* and the online help.

Access CICS resource definitions from any panel as follows:

- To access the CICS job definitions, enter **/RADMIN.R.JOB**.
- To access the CICS started task definitions, enter **/RADMIN.R.STC**.
- To access the database definitions, enter **/RADMIN.R.CICDB**.
- To access the file definitions, enter **/RADMIN.R.CICFIL**.
- To access the link definitions, enter **/RADMIN.R.CICLNK**.
- To access the transaction definitions, enter **/RADMIN.R.CICTRN**.

Naming of CICS-owned Resource Definitions

A CICS-owned resource that is defined in the knowledge base is known as *owner_name.resource_name*.

Use the following naming convention when completing the Name field on the General Description panel:

For a...	Use the name of the...
DL/I database	database.
file	CICS FILE resource.
link between CICS regions	CICS CONNECTION resource.
transaction	CICS TRANSACTION resource.

Resource Templates

The Auto Populate Facility enables you to define most supported CICS resources semi-automatically by using templates.

CA SOLVE:Operations Automation for CICS, however, does not provide templates for CICS job resource definitions. To use the facility, you need to first define these templates.

Also, the facility does not support database resources, which you need to define manually.

Define a CICS Job Template

The easiest way to define the CICS job template is to base it on a supplied CICS started task template.

To define a CICS job template

1. Enter **/RADMIN.T.R.STC** to access the list of started task templates.

   The Started Task List panel appears.

2. Enter **CCC** beside the CICS started task template you want to copy.

   The Resource Class List panel appears.

3. Enter **S** beside the JOB class.

   The Batch Job General Description panel appears.

4. Review and update the new template to suit the JOB class, and then press F3 (File).

   A CICS job template is defined in the template system.

Default Job Library Variable

You can use the &&000$RMRJCL global variable to set the default job library.

The following template definition panel uses the global variable:

```
SOLVPROD----- ResourceView : JOB CICSESA4 Activation Details -----$TEMPLAT-0099
Command ===>                                                    Function=BROWSE

. Activation Processing ---------------------------------------------------.
| Submit JCL Member .. &&000$RMRJCL(&ZRMDBNAME)                             |
```

The value of the variable is specified in the AUTOFILES parameter group. You can access the list of parameter groups from any panel by entering **/PARMS**. Select the group, and the value is specified in the Default JCL Library Name field on the second page.

Database Template Considerations

Sample templates help you provide monitoring functions for local DL/I databases and DL/I databases controlled by Database Control (DBCTL) regions.

When you use the templates to define a DL/I database resource that is accessible to CICS, you should review the definition and replace the question marks by the program specification block (PSB) name. Review, in particular, the Display and Heartbeat Details panel and the extended filters for the specified messages.

Operations Commands

You can specify the following commands in the command fields of a CICS-owned resource definition:

- CICSCMD commands that perform activation, inactivation, and display functions
- System commands

CICS Started Task Activation Process

The CICSTART special activation process is supplied for CICS started tasks. The process performs the following functions:

- Issues a WTOR message to enable you to specify whether a cold or a warm start is to be performed.
- Uses the CICS START=AUTO option if no reply is received within a specified time interval.

Automation Log Considerations

A CICS resource definition is based on the standard ResourceView resource definition. However, the setting of the Log All System Msgs field on the Automation Log Details panel is irrelevant for CICS-owned resources such as transactions. In those definitions, only messages that match the message rules are logged.

How Parent-Child Relationships Between Resources Work

The region uses the relationships between resources during automated system startup and shutdown to determine the order in which to start or stop resources.

There are two parties to a relationship: parent and child.

During automated operation, a parent must be active before dependent resources (its children) may be started. For example, JES must be active before CICS may be started.

Similarly, a parent cannot be stopped automatically unless all its children are inactive.

Relate a Resource with Other Resources

When you have defined the resources in a system image, you can define the relationships between the resources. You can define relationships between resources in the local system image, and between resources in a local system image and a shared system image with the same home system.

Example: Make a DB2 Resource the Child of a CICSD Resource

You want to make a DB2 started task the child of a CICSD resource.

To define the relationship

1. Enter **R** beside the CICSD resource in the resource list.

   The Existing Relationships List panel displays. The following panel shows that JES2 is defined as a parent of CICSD:

```
SOLVPROD----- ResourceView : CICSD Existing Relationships List ---EASTTEST-0001
Command ===>                                                  Scroll ===> 10

              R=Existing Related UP=Unlink Primary UA=Unlink Alternate U=Unlink
    Name              System Image Class  Primary Alternate
    JES2              EASTTEST0001 STC    PARENT
    CICSD             EASTTEST0001 STC    SELF    SELF
    **END**
```

2. Press F11 (Relate).

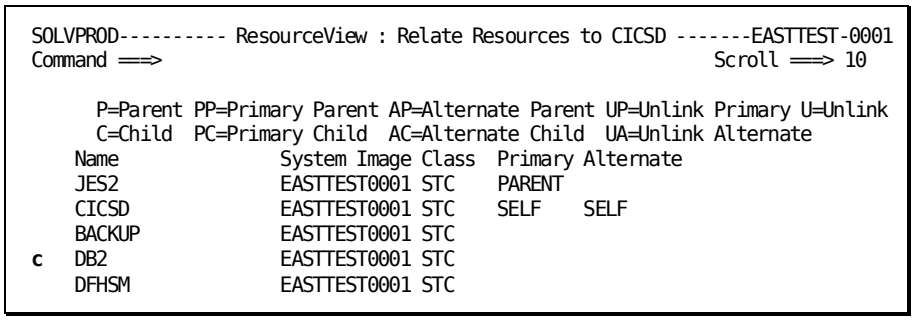
   A selection list of the resource classes is displayed.

3. Enter **S** beside the STC class.

```
SOLVPROD-------- Automation Services : Resource Class List --------------------
Command ===>                                                  Scroll ===> 10

                     Use 'S' to select class(es) from which to relate resources
    Class    Description
    DASD     Direct Access Device
    INIT     JES Initiator
    INTNL    Automation Services Internal
    JES      Job Entry Subsystem
    JOB      Batch Job
    LINE     JES Line
    PRT      Printer
    SPOOL    JES Spool
s   STC      Started Task
    SVC      Service
    TAPE     Tape or Cartridge
    USRCLS   User Class Resource
```

   A list of all started tasks in the system appears.

4. Enter **C** next to a DB2 resource to define it as a child of CICSD.



The updated list of resource relationships appears. The relationships show CICSD as the parent and DB2 as the child.

Primary and Alternate Relationships

For each resource, you can define a primary relationship and an alternate relationship with other resources. During startup and operation, the region uses the primary relationship. During shutdown, the region uses the relationship specified in the OPSYSIDS parameter group. The default is the primary relationship.

Set the Default System Image

The region loads a system image during initialization. The system image loaded when the region is initialized is controlled by the AUTOIDS parameter group.

To set up the system image to load on restart

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups panel appears.

2. Enter **U** beside the $RM AUTOIDS parameter group.

   The AUTOIDS - Automation Identifiers panel appears.

3. Enter **?** in the System Image Name field.

   The ResourceView : System Image List panel appears.

4. Select the System Image that you want to load at restart and press F6 (Action).

   Important! F6 (Action) replaces the currently-loaded system image. If you do not want to load the system image now, skip this step.

5. Press F3 (File).

   The system image loads each time the region starts.

Load a System Image

You define the operations requirements of the resources to be managed on a system in a system image. You must create a system image definition before you can define the resources you want to manage.

The region loads a system image during region initialization. During operation, you may need to change the system image by loading another image.

Note: When you request to load a system image, the $RMEXSTR exit NCL procedure is executed before the starting process. This procedure may be customized at your site to perform any required tasks before any automated resources are started. The starting process cannot proceed if the exit sets a non-zero return code.

For products that use desired state automation, resources are started according to any relationships defined in the system image, and subject to resource availability.

To load a system image

1. Enter **/RADMIN.I** at the prompt.

   The ResourceView : System Image List appears.

2. Enter **L** beside the system image that you want to load.

   The LOAD Command Parameter Specification panel appears.

3. Complete the following fields:

   SysName to be Loaded

   Enter **?** and select a system image from the displayed prompt list.

   Global Automation Mode

   Specify the global operation mode for your system image.

   Perform COLD Start?

   If the Checkpoint Restart Status field is set to ACTIVE, you can enter NO in the Perform COLD Start? field to specify a warm load.

4. Press F6 (Action) to load the system image.

   The Command Confirmation panel appears.

5. Enter **CONFIRM** in the Response field.

   The system image is loaded.

Important! Resources that are monitored by the region are defined to the system image. Loading a system image affects all users of this region and may influence the resources in the system image.

Checkpoint Restart Function

The checkpoint restart function lets you preserve manual overrides across system restarts.

When checkpoint restart is active, any override placed on a resource is stored in the resource definition as checkpoint data. This checkpoint data is applied automatically to the resource when you load the system image with a Warm Start, restoring previously placed overrides.

When checkpoint restart is inactive, any override placed on a resource is not stored as checkpoint data; however, previously stored data is retained. With checkpoint restart inactive, a Warm Start does not apply any stored checkpoint data.

Note: Setting checkpoint restart inactive does not clear the stored checkpoint data. If you later set checkpoint restart to active, then a Warm Start applies the previously stored checkpoint data.

If you no longer want to restore previously placed overrides, load the system image with a Cold Start. All checkpoint data is cleared from the resource definitions, and the resources are loaded without overrides.

Cold Start also clears checkpoint data from the following resources:

- Resources in shared system images (both active and inactive) that satisfy the following conditions:
  - The resource has the local system as the home system.
  - The resource is not active on another system.
- Resources in z/VM system images where the z/VM system image has the local system as the home system

Note: The local system is where the system image is being loaded.

Global Operation Mode

The global operation mode determines the mode of operation for a loaded (active) system image. Your region can run in a global operation mode of AUTOMATED or MANUAL.

As the name global suggests, the setting of the global operation mode limits the control of all resources defined to a system image. For example, if the global operation mode is MANUAL and the resource operation mode is AUTOMATED, the resource can run in the MANUAL operation mode only. If the global mode is changed to AUTOMATED, then that resource runs in its assigned mode.

You can issue a GLOBAL command from the resource monitor to set the global operation mode. For example, you have finished testing a system image on a development system in the MANUAL operation mode and you want to change the global operation mode to AUTOMATED. If you are experiencing severe problems on a production system, you can change the global operation mode from AUTOMATED to MANUAL.

Important! Changing the global operation mode affects all resources that are defined in the loaded system image. If you are changing the mode from MANUAL to AUTOMATED, verify that all resources are defined correctly before the change.

Set Global Operation Mode

To set the global operation mode

1. From the status monitor, enter **GLOBAL** at the prompt.

   A Global Command Parameter Specification panel appears.

2. Enter **AUTOMATED** or **MANUAL** in the Global Automation Mode field and press F6 (Action).

   A confirmation panel appears.

3. Enter **CONFIRM** in the Response field.

   The region changes the operation mode of all resources.

Example: Set Global Operation Mode

If the region is running in the MANUAL operation mode and you want to test the effects of automation on the resources in the system, set the global operation mode to AUTOMATED.

Enter the following command at the prompt of the monitor:

```
GLOBAL MODE=AUTOMATED
```

The Execute GLOBAL Command panel is displayed. Enter **S** next to the required system image. The region sets all of the resource operation modes to their normal value. This normal value is the mode defined in the resource or set by an override.

Chapter 9: Implementing Status Monitor Filters

This section contains the following topics:

Implement the Status Monitor Filters

You use filters to customize a Status Monitor panel. For example, you can define a filter that causes the Status Monitor to display only those resources that are applicable to a subset of your network.

A Status Monitor filter uses a Boolean expression, which you define on the Status Monitor Filter panel, to determine what to display on the monitor. You restrict the display by using the resource attributes such as names and status.

When you save a filter definition in the knowledge base, the definition propagates automatically to all the connected regions—that is, the definition is global.

Access Status Monitor Filter Definitions

Status Monitor filters let you configure your view of monitored resources to suit your requirements. You can selectively view different groups of resources by swapping filters.

To access Status Monitor filter definitions, enter **/ASADMIN.F** at the prompt.

The Status Monitor Filter List appears.

The panel displays the list of filter definitions in the knowledge base. You can add a new definition, or browse, update, copy or delete an existing definition.

Add a Status Monitor Filter

To add a Status Monitor filter definition

1. Access the Status Monitor Filter List.
2. Press F4 (Add).

   The Status Monitor Filter panel appears.

   Note: If you change your mind and do not want to add the filter, press F12 (Cancel) to cancel the operation any time before Step 5.
3. Complete the Name and Description fields in the Filter Definition window to identify the new filter.

   Note: Press F1 (Help) for a description of the fields.
4. Specify a Boolean expression (see page 100) in the Filter Expression window to define the filter.
5. Press F3 (File).

   The new definition is saved.

Status Monitor Filter Panel

The Status Monitor Filter panel specifies the details of a Status Monitor filter. The operation you are performing is displayed at the top right of the panel, for example, Function=UPDATE.

The panel displays two windows. The Filter Definition window identifies the filter by name and description, and the Filter Expression window specifies the Boolean expression that defines the filter.

Example: Status Monitor Filter Panel

```
PROD-------- Automation Services : Status Monitor Filter -------Function=UPDATE
Command ===>                                                    Scroll ===> 10

+ Filter Definition ------------------------------------------------------------+
| Name ......... ATTENTION                                                     |
| Views ........                                                               |
| Description .. RESOURCES THAT ARE IN ATTENTION STATE                         |
| Last Updated at 22.09.04 on WED 24-MAY-2006 by USER01                        |
+------------------------------------------------------------------------------+
+ Filter Expression ------------------------------------------------------------+
|                                                                              |
|                                              D=Delete I=Insert R=Repeat      |
|      "(" Field   Opr Value                                  Gen ")" Bool     |
|          LOGSTAT =   "ATTENTION"                                             |
|     **END**                                                                  |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|                                                                              |
|  F1=Help     F2=Split    F3=File     F4=Save                                 |
|  F7=Bkwd     F8=Forward  F9=Swap                          F12=Cancel      |
+------------------------------------------------------------------------------+
```

Status Monitor Views

A view customizes your Status Monitor for the specific purpose of monitoring certain classes of objects. Each view has associated with it a selected set of filters and display formats.

To see the supported views, enter **?** in a View field.

How You Define the Status Monitor Filter Expression

Use the Filter Expression window on the Status Monitor Filter panel to specify the Boolean expression that defines the filter. The expression uses resource attributes as criteria to determine what to display on the Status Monitor.

To display the list of valid values for a field, enter a question mark (?) in the field.

Use the following action codes to help you enter the expression:

D

Deletes the selected line.

I

Inserts a blank line after the selected line.

R

Repeats a selected line.

Example: Define a Status Monitor Filter

This example defines a filter named RSCALERT that enables an operator to monitor resources that have a DEGRADED, FAILED, or UNKNOWN logical state. The following panel shows the completed filter.

```
PROD--------- Automation Services : Status Monitor Filter ------Function=BROWSE
Command ===>                                                  Scroll ===> CSR

. Filter Definition ------------------------------------------------------------.
| Name ......... RCSALERT                                                      |
| Views ........                                                               |
| Description .. Resources in DEGRADED, FAILED, or UNKNOWN state               |
| Last Updated at 15.09.30 on WED 24-MAY-2006 by USER01                        |
'------------------------------------------------------------------------------'
. Filter Expression ------------------------------------------------------------.
|                                                                              |
|                                                                              |
|      "(" Field   Opr Value                                    Gen ")" Bool   |
|      (   LOGSTAT =   "DEGRADED"                                       OR     |
|          LOGSTAT =   "FAILED"                                         OR     |
|          LOGSTAT =   "UNKNOWN"                                    )          |
|     **END**                                                                  |
|                                                                              |
|                                                                              |
|                                                                              |
|  F1=Help      F2=Split     F3=Exit      F4=Edit      F5=Find      F6=Refres  |
|  F7=Backward  F8=Forward   F9=Swap                              F12=Max     |
'------------------------------------------------------------------------------'
```

The filter expression causes a Status Monitor to display only services that have the DEGRADED, FAILED, or UNKNOWN logical state.

Maintenance of Status Monitor Filter Definitions

You can browse, update, copy, and delete filter definitions from the Status Monitor Filter List panel.

If the Filter Expression window does not fully display the Boolean expression while you are browsing a definition, press F12 (Max) to expand the window.

Note: After you update a filter definition, an operator who is already using that filter does not see the update. To use the updated filter, the operator must enter the REFILTER command.

Chapter 10: Implementing Resource Templates

This section contains the following topics:

Resource Templates

Important! The supplied INTNL class resource templates are required for the region to function properly. Do not modify these templates.

After you have defined a system image, you can define resources in it. Your product includes sample resource templates, which you can use to define commonly used resources. The templates supply values for certain resource definition fields, and simplify the task of creating your own specific resource definitions. You can modify the sample templates or create your own templates. You can create templates for the different resource types in each class of resource.

You can maintain several versions of templates as different $TEMPLAT system images. Each version can contain, in addition to the resource templates, the availability maps and processes used by resource templates.

USRCLS Class Template

No sample USRCLS class templates are supplied. However, you can create your own templates to facilitate the definition of similar resources. The templates provide the methods for operating USRCLS class resources (if supported by your product).

Set Up Your Template System

Templates are defined in a $TEMPLAT system image. Your template system may contain different versions of templates. Group each version in a different $TEMPLAT system image.

Before you work on templates, copy the supplied templates to a different $TEMPLAT version. Start with version 0010; versions 0001 through 0009 are reserved for software updates.

To copy a $TEMPLAT system image

1. Enter **/RADMIN.T.I** at the prompt.

   The Template System Image List panel appears.

2. Enter **C** beside the system image you want to copy.

   The System Image Definition panel opens.

3. Change the value in the Database Version field to uniquely identify the new copy (for example, 0010), and update the description fields as required.

4. Press F3 (File).

   The System Image Copy panel appears advising you of the status of the copying process. When the copying process is complete, the System Image List panel appears.

5. Set up one $TEMPLAT system image version for general use. Review the templates to ensure that they are suitable for the resources on your system. The version to use is set in the OPSYSIDS parameter group under the NAMES category during region initialization. Enter the **/PARMS** shortcut to access the Customizer : Parameter Groups panel that enables you to access the parameter for update.

$TEMPLAT System Image for Multiple Products

Each product supplies its own templates for the supported resource classes. If you want to run different products in the same region, merge the $TEMPLAT system images that contain those templates.

Note: For information about how to merge system images, see the *Reference Guide*.

Associate a Template to a Resource Class

To associate a template to a resource class

1. Enter **/RADMIN.T.R** at the prompt.

   The Resource Template Definition List appears.

2. Enter **S** next to the resource class to which you want to associate the template.

   A list of templates associated with the resource class appears.

3. Enter **AP** in front of the template.

   The Automation Services : Apply Template panel appears.

4. Define how you want to apply the template and press F6 (Action).

   The ResourceView : System Image List appears.

5. Select the system image to which you want to apply the template.

   The Automation Services : Messages List panel appears with details of the process.

6. Press F3 (File).

   All resources on the selected images that are associated with the template are updated.

Resource Template Definitions

Note: The name of a template must contain alphanumeric, @, #, $, ., :, -, (, and) characters only. It must not be a number.

The panels used to add a resource template definition for a particular resource class are the same as the panels that you use when you add a resource definition for that class. You can define any information that will be used generically by a specific resource.

Variables

You can use a variable to supply the value for a field in the resource template definition.

Disable Substitution of Variables

Variables in a template are substituted by their values when you apply the template to a resource definition. You can disable variable substitution—that is, you want the variable to appear in the resource definition, *not* the value of the variable.

To disable the substitution of a variable during application, replace the ampersand (&) in front of the variable name by the underline character (_).

For example, if you specify _ZMSGTEXT in a template and apply the template to a resource definition, _ZMSGTEXT becomes &ZMSGTEXT in the resource definition.

Specify a Variable to Represent a Left-justified Fixed-length Field

Some messages contain left-justified fixed-length fields for resource names. If the name is not of the maximum length, the name is left justified. You cannot use normal variables because they do not provide padding.

To handle left-justified fixed-length fields, use less-than signs (<).

Each < represents one character. For example, <<<<< represents a five-character field with left justification.

Specify a Variable to Represent a Right-justified Fixed-length Field

Some messages contain right-justified fixed-length fields for resource names. If the name is not of the maximum length, the name is right justified. You cannot use normal variables because they do not provide padding.

To handle right-justified fixed-length fields, use greater-than signs (>).

Each > represents one character. For example, >>>>> represents a five-character field with right justification.

Maintenance of Resource Template Definitions

You can browse, update, copy, and delete resource template definitions. You can copy a resource template definition between or in $TEMPLAT system images.

Apply Updated Templates

You may have defined a number of resources by using a template and that template has since been updated. You can use the AP action code to reapply the template to update those resource definitions.

To apply updated templates

1. From the templates list, enter **AP** beside a template.

   The Apply Template panel appears.

2. Specify how the updates are performed.
3. Press F6 (Action).

   A list of system images appears.

4. Enter **S** beside the system images that contain the resource definitions that you want to update and then press Enter to apply the template to the included definitions.

Availability Maps in a Template System Image

You can define availability maps in a $TEMPLAT system image. You can then use these maps with resources built from the templates.

The procedures for creating and maintaining maps for resource templates are similar to the procedures for creating and maintaining maps for resource definitions.

Access Map Definitions in a Template System Image

To access the map definitions in a $TEMPLAT system image

1. Enter **/RADMIN.T** at the prompt.

   The Template Definition Menu appears.

2. Enter **A** at the prompt.
3. (Optional) If you want to use a different version of the $TEMPLAT system image, change the value in the Template Version field and then press Enter.

   The relevant map list panel appears. The panel lists all the maps in the selected $TEMPLAT system image.

Define and Maintain Processes in a Template System Image

You can use the processes in a $TEMPLAT system image in a resource template belonging to the same image. You can create new processes or change existing processes.

The procedures for creating and maintaining processes for resource templates are similar to the procedures for creating and maintaining processes for resource definitions (see page 111).

Access the Process Definitions in a Template System Image

To access the processes in a $TEMPLAT system image

1. Enter **/RADMIN.T** at the prompt.

   The Template Definition Menu appears.

2. Enter **P** at the prompt and, if you want to use a different version of the $TEMPLAT system image, change the value in the Version field.

   The Process List panel appears. The panel lists the processes in the selected $TEMPLAT system image.

Convert a Resource Definition into a Resource Template

You can convert a resource definition into a resource template to facilitate future definition of similar resources. After you are satisfied that a resource definition is working correctly, you can convert the definition into a template.

To convert a resource definition into a resource template

1. Use the Copy action to create another copy of the definition.
2. Change the system name on the General Description panel to $TEMPLAT, and specify the version of the $TEMPLAT image into which you want to copy the definition in the Database Version field.
3. Name the template on the General Description panel.
4. Replace the resource names on the other definition panels by *one* of the following:
   - &ZRMDBNAME if the name field is not of fixed length
   - Less-than signs (<) if the name field is of fixed length with left justification—this typically occurs in the message text
   - Greater-than signs (>) if the name field is of fixed length with right justification—this typically occurs in the message text

   Note: Keeping the name length to less than the maximum number of characters enables you to easily recognize the fixed length name fields in a message. For example, a seven-character name is displayed with an extra space in an eight-character fixed length field.
5. Replace the ampersand (&) in front of a variable by the underline character (_).
6. File the definition. Any associated availability map and processes are also copied if they do not exist already in the specified $TEMPLAT system image.

Chapter 11: Implementing Processes

This section contains the following topics:

How to Implement Processes

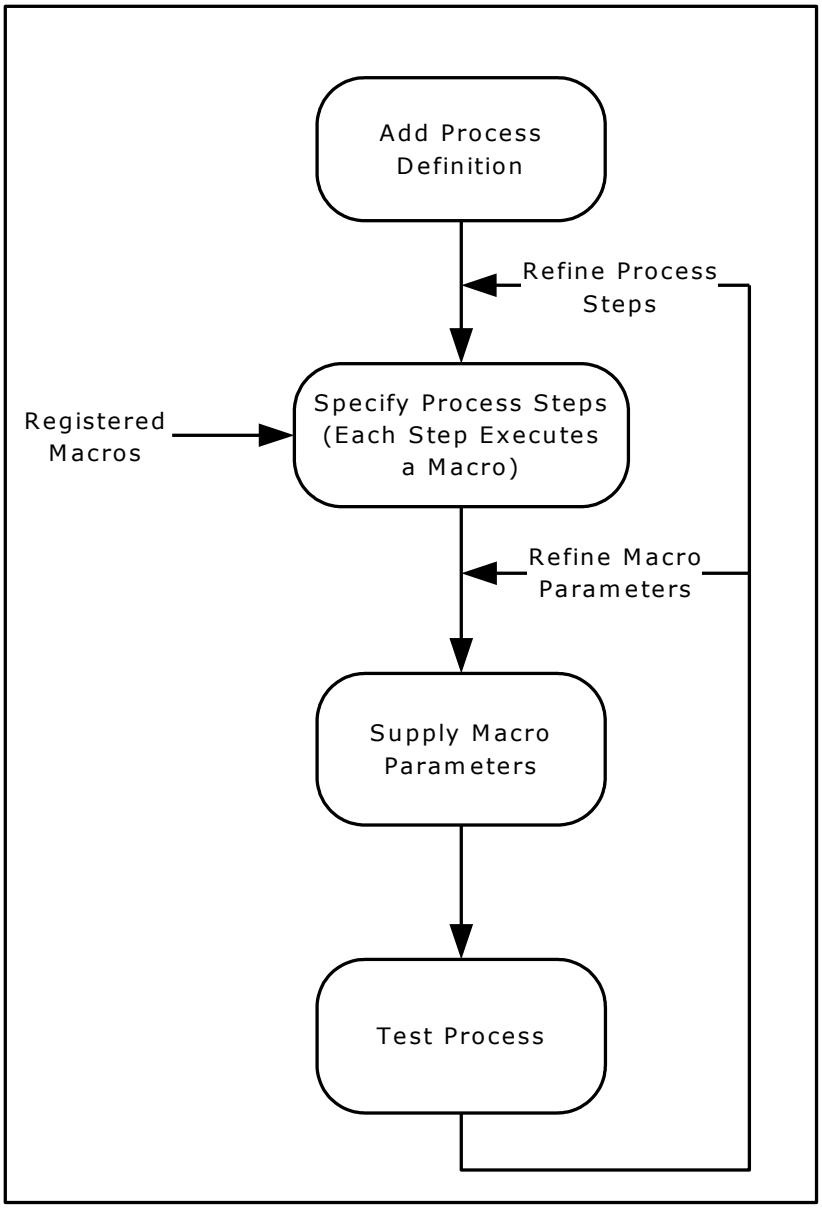
A process is a series of steps that can be executed in sequence to perform complex processing.

You define processes to automate complex operations tasks.

Processes can be executed as follows:

- From a resource definition—you can specify a process in a resource definition. The process is invoked when required for that resource.
- From an availability map—you can specify a process in an availability map (for example, to perform tasks at particular times).
- From an event rule—you can specify a process in an event rule. The process is invoked when an event triggers the rule.
- As a single task—you can run a process as a single, independent task. Use this feature to debug processes or as a quick way of executing a process manually.
- Interactively—you can run a process in the INTERACTIVE mode. Use this feature to check the results of processing single steps, or of processing a sequence of steps one at a time. You can display individual step logs and, if required, change the step parameters.

The following illustration shows the typical stages in defining a process.



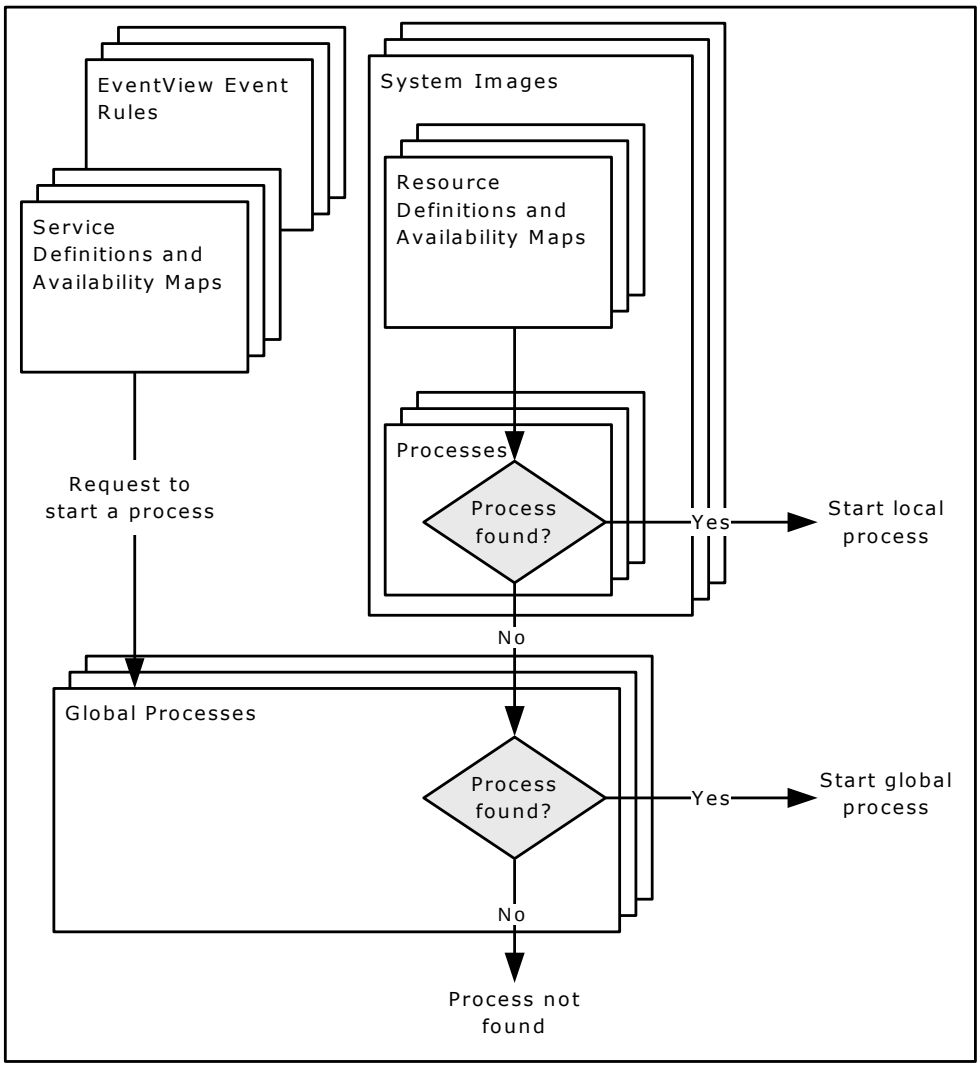
Process Types

A process can be global (available to all components) or local (available to a specific system image only). A global process is available to all components; however, a local process is available only if it belongs to the local active image.

ServiceView and EventView components can use global processes only. ResourceView components can use both types of processes, according to the following rules:

- If a process is required and one exists in the local active system image, that process is used.
- If the required process does not exist in the active system image, the global process of the same name is used.

The following illustration shows how processes are searched for execution.



Access Process Definitions

Each system image has its own set of processes and access to global processes belonging to the $PROCESS 0001 system image.

To access the local process definitions in a system image

1. Enter **/RADMIN**.

   The Resource Administration menu appears.

2. Type the option code **P**, and the name and version of the system image, and press Enter.

   The Process List panel appears. This panel lists the processes in the system image and the global processes (displayed in blue on a color terminal).

To access the global process definitions

1. Enter **/RADMIN.GP**.

   The Process List panel appears. This panel lists the global processes.

How to Define a Process

From the Process List panel, press F4 (Add) to add a process definition. A Process Definition panel is displayed.

To define a process, first decide what you want the process to do, then break it down into steps, each step representing an action. Specify a macro for each step. A macro is an NCL procedure that performs the processing for that step. Authorized users can use the Register Macros option to register new macros.

Step processing can be conditional on the processing result of an earlier step. In the following example, STEP2 runs if STEP1 processing returns a code of 0. STEP3 runs if STEP1 processing returns a code greater than 0.

```
                Condition
StepName    Step/RC  Opr RC
STEP1       STEP1
STEP2       STEP1     =   0
STEP3       STEP1     >   0
```

When you define a process on the Process Definition panel, complete the following fields:

- Name and Description fields to identify the process
- StepName and Macro fields to define each step

If you want to find out what macros are available, enter **?** in a Macro field to display the list of available macros.

Important! $NCL is the name of a special process definition. Do not use this name when you add process definitions.

Conditions are optional. Use relational operators in the Opr fields to set the conditions. Enter **?** in an Opr field to identify the valid relational operators.

You can repeat and delete steps, and insert blank lines.

Press F11 (Right) to display the parameters for each step.

The return code from a process is the return code from the last executed process step.

Example: Issue Multiple System Commands

The following shows an example of a process that issues multiple system commands.

```
PROD---------- Automation Services : Process Definition ----------Function=Add
Command ===>                                                  Scroll ===> PAGE

+ Process Definition ------------------------------------------------------------+
| System Name .. PROD      Version .. 0001   Last Updated By                     |
| Name ......... TEST PROC                   at           on                    |
| Description .. ISSUE SYSTEM COMMANDS                                           |
+--------------------------------------------------------------------------------+
+ Process Steps ------------------------------------------------------------------+
|                                                                                |
|                                        D=Delete I=Insert P=Parms R=Repeat      |
|                    Condition                                                   |
|      StepName   Step/RC  Opr RC    Macro    Description                       |
|      STEP1                         SYSCMD   EXECUTE A COMMAND                 |
|      STEP2      STEP1    =   0     SYSCMD   EXECUTE A COMMAND                 |
|      STEP3      STEP2    =   0     SYSCMD   EXECUTE A COMMAND                 |
|      STEP4      STEP1    =   99    SYSCMD   EXECUTE A COMMAND                 |
|                                                                                |
|                                                                                |
|                                                                                |
|                                                                                |
|  F1=Help     F2=Split    F3=File     F4=Save                                   |
|  F7=Bkwd     F8=Forward  F9=Swap               F11=Right   F12=Cancel        |
+--------------------------------------------------------------------------------+
```

If STEP1 completes successfully, STEP2 executes the next shutdown command. If STEP2 completes successfully, STEP3 issues the final shutdown command.

If STEP1 fails, STEP4 executes and issues a CANCEL command.

Set Macro Parameters

When you select a macro, it contains either no parameters or default parameters.

To set the parameters for a macro

1. Enter **P** next to the process step.

   A Macro Parameter Definition panel appears.

2. Change the parameters as required and press F3 (OK). The parameters required by each macro depend on the purpose of the macro.

Example: Set Macro Parameters

The following shows the parameters set for Step 1 in the previous example.

```
PROD----- Automation Services : SYSCMD Macro Parameter Definition -------------
Command ===>                                                  Function=UPDATE

+- System Command ------------------------------------------------------------+
|                                                                             |
| Command ..... F CA7T,/LOGON MASTER_____________________________________     |
|                                                                             |
| Jobname ..... ________                                                      |
|                                                                             |
| Wait Time ... 30__  Wait Time Expiry Return Code ... 99_                    |
+-----------------------------------------------------------------------------+
+- Response Message Analysis --------------------------------------------------+
|                                D=Delete Extended Filter  S=Extended Filter  |
|           Message Text                                    Return Extended   |
|                                                           Code   Filter?    |
|   ___     CA-7.023 - V3.0 (9106) OPERATOR IS LOGGED ON_   0__    NO         |
|   ___     ____________________________________________    ___               |
|   ___     ____________________________________________    ___               |
|   ___     ____________________________________________    ___               |
|   ___     ____________________________________________    ___               |
|                                                                             |
+-----------------------------------------------------------------------------+
 F1=Help      F2=Split     F3=OK
                           F9=Swap                              F12=Cancel
```

The parameters include:

- The system command issued
- The text of the expected response
- A processing return code of 0
- A wait time of 30 seconds
- A time-out return code of 99

You can also specify an extended filter for the analysis of the response message text. For example, a response can contain variable information and you want to accept the message only if it contains specific values.

Variable as a Macro Parameter

You can use a variable to hold the value of a macro parameter. You pass the value of any variables required by a process as parameters when you specify the process, for example, in a resource definition.

Important! Do *not* specify variable names that start with #, $, or Z.

Example: Use a Variable as a Macro Parameter

You have defined a process that contains the SYSCMD macro which issues the $DU,&PRT command. When you use the process, you supply the value of the &PRT variable by specifying the following parameter: PRT=*printer-name*. Specify the name of the variable only (without the &).

Generic Processes Using Resource Variables

You can define generic processes that perform functions that are dependent on how they are initiated by using resource variables. These variables contain information about a resource that is defined to the knowledge base. They are useful for building automated paging, standardized startup for CICS regions, and many other tasks where a uniform solution is required. Using a generic process reduces any overhead associated with building individual processes for individual resources.

Note: For information about knowledge base variables, see the *Reference Guide*.

Example: Use Process to Page Support

Service level agreements require that appropriate support personnel are pageable if any production CICS region is under stress. Unicenter Automation Point is available at your site to monitor the condition and provide the paging function. Different CICS regions have different support personnel assigned.

You implement the following method in the CICS resource definitions:

1. Specify details of the support personnel.
2. Identify and specify the message to trigger automated paging.
3. Specify an event-related action for this message using the following generic process:

	Condition					
	StepName	Step/RC	Opr	RC	Macro	Description
	S1				WTOR	WTOR TO L1 SUPPORT
	S2	S1	EQ	32	WTOR	TIMED OUT - CALL L2
	S1OK	S1	EQ	0	SETSTATE	L1 RESPONDED - SET EXT. DISPLAY
	S2OK	S2	EQ	0	SETSTATE	L2 RESPONDED - SET EXT. DISPLAY
	S3	S2	NE	0	GENALERT	NO SUPPORT - RAISE ALERT
	S4	S2	NE	0	SETSTATE	NO SUPPORT - SET EXT. DISPLAY

   a. At Step S1, the resource sends a WTOR message, using knowledge base variables (for example, &ZRMDBREOPAG1 that contains the pager number) to provide details of the support personnel responsible for the failing resource.

      Unicenter Automation Point or by an operator intercepts the WTOR message, and the indicated first-level support person is paged. Response to the message indicates the success or failure of paging.

   b. If paging of the first-level person is successful, Step S1OK sets the extended display of the resource to indicate that the support person has acknowledged the paging.

      If no reply is received within a specified period, Step S2 sends another WTOR message to invoke paging of the second-level support person.

   c. If paging of the second-level person is successful, Step S2OK sets the extended display of the resource to indicate that the support person has acknowledged the paging.

      If paging fails, Step S3 raises an alert and Step S4 sets the extended display of the resource to indicate that no support personnel have responded.

Processes to Generate Alerts

You can use a process in a ResourceView resource definition or an EventView message rule to generate alerts in response to problems occurring in a resource.

The GENALERT macro enables you to generate an alert from a process.

Example: Generate Alert on Security Violation

The DFHAC2003 message indicates that a CICS security violation has occurred. You may want to be warned of these violations. The following panels show the message rule definition that generates an alert under this condition by using the SECALERT process definition:

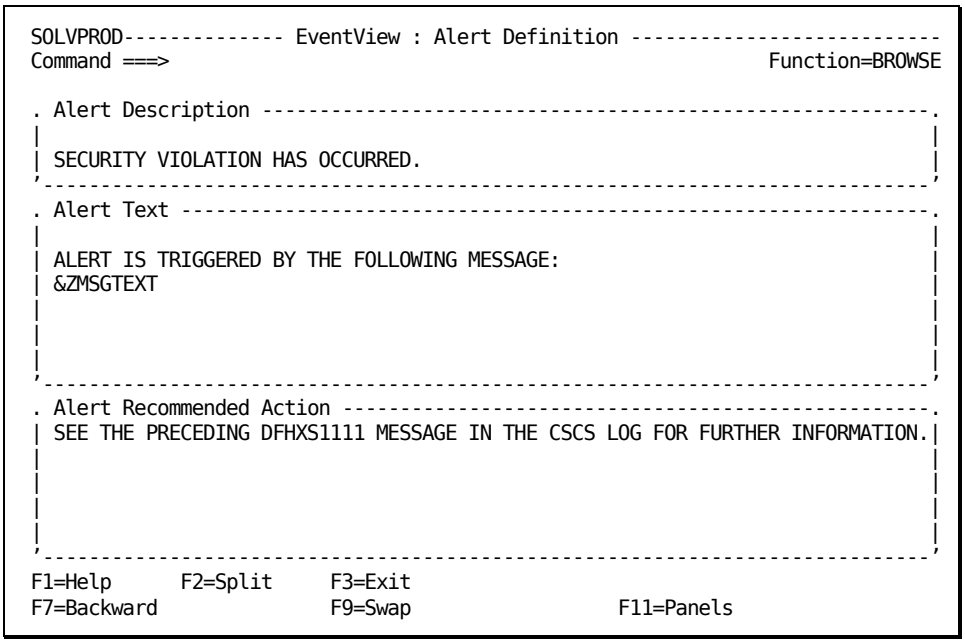
```
SOLVPROD--------------- EventView : Message Filter -------------------CICSSEC--
Command ===>                                                  Function=BROWSE

Ruleset Name ........ CICSSEC                       Rule Status .... ACTIVE
Short Description ... CICS security alerts

. Expected Message ------------------------------------------------------------.
|                                    S=ListPanels E=ExtFilter T=TestVars |
|      Message Text  ( WildChar = * )                                ExtFlt |
| ___  DHFAC2003                                                     NO     |
```

```
SOLVPROD---------- EventView : DHFAC2003 Message Actions ------------CICSSEC--
Command ===>                                                  Function=BROWSE

Reply Text .......

System Command ...

MS Command .......


. Automation Actions ----------------------------------------------------------.
|                                                  S/B=Browse U=Update L=List |
|      Process     Parameters                                                 |
| ___  SECALERT                                                               |
```

The following panels show the SECALERT process definition and the parameters used by the GENALERT macro:

```
SOLVPROD------- Automation Services : Process Definition -------Function=Browse
Command ===>                                                  Scroll ===> CSR

. Process Definition ------------------------------------------------------.
| System Name .. $PROCESS  Version .. 0001   Last Updated By USER01          |
| Name ......... SECALERT                    At 16.21.13  On WED 24-JUL-1996 |
| Description .. CICS security violation alert generator                     |
'----------------------------------------------------------------------------'
. Process Steps -----------------------------------------------------------.
|                                                                           |
|                                                                  P=Parms  |
|                    Condition                                              |
|      StepName   Step/RC  Opr RC    Macro    Description                   |
| P    A                             GENALERT GENERATE AN EVENTVIEW ALERT   |
|     **END**                                                               |
```

```
SOLVPROD-------------- EventView : Alert Attributes ---------------------------
Command ===>                                                  Function=BROWSE

. Alert Reference Key -----------------------------------------------------.
|                                                                          |
| Reference ... CICS_SECURITY_ALERT_&ZMSGWORD20                            |
|                                                                          |
'--------------------------------------------------------------------------'
. Alert Attributes --------------------------------------------------------.
|                                                                          |
| Severity .... 2                                                          |
| Type ........ DEFAULT                                                    |
| Origin ...... ALERTMACRO                                                 |
```



How You Test a Process

After you have defined a process, you can test it by executing it as a single task or by executing it in the interactive mode.

Test a Process Interactively

From the Process List panel, enter **I** beside a process to execute it in the interactive mode. The Process Definition panel for that process appears. You can:

- Enter **E** beside a step to execute only that step irrespective of the condition.
- Use F12 (Step) to execute a number of steps in sequence. Pressing F12 (Step) executes the next step in the sequence. The execution of each step depends on the condition specified for the step.
- Enter **L** beside an executed step to see the processing log. The log display is positioned at the latest entries relating to the selected step.
- Enter **P** beside a step to view the macro parameters.

To interactively edit and test the process steps

1. Press F4 (Edit) to access the Interactive Edit function to edit the process steps.
2. Modify the steps, as required.
3. When you complete the modifications, press F4 (OK) to return to the INTERACTIVE mode. You can also press F3 (File) to return to that mode. Pressing F3 (File) saves the modifications.
4. Test the modified process.
5. Press F3 (Exit) and F3 (File) again to save the modified steps.

   If the test is not satisfactory, restart from Step 1.

Test a Process by Execution as a Single Task

To test a process by execution as a single task

1. From the Process List panel, enter **E** beside a process.

   The task is executed as a single, independent task. The Optional Process Parameter Specification panel appears.

   Note: When you use the E action code to execute a process, the process is executed under the BSYS background user ID.
2. Supply any parameters required by the process in the Parameters field, then press F6 (Action).

   When the process has executed, a processing log appears. This log contains the processing results.

How You Log Process Activities

Process activities are written to the activity log while you are testing a process. However, you can control the logging when a process is executed, for example, from a resource definition. Use the $LOG process parameter to control the logging as follows:

$LOG=BOTH

Logs activities in full and summary form.

$LOG=FULL

Logs activities in full.

$LOG=NO

(Default) Does not log activities.

$LOG=SUMM

Logs activities in summary form only.

Maintenance of Process Definitions

You can browse, update, copy, and delete process definitions from the Process List panel.

Back Up Global Processes

To assist you with the maintenance of your global processes, you can create backup versions of your global process image. By creating a backup version of your global process image, you can perform the following:

- Update global process definitions in any version of a global process image.
- Restore a global process definition from a backup global process image.
- Merge two versions of a global process image.

To create a backup version of a global process image

1. Enter **/ASADMIN.GPI** at the prompt.

   The Global Process Image List appears.

   Note: If you have not created a backup before, there is only one global process image listed: $PROCESS 0001. The active global process image can only be $PROCESS 0001. $PROCESS 0001 cannot be deleted.

2. Enter **C** beside the global process image you want to copy.

   The Global Process Image Definition panel appears.

3. Enter a new Database Version, Short Description, and Long Description.
4. Press F3 (File).

   The backup version of the global process image is saved. A copy in progress panel appears while the copy occurs. The Global Process Image List appears with the backup version displayed in the list.

   If the global process image you have specified already exists, the Confirm System Image Merge panel appears.

Update Global Process Definitions in a Backup Global Process Image

You can access a list of all the global process definitions in any version of a global process image. From this list you can update any global process definition contained in the global process image.

To update a global process definition in the $PROCESS 0002 backup image created above

1. Enter **L** (List Processes) beside the $PROCESS 0002 global process image in the Global Process Image List.

   The Global Process List panel appears showing all of the global process definitions in that global process image.

   Note: You can access the list of global processes for another version of the global process image by changing the version number on the Global Process List panel and pressing Enter.

2. Enter **U** beside the global process definition that you want to update.

   The Process Definition panel appears for that global process definition.

3. Update the global process definition, as required.

4. Press F3 (File).

   The changes are saved and the Global Process List appears.

Restore a Global Process Definition from a Backup Global Process Image

If you have made changes to a global process definition and you are having trouble with its implementation, you can restore it from a previous version of the definition.

To restore global process definition $PROC01 from $PROCESS 0002 to $PROCESS 0001

1. Enter **C** beside $PROC01 in the global process list.

   The Process Definition panel appears.

2. Change the Database Version from 0002 to 0001 and press F3 (File).

   The changes are saved. Because there is already a copy of the global process in the target global process image, the Confirm Copy Replace panel appears.

3. Press Enter to confirm the replace or F12 (Cancel) to cancel the request.

   The Global Process List appears.

Change a Global Process to a Local Process

You can change a global process to a local process while performing a copy on any global process in the global process selection list.

To change global process PROC01 to a local process in the SYS01 system image

1. Enter **C** beside PROC01 in the Global Process List.

   The Process Definition panel appears.

2. Change the System Name to SYS01 and press F3 (File).

   The Global Process List appears.

   To view the new local process, access the list of processes for the system image that you copied it to.

Merge Two Global Process Images

You can merge two global process images and replace the active global process image with a backup version.

To merge global process images $PROCESS 0002 and $PROCESS 0001

1. Enter **C** beside the $PROCESS 0002 on the Global Process Image List.

   The Global Process Image Definition panel appears.

2. Change the Database Version number to 0001 and press F3 (File).

   The Confirm System Image Merge panel appears.

3. Enter **YES** in the input field if you want to overlay like-named components.

4. Press F6 (Confirm).

   The global process images are merged.

Chapter 12: Implementing Availability Maps

This section contains the following topics:

Availability Maps

An availability map enables you to define the availability requirements for a service or resource. An availability map also enables you to schedule the execution of processes. You can add an availability map at any time. The map becomes effective as soon as you attach services or resources to it.

How You Implement Availability Maps

The desired state information specified in a service or a resource definition determines its status. The definition can include an availability map that schedules changes to the default availability. Timers activate these changes.

Note: The default desired state determines the default availability of a service or resource. The state is set in the AUTOIDS parameter group during region initialization. The Customizer : Parameter Groups panel lists the region parameter groups. Enter the **/PARMS** shortcut to access the panel.

Availability maps enable you to schedule changes to the default availability requirements of one or more services or resources. The service image and each system image have its own set of availability maps. You define an availability map (for example, MAP1) and attach as many services or resources to the map as required. Because availability maps are not limited to a seven-day cycle, you can define changes to the availability requirements that apply daily, on the same day every week, on the same date every month, for a specific date and time, and so on. You can also suppress changes temporarily and update timer information at any time.

An availability map has two parts: a map definition and a timer definition. The map definition contains information about the map itself. The timer definition contains information about when to change the desired state of the services or resources that use this map. The timer definition can also contain information about when to change the operation mode and when to start processes to perform special tasks.

Creating an availability map has the following two stages:

1. Creating an availability map.
2. Attaching services or resources to a map.

Note: For information about how availability and resource relationships affect operations, see the *Reference Guide*.

More information:

How You Specify the Availability of the Service and Its Members (see page 140)

Rules for Availability Map Definitions

The following rules apply to availability maps:

- If the timer definition is blank, it means that default availability requirements apply to all the services or resources attached to that map.
- A map only applies to the service image or the system image for which it is defined.
- Map names must be unique in the image to which the map applies.

Access Availability Map Definitions

You can define as many maps for a system image as you want. After the map is defined, you can define timer information and attach services or resources to the map. Use the Availability Maps option to create and maintain availability map definitions.

The service image and each system image have its own set of availability maps.

To access service availability map definitions

1. Enter **/SADMIN.A** at the prompt.

   The Availability Map List appears.

To access resource availability map definitions

1. Enter **/RADMIN** at the prompt.

   The Resource Administration menu appears.

2. Enter **A** at the prompt and the name and version of the system image that owns the maps you want to create or access, and then press Enter.

   The Availability Map List panel appears. This panel lists the availability maps for the specified service or system image.

Note: To display the maps owned by another system image or resource, you can enter another name (resources only) or version number at the top of this panel.

Temporary Availability Maps

A temporary availability map is an availability map created from the status monitor to override the current map attached to a service or resource. A temporary map has an expiry time when the map is deleted automatically. You can use a temporary map as any other map, remembering that it has a defined life time.

Create an Availability Map

To create an availability map

1. Press F4 (Add) from the Availability Map List panel.

   The Availability Map panel appears.

2. Specify the timer information that sets the availability requirements. For information about the fields, press F1 (Help).

How You Define Timers

You can define two types of timer information:

- For all services or resources, define the timer, leaving the SVC/Resource Name field blank. This timer information applies to any services or resources attached to the map.
- For a specific service or resource, define the timer with the name of the service or resource in the SVC/Resource Name field. This timer information applies to the named service or resource if the service or resource is attached to the map.

You can use the action codes to repeat or delete rows of information, or to insert blank lines.

Use the following values in the Day field to simplify data entry:

*****

Repeats the timer for all days (that is, Monday through Sunday).

W/D

Repeats the timer for weekdays (that is, Monday through Friday).

W/E

Repeats the timer for weekends (that is, Saturday and Sunday).

Leave the Day field blank if you fill in the Date field. If the Mode field is left blank, you do not override the operation mode.

Scheduling of Processes

If you want the map to start processes at defined times, press F11 (Right) to display the fields for specifying processes.

Manual Overrides

You can use a timer to reset manual desired state and operation mode overrides. Specify RESET in the Des.State and in the Mode fields.

When a manual override exists, the scheduled change to the overridden parameter cannot be made. If you want to help ensure that the scheduled changes are made, reset the overrides first.

Note: For more information about how to perform manual overrides, see the *User Guide*.

Availability Map Example

This example describes how to define an availability map for services.

In this example, you define a map for the defined services to schedule such things as availability during holidays and when system maintenance is required. The map is named MAP1.

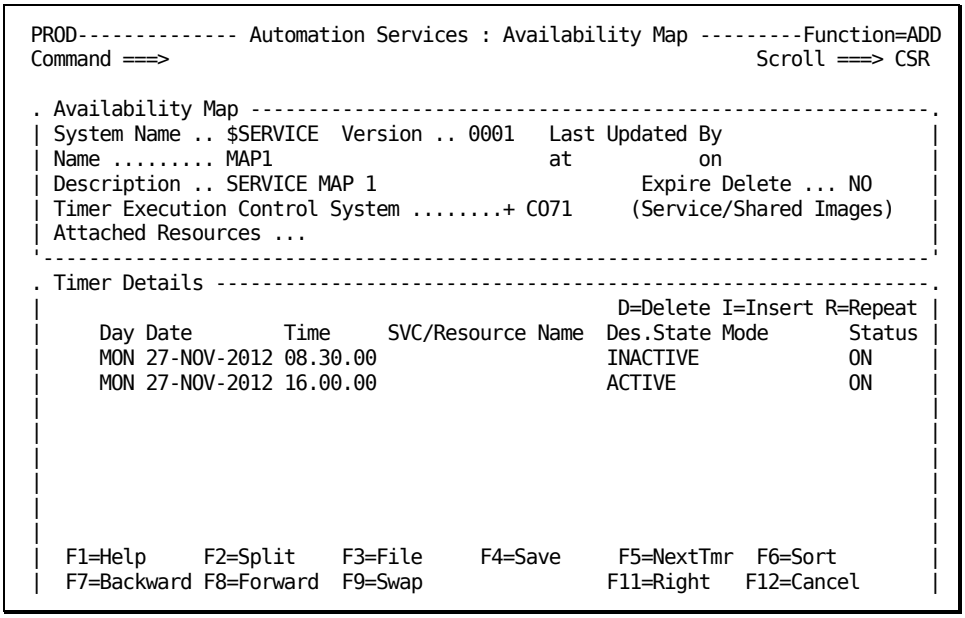
Use the **/SADMIN.A** path and the F4 (Add) function key to access the Availability Map panel. On the panel, you type the following values:

- **MAP1** in the Name field
- A description in the Description field
- **N** in the Expire Delete field to retain expired timer events (These events occur on specific dates.)

You can now specify timer details.

You want to stop all services on 27 November 2012 at 0830 hours for system maintenance and reactivate all services at 1600 hours on the same day. (Resources that belong to the services have a scheduled INACTIVE desired state for all times. That is, the services control the availability of those resources by using the ACTIVE desired state overrides.)

In the Timer Details box, type the information about the date, the time, the change to the status, and whether to process this change. To have a change processed, specify **ON** in the Status column. The following shows the completed Availability Map panel.



Timer Information

The Next Timers Execution Time panel lists information about upcoming changes to availability. You can obtain different views of this timer information by:

- Viewing the timer information in all availability maps for the services or in a system image
- Viewing the timer information in one availability map

The views list the next invocation of the defined timers. For example, a timer that executes every Monday is listed once only.

View All Timer Information

You can view a list of the upcoming changes scheduled in all maps defined in the service image or in a system image. The changes are listed in chronological order.

To view this information from the Availability Maps List panel, press F12 (NextTmr).

A Next Execution Time panel appears, listing the upcoming changes for all the maps.

View the Timer Information in One Availability Map

You can view a list of the upcoming changes scheduled in an individual map. The changes are listed in chronological order.

To view the timer information in an availability map

- From the Availability Map List panel, enter **N** next to an availability map to select the NextTimers action.
- From an Availability Map panel (while you are working on an availability map definition, a resource definition, or a service definition), press F5 (NextTmr).

A Next Execution Time panel appears, listing the upcoming changes for the selected map.

Attach a Service or Resource Definition to an Availability Map

After a map is defined, you can attach service or resource definitions by using:

- The Availability Map List
- The service or resource definition panels

To attach a service or resource definition to an availability map from the Availability Map List

1. Enter **AR** next to the availability map to which you want to add a resource or service.

   The Attach Resources panel appears.

2. Enter **S** next to the resource or service that you want to add to the availability map.

   The Attach Resources Results panel appears, which tells you whether the operation was successful.

3. Press F3 (File).

   The Availability Map List appears.

   Note: To display the resources or services that are attached to an availability map, enter **LR** next to the availability map in the list.

To attach a service or resource to a map while you are working on the definition

1. Select the General Description panel.
2. Enter the name of the availability map in the Availability Map field, and press F3 (File).

   The details are saved.

Detach Service or Resource Definitions from an Availability Map

You can detach service or resource definitions from an availability map (for example, if you want to change a resource definition and test it separately).

You can detach a service or a resource from an availability map by using:

- The Availability Map List
- The service or resource definition panels

To detach a service or resource from an availability map from the Availability Map List

1. Enter **/SADMIN.A** (for services) or the **/RADMIN.A** (for resources) at the prompt.

   The Availability Map List panel appears.

2. Enter **LR** next to the map from which you want to detach services or resources.

   A list of the attached services or resources appears.

3. Enter **DT** next to the services or resources that you want to detach from the map and press Enter.

   The services or resources are detached from the map.

To detach a service or a resource from a map while you are working on the definition

1. Select the General Description panel.
2. Remove the name of the availability map from the Availability Map field and press F3 (File).

   The service or resource is detached from the map.

Maintenance of Availability Map Definitions

You can browse, update, copy, and delete timer information and availability map definitions from the Availability Map List panel.

Chapter 13: Implementing Services

This section contains the following topics:

Services

A service is a collection of resources that support a business or operations function. After you have defined the resources, you group relevant resources in service definitions. You use service definitions to specify the service availability requirements of your organization.

Note: You can define and manage services from focal point regions only. Services are not visible in subordinate regions, but you can include resources managed by a subordinate region in a service.

Resource Management Using Services

By monitoring services instead of the resources, you can tell directly whether the availability of a business function is being satisfied. During automated operation, the availability of the resources is determined by the services to which they belong. If a service needs to be active, it requires its resources to be active also.

Services enable you to define the operations policies for groups of resources used by your organization. By using service definitions, you can specify the following characteristics of a service:

- Over what periods of time the service must be available.
- The services and resources (the members) required for the provision of the service. Services can share members and can include resources from different systems.
- How a member status affects the provision of the service.

Active service definitions in the knowledge base determine the availability of services.

You define the availability of a service by specifying the desired state of the service at particular times. In a multisystem environment, you can specify a system as the *service automation focal point system*, and the scheduled times refer to the local times of that system.

The region uses the defined availability of the service, the defined relationships between the members, and the defined operations policies and methods of the members to maintain the availability of that service.

When a service starts, it requires that all its members be started too, irrespective of their desired states. When a service stops, it removes its availability requirements from its members.

Service Definitions

When you have populated the system images with resources, you can define the services provided by them. A service comprises resources and other services as members. A service can include resources that reside on other systems.

You specify how the status of a member affects the provision of the service. The failure of some members can stop a service, while the failure of other members might only degrade the service.

You specify the availability of the service by using an availability map. The availability is defined in terms of the time on the service automation focal point system. In a multisystem environment, you specify a system as the focal point system, and the scheduled times refer to the local times for that system. If the map schedules the starting of processes, the processes are started in the region on that system only.

Access Service Definitions

Service definitions are stored in the knowledge base in a structure similar to that of resource definitions. Service definitions belong to the service system image, $SERVICE. Version 0001 of this image is always active. The definitions have a class of SVC.

To access service definitions, enter **/SADMIN.S** at the prompt.

The ServiceView : Service List panel appears. The panel lists the services in the knowledge base.

Note: To assist with maintenance of your service definitions you can create backup versions of the $SERVICE 0001 service image.

Service Definition Panels

You can use variables as data in a service definition.

To add a service definition, press F4 (Add) from the Service List panel. A Service General Description panel appears. You define the service by entering data on the following panels:

Service General Description

You must complete this panel. The panel enables you to identify the service, specify the service operation mode, and define the availability requirements for the service.

Service Filters

Complete this panel. The panel enables you to select members for the service and specify how important a member is to the service.

State Thresholds

The panel enables you to define how the statuses of the service members affect the status of the service.

State Change Exits

The panel enables you to specify state change exit processes that are invoked if the service changes to a given state.

Automation Log Details

The panel enables you to change the logging requirements.

Owner Details

The panel enables you to identify up to two people who can be contacted if the service has operational problems.

Extended Function Exit

The panel enables you to specify an exit NCL procedure that can be used to extend the service functions provided in the region.

General Description

The Service General Description panel specifies the service name, the operation mode, a description of the service, and the availability map to apply.

Operation Modes

Specify an operation mode of AUTOMATED, IGNORED, MANUAL, OFF, or STARTAUTO. During operation, the global operation mode can restrict the mode specified in the Operation Mode field.

The operation modes have the following effects on a service:

AUTOMATED

Specifies that the region monitors and automates the control of the service.

When the desired state of the service is set to ACTIVE, the service places an ACTIVE desired state override on its members. The region then determines the actual state of the service from the actual states of the members.

When the desired state of the service is set to INACTIVE, the service removes the ACTIVE desired state overrides from its members. The service acquires an INACTIVE actual state immediately.

IGNORED or MANUAL

Specifies that the region monitors but relinquishes control of the service to the operators. A service in the IGNORED mode always appears green on your monitors.

When the desired state of the service is set to ACTIVE, the service does not place the ACTIVE desired state overrides on its members. The overrides occur only when an operator starts the service manually by using the A(ctivate) command.

Similarly, setting the desired state of the service to INACTIVE does not affect the members. The members are affected only when an operator stops the service manually by using the T(erminate) command.

OFF

Specifies that the region does not monitor or control the service. The definition remains in the knowledge base, but the service does not appear on your monitors.

STARTAUTO

Specifies that the region starts the service in the AUTOMATED mode. As the service achieves its desired state, the region switches the service to MANUAL mode.

How You Specify the Availability of the Service and Its Members

You can use an availability map to define the changes to the normal availability of the service.

In a multisystem environment, you specify a system as the service automation focal point system. The scheduled times refer to the local times on that system. If the map schedules the starting of processes, the processes are started in the region on that system only.

To attach an existing map, enter the name of the map in the Availability Map field. Press F10 (Edit Map) to update the timer details.

Note: You can create a map from the service definition. You can name a new map and define it, or access an existing map, change the name, and update the copy. The map is created in the knowledge base when you save the definition.

The availability of a service overrides the availability of its members. If a member always operates as part of a service, you can let the service handle the availability of that member.

To let the service control the availability of a member, set the desired state of the member to INACTIVE:

- If the default desired state is ACTIVE, attach the member to an availability map that specifies an INACTIVE desired state for all times.
- If the default desired state is INACTIVE, you do not need to attach an availability map to the member. The member is desired inactive by default.

Important! The default desired state is set in the AUTOIDS parameter group during region initialization. After it is set, do not change the default because the availability requirements defined in the knowledge base are based on this default. The Parameter Groups panel lists the region parameter groups. Enter the **/PARMS** shortcut to access the panel.

How Service Activation Works

When the service is required to become active, it desires the member to become active, overriding the desired state set by the availability map or by default.

When the desired state of a service changes from INACTIVE to ACTIVE, all its members need to be started. The region places an ACTIVE desired state override on each of the service members. If any of the members are inactive, the region starts those members. The service is indicated as active when a predefined percentage of its members are active.

How Service Inactivation Works

When the desired state of a service changes from ACTIVE to INACTIVE, this service is indicated as inactive immediately. The region removes the ACTIVE desired state overrides from the service members. The members might or might not become inactive, depending on, for example, whether they are required by other services.

Service Status

The status of a service with a desired state of ACTIVE is determined by the status of its members. The failure of some members can stop a service, while the failure of other members might only degrade the service. You can distinguish between a failed service and a degraded service and respond accordingly. You specify the importance of a member to the service in the service definition.

Select Service Members

To select service members

1. From the General Description panel, press F8 (Forward).

   The Service Filters panel appears. This panel defines the filters that select the members of the service.

2. Define the filters by specifying the following criteria:

   - The service class (SVC, if the member is another service) or resource class in the Class field.
   - The name of the member in the Name field. You can use the following wildcard characters:
     - The underline character (_) represents a single character. For example, PROD_X3A matches PROD1X3A, PROD2X3A, …
     - The percent character (%) represents zero or more characters. For example, PROD%X3A matches PRODX3A, PROD1X3A, PROD2X3A, …
     - You can also use the asterisk (*) as a wildcard character. An asterisk behaves the same way as the % character, but you cannot have the * at the beginning of or embedded in the specified value. For example, * and PROD* are valid values.
   - The SMF ID of the system that owns the member in the SMF ID field. The default is the SMF ID of the local system.

     When you have resources with the same identification defined on different systems and you want to include all those resources as members, specify ***** in the SMF ID field.

- The type of resource (as specified in the resource definition) in the Type field. You can use the asterisk (*) wildcard character by itself or at the end of the specified value.

  Note: The Type field is irrelevant for a service. Leave the value to the default.

- A weight that indicates how important the member is to the service in the Weight field.
- The type of weight in the Weight Type field.

You can define up to 97 lines of members.

Weight of a Service Member

The weight indicates how important a member is to the service. The valid values are 0 percent through 100 percent.

If the weight is 100 percent, the actual state of the member affects the actual state of the service directly. For example, if the member fails, the service fails.

If the weight is 0 percent, the member has no effect on the service.

If the weight is between 0 percent and 100 percent, the effect of the member on the service depends on the state thresholds (see page 143).

You can apply the following types of weights to service members:

Fixed Weight

With a fixed weight, every member included in a line entry has the weight specified in the Weight field.

In the following examples, the weight is 100 percent fixed:

- If the line entry includes only one member (for example, the PRODA started task on the EASTTEST 0001 system), the member has 100 percent weighting.
- If the line entry includes more than one member (for example, the PRODA started tasks on all the connected systems (SMF ID=*)), each member has 100 percent weighting.

Proportional Weight

You can use the proportional type of weight when the line entry includes more than one member. With a proportional weight, every member included in the line entry has an equal proportion of the weight specified in the Weight field. For example, if the weight is 100 percent proportionally applied to two members, each member has 50 percent weighting in the service.

View the Service as Defined by the Service Filters

The service filters select the members for a service. Only members defined in active system images are selected. The members can change if the active system images change (for example, when a connected region has a different system image loaded).

To view the members in a service, press the F5 (Model) function key.

State Thresholds

From the Service Filters panel, press F8 (Forward) to go to the State Thresholds panel. Use this panel to define how the actual states of the members affect the actual state of the service.

The actual state of a service can be *one* of the following:

- UNKNOWN
- FAILED
- ACTIVE
- STARTING
- DEGRADED

You must assign a threshold to the first four states.

Thresholds are evaluated in the order shown. The service takes on the state of the first threshold equaled or exceeded, irrespective of whether other thresholds are Equaled or exceeded. For each actual state, you specify a percentage threshold value that, if equaled or exceeded, causes the service to take on that state (unless a state of higher severity has also satisfied its threshold requirement). This threshold is expressed as a combined weight of the members required to deliver the service.

Each member of the service has a weight associated with it. The weight expresses the level of impact the individual resource has on the threshold calculation for the actual state of the service.

If members are not ACTIVE, you can use their logical state to calculate the threshold for the actual state of the service. In this case, if a member has a logical state of OK, its weight is added to the combined weight for the ACTIVE state. If a member has a logical state of UNKNOWN or STARTING, their weight is added to the combined weight for the corresponding actual state. If a member has any other logical state, their weight is added to the combined weight for the FAILED actual state.

Note: If a service filter finds no members, the weight specified in the Weight column on the Service Filters panel is added to the combined weight for the UNKNOWN state.

Using the logical state rather than the actual state to calculate the threshold has advantages. For example, you can shut down a resource that is part of a service without affecting the service. The service sees a logical state of OK, even though the resource is INACTIVE, and treats it as though it is ACTIVE. Alternatively, when a resource fails and you set it to IGNORED, the service sees the resource as ACTIVE (OK), and the service continues unaffected.

State Change Exits

From the State Thresholds panel, press F8 (Forward) to scroll forward to the State Change Exits panel. This panel lets you specify the following types of exit processes:

- A process that executes before the service is started. By using this feature, you can add your own preactivation tasks to the internal service starting method.
- Processes that execute on specified state changes. For example, if a service fails, you can invoke a procedure that writes a problem report. You can specify a process to execute on changes to the actual state, the desired state, or the logical state of the service.

In a multisystem environment, you can specify whether the processes are executed in a specific region only or in all connected regions.

Define the Logging Details

From the State Change Exits panel, press F8 (Forward) to scroll forward to the Automation Log Details Panel. This panel contains the following information:

- Size of the temporary log for the service (known as a *transient log*)
- Destination of the logged information
- Type of information logged

Owner Details

From the Automation Log Details panel, press F8 (Forward) to scroll forward to the Owner Details panel. This panel lets you identify up to two people who can be contacted if this service has operational problems.

Extended Function Exit

From the Owner Details panel, press F8 (Forward) to scroll forward to the Extended Function Exit panel. The panel lets you provide additional operator functions. Specify the exit NCL procedure that provides these functions. The procedure is invoked when an operator issues the XF command against the service.

The extended function exit NCL procedure has access to variables that contain all of the service details with the prefix ZRM.

Maintenance of Service Definitions

You can browse, update, copy, and delete service definitions from the Service List panel.

Note: If you only want to hide a service definition from the region, set the operation mode to OFF. The definition remains in the knowledge base but is not used.

Back Up Service Definitions

To assist you with the maintenance of your service definitions, you can create backup versions of your service image. By creating a backup version of your service image or definitions, you can perform the following:

- Update service definitions in any version of a service image
- Restore a service definition from a backup service image
- Merge two versions of a service image

To create a backup version of a service image

1. Enter **/SADMIN.SI** at the prompt.

   The service image list appears.

   Note: If you have not created a backup before, there is only one service image listed: $SERVICE 0001.The active service image can be $SERVICE 0001 only. $SERVICE 0001 cannot be deleted.

2. Enter **C** next to the service image you want to copy.

   The ServiceView : Service Image Definition panel appears.

3. Enter a new Database Version, Short Description, and (optionally) a Long Description.

4. Press F3 (File).

   A copy in progress panel opens while the copy occurs. The Service Image List appears with the backup version displayed in the list.

Update Service Definitions in a Backup Service Image

You can access a list of all the service definitions in any version of a service image. From this list you can update any service definitions contained in the service image.

To update a service definition in the $SERVICE 0002 backup image

1. Enter **L** (List Services) beside the $SERVICE 0002 service image in the Service Image List.

   The ServiceView : Service List panel appears showing the service definitions in that service image.

   Note: You can access the list of service definitions for another version of the service image by changing the version number on the Service Image List panel and pressing Enter.

2. Enter **U** beside the service definition that you want to update.

   The ServiceView : Panel Display List appears for that service definition.

3. Update the service definition, as required.

4. Press F3 (File) to save the changes.

   The ServiceView : Service List panel appears.

Restore a Service Definition from a Backup Service Image

If you have made changes to a service definition and you are having trouble with its implementation, you can restore it from a previous version of the definition.

To restore service definition SERV01 from $SERVICE 0002 service image to $SERVICE 0001

1. Enter **C** beside SERV01 in the ServiceView : Service List.

   The ServiceView : Service Image Definition panel appears.

2. Change the Database Version from 0002 to 0001 and press F3 (File).

   Because there is already a copy of the service in the target service image, the Confirm Copy Replace panel appears.

3. Press Enter to confirm the replace or F12 (Cancel) to cancel the request.

   The ServiceView : Service List panel appears.

Merge Two Service Images

You can merge two service images and replace the active image with a backup version.

To merge service images $SERVICE 0002 and $SERVICE 0001

1. Enter **C** beside the $SERVICE 0002 on the ServiceView : Service Image List panel.

   The ServiceView : Service Image Definition panel appears.
2. Change the Database Version number to 0001and press F3 (File).

   The Confirm System Image Merge panel appears.
3. Enter **YES** in the input field if you want to overlay like-named components.
4. Press F6 (Confirm).

   The service images are merged.

Chapter 14: Implementing the Graphical Monitor

This section contains the following topics:

Graphical Monitor

The graphical monitor presents the status of resources in icons on an icon panel.

You customize the graphical monitor by using icon panels. You can change the icon panel to obtain a different view of the monitored systems and networks. By zooming (Z) in on an icon, you can selectively view the group of resources that it contains.

The graphical monitor monitors groups of resources as a single entity.

How You Customize the Graphical Monitor

To customize the graphical monitor, you define resource groups, icons, and icon panels. You arrange icons on icon panels and attach resource groups to the icons so that each icon on the panel represents a group of resources. After you generate an icon panel, an operator can use that panel to customize the graphical monitor.

You generate an icon panel as follows:

1. Define the required resource groups.
2. Define the icons to use on an icon panel.
3. Define the icon panel.
4. Place the defined icons on the panel and attach resource groups to them.

When you save a resource group, icon, or icon panel definition, or generate an icon panel description file, it propagates to all the connected regions. That is, the definition of the generated icon panel is global.

Resource Groups for Icons

A resource group represents a group of resources that you have defined in the knowledge base. To define a resource group, use *one* of the following methods:

- **Specify an Icon Panel**

  The panel displays icons representing other resource groups. Use the Zoom Icon Panel Definition panel to specify the icon panel.

- **Specify a Group of Resources**

  You can identify up to 16 resources by class and name. Thus, the identified resources are independent of system images. In a multisystem environment, the specified class and name points to resources in all the system images that are loaded in the linked regions. You can, however, specifically exclude remote resources. Use the Resource Filter Definition panel to specify the resources to group.

- **Specify a Resource Group Filter**

  A resource group filter uses a Boolean expression to define a group of resources. You group the resources by their static attributes such as names and parent system images. Use the Resource Group Filter Definition panel to define the Boolean expression.

Access Resource Group Definitions

The Resource Groups List displays the list of resource group definitions in the knowledge base. You can add a new definition, or browse, update, copy or delete an existing definition.

To access resource group definitions, enter **/GADMIN.G** at the command prompt.

The Resource Group List appears.

Add a Resource Group Definition

To add a resource group definition

1. Enter **/GADMIN.G** at the prompt.

   The Resource Group List appears.

2. Press F4 (Add) to add a group definition.

   The Resource Group Definition panel appears.

   Note: If you change your mind and do not want to add the group, press F12 (Cancel) to cancel the operation any time before Step 6.

3. Complete the Name and Description fields to identify the new group.
4. Select *one* of the following options to define the group:
   - Select option A to specify an icon panel.

     The Zoom Icon Panel Definition panel appears. Proceed to Step 5a.
   - Select option B to specify a group of resources by class and name.

     The Resource Filter Definition panel appears. Proceed to Step 5b.
   - Select option C to specify a resource group filter.

     The Resource Group Filter Definition panel (see page 152) appears. Proceed to Step 5c.

   Note: Options B and C are related. You can use option B to specify the services and resources in the group directly. If you then select option C, the specification defined by using option B is expanded into a Boolean expression.
5. Depending on the option you select, proceed as follows:
   a. Specify the name of a generated icon panel in the Zoom Icon Panel Name field. You can enter a question mark (?) in the field to access the icon panel prompt list from which you can select the required panel.

      After you specify the name, proceed to Step 6.
   b. Identify the resources by class and name in the ClassDsc and Resource Name fields. You can enter a question mark (?) in the fields to access the resource class and resource name prompt lists from which you can select the required class and name.

      If you want to exclude the resources from remote systems, specify **Y** (yes) in the Exclude Remote System Resource field. The default is NO.

      After you identify the resources, proceed to Step 6.
   c. Press F10 (EditFltr) to edit the filter. See the online help for a description of the fields.

      Specify the Boolean expression (see page 152) in the Filter Expression window to define the filter.

      Press F3 (OK) to exit the edit mode, then proceed to Step 6.
6. Press F3 (File) to file the new definition when you finish defining the group.

Resource Group Filter Definition Panel

The Resource Group Filter Definition panel specifies the details of a resource group.

The panel displays two windows. The Filter Definition window identifies the filter, and the Filter Expression window specifies the Boolean expression of the filter.

Example: Resource Group Filter Definition Panel

This example defines a group that contains all started tasks except those resources with a name of PCICS1.

```
PROD1------ Automation Services : Resource Group Filter Definition ------------
Command ===>                                                  Function=UPDATE

. Filter Definition ------------------------------------------------------------.
| Name ......... $ICRSRC                                                       |
| Description .. RESOURCE GROUP "RSRC" DIRECT FILTERING                        |
| Last Updated at 17.11.27 on SUN 06-FEB-2011 by USER01                        |
'------------------------------------------------------------------------------'
. Filter Expression ------------------------------------------------------------.
|                                                                              |
|      "(" Field    Opr Value                                  Gen ")" Bool    |
|      (   CLSNAME  EQ  "STC"                                          AND     |
|          NAME     NE  "PCICS1"                                   )           |
|     **END**                                                                  |
```

Resource Group Filter Expression

Use the Filter Expression window on the Resource Group Filter Definition panel to specify the Boolean expression that defines the filter. The expression uses resource attributes to determine what belongs to the group.

Use the following action codes to help you enter the expression:

D (Delete)

Deletes the selected line.

I (Insert)

Inserts a blank line after the selected line.

R (Repeat)

Repeats a selected line.

Maintenance of Resource Group Definitions

You can browse, update, copy, and delete group definitions from the Resource Group List panel.

Note: During an update, if the resources in the resource group are specified by using option C, you have no access to option B.

Except as noted above, you can change the method of definition during an update. Saving a definition by a new method automatically overrides the definition by the current method.

Icons

An icon is a graphic that you can use to represent resource groups on the graphical monitor. You use icons to build icon panels. You position one or more icons on a panel and attach resource groups to the icons, one group for each icon. When used, an icon displays a status determined by the status of the underlying group members. An operator can zoom in on an icon using the Z (Zoom) command. This action displays another icon panel or a group of resources in the Status Monitor, as determined by the attached resource groups. Use the Icon Editor to define an icon.

Access Icon Definitions

To access icon definitions, enter **/GADMIN.I** at the prompt.

The Icon List panel appears.

The panel displays the list of icon definitions in the knowledge base. You can add a new definition, or browse, update, copy or delete an existing definition. You can also edit a definition from the Icon Panel Generator panel.

Define an Icon

You use icons to build the panel for your graphical monitor.

To define an icon

1. Enter **/GADMIN.I** at the prompt.

   The Icon List panel appears.

2. Press F4 (Add).

   The Icon Editor panel appears.

3. Complete the following fields:

   Name

   Specifies the name of the icon.

   Description

   Describes the icon.

   Icon Height

   Specifies the height of the icon in lines.

   Icon Width

   Specifies the width of the icon in characters.

   Note: If you change the default size, press Enter to update the shape of the icon in the Edit Area window.

   Specify the values you want to display (see page 156) on the icon.

4. Press F3 (File).

   The new definition is saved.

Icon Editor Panel

The Icon Editor panel specifies the details of an icon. The operation you are performing is displayed at the top right of the panel, for example, Function=UPDATE.

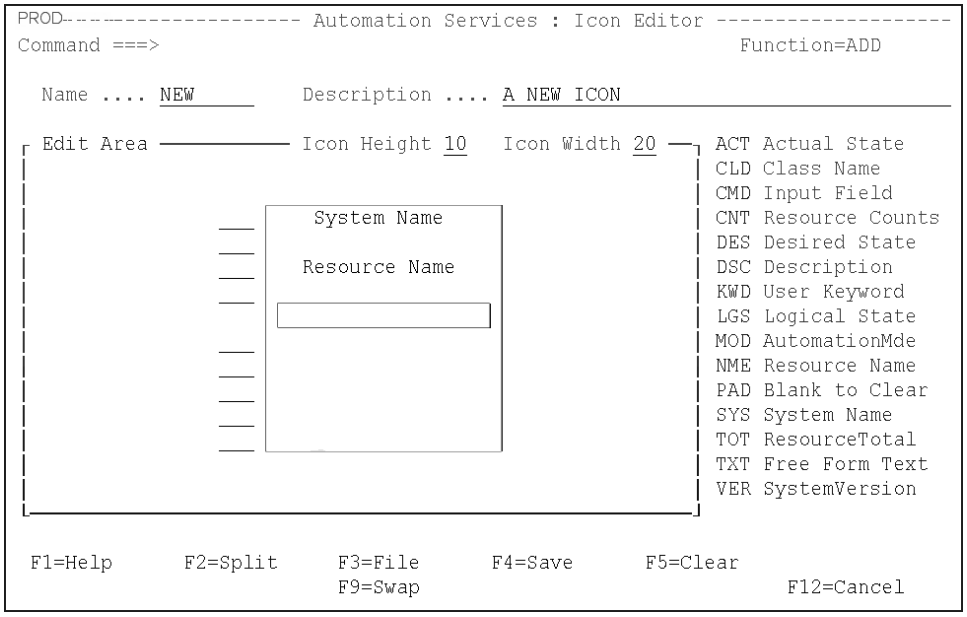
The panel specifies the following information:

- Name and description of the icon
- Size of the icon (height and width)
- Actual icon representation

The Edit Area window specifies the values you want to display on the icon.

Example: Icon Editor Panel

This example defines a resource icon.



How You Edit the Icon

Use the Edit Area window on the Icon Editor panel to specify what you want to display on the icon.

The icon contains the number of lines specified in the Icon Height field. Use the three-character codes listed to the right of the Edit Area window to specify the values you want displayed on the icon. To use a code, enter the code in a line field. You can use the code on any line, irrespective of whether the line is blank or not. Except for the TXT code, executing a code on a line overrides what is already there.

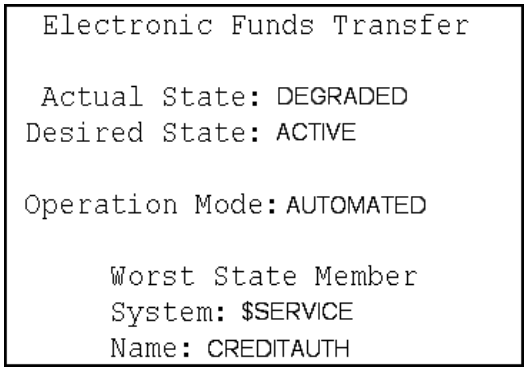
You can type codes in more than one line field, then press Enter to execute the codes.

Pressing F5 (Clear) clears the icon. Use the PAD code to clear a line.

Note: For information about the codes, see the online help.

Icon Definition Example

In this example, an icon, EFTPOS, is defined for the group of services and resources that support electronic funds transfer. The finished icon as it appears to an operator is shown in the following figure:



To define the icon

1. Enter **/GADMIN.I** at the prompt.

   The Icon List panel appears.

2. Press F4 (Add).

   The Icon Editor panel appears.

3. Enter **EFTPOS** in the Name field and a description in the Description field, for example, Electronic funds transfer.

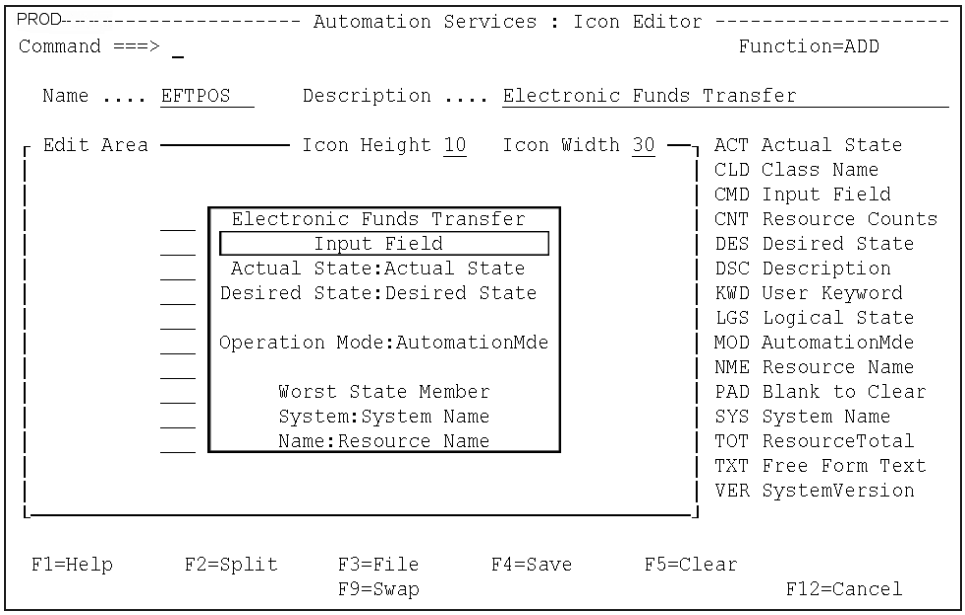
4. You want the icon size to be 10 lines by 30 characters. Change the icon width to 30, and press Enter to update the shape of the icon.

5. Enter **TXT** in the first line field. A text field appears in the icon.

6. Enter **ELECTRONIC FUNDS TRANSFER** in the text field.
7. Enter **CMD** in the second line field to create the command entry field.
8. Enter **ACT** in the third line field.
9. To add the description to the field, enter **TXT** in the line field. Enter **Actual State** in the displayed text field. A colon is inserted automatically after the entered text.
10. Repeat steps 8 and 9, using the appropriate codes, for the following information:
    - Desired state (line 4)
    - Operation mode (line 6)
    - Identity of the group member that has the worst logical state (lines 9 and 10)

    Use steps 5 and 6 to enter the Worst State Member label (line 8).
11. Press F3 (File) to file the definition when you finish with the panel.

The following shows the completed Icon Editor panel.



Maintenance of Icon Definitions

You can browse, update, copy, and delete icon definitions from the Icon List panel.

Icon Panels

An icon panel defines what is displayed on the graphical monitor. You arrange icons on the panel and attach resource groups to the icons.

You can define your own icon panel or select one of the predefined panels provided with your product.

When you create an icon panel, you create an icon panel definition and the icon panel description file. An operator uses the panel to customize the graphical monitor. You can generate an icon panel (that is, the description file) only if all the icons on the icon panel definition have attached resource groups. Use the Icon Panel Generator to define and generate the icon panel.

Important! Icon panels defined on a 3270 Model 4 or equivalent terminal cannot be used on Model 3 and Model 2 terminals. Icon panels defined on a Model 3 terminal cannot be used on Model 2 terminals.

Access Icon Panel Definitions

To access icon panel definitions, enter **/GADMIN.P** at the prompt.

The Icon Panel Definition List panel appears. The panel displays the list of icon panel definitions in the knowledge base. You can add a new definition, or browse, update, copy, or delete an existing definition.

Define an Icon Panel

When you define an icon panel, you can create a new panel or select a pre-defined panel. A default panel is distributed for your product; however, if you have installed more than one product in your environment, $RMDYNAMIC is your default icon panel.

Note: $RMDYNAMIC is the default icon panel when more than one product is present in a region. It dynamically displays one icon per product found on the region. As such, it is different to other icon panels and should not be edited or regenerated by users. If it is regenerated in error, panel $RMDYNAMICBU is available in the ICOPANL data set to use to recover $RMDYNAMIC.

To define an icon panel

1. Enter **/GADMIN.P** at the prompt.

   The Icon Panel Definition List panel appears.

2. Do *one* of the following:

   - Press F4 (Add) to add a new icon panel definition.

     The Icon Panel Generator Initial Help panel appears.

   - Select one of the pre-defined defaults for your product.

     The Icon Panel Generator Initial Help panel appears.

   Note: Pressing F4 (Remove Help Screen) exits and removes permanently the help panel. That is, the help panel does not appear the next time you work on an icon panel definition.

3. When you finish reading the help text, press Enter.

   The Icon Panel Generator panel appears.

   If you selected one of the pre-defined defaults, go to Step 5.

   If you are defining a new icon panel, go to Step 4.

4. Complete the following fields:

   Name

   Specifies the name of the icon panel.

   Description

   Describes the icon panel.

5. Use the function keys to create or edit your panel. The left limit of the icon placement area is column 2, and the top limit of the icon placement area is row 5. The right and bottom limits are dependent on the size of your screen and the width and height of the icon.

6. Press F3 (File).

   The new icon panel is generated.

   Note: If an icon in the panel definition does not have an attached resource group, you cannot generate the new panel. A message is displayed on your screen to this effect. You can either attach any missing resource groups so that you can generate the panel or press F3 (File) again to file the definition without generating the panel.

Icon Panel Generator Panel

The Icon Panel Generator panel specifies the details of an icon panel. The operation you are performing is displayed at the top right of the panel.

The panel specifies the following information:

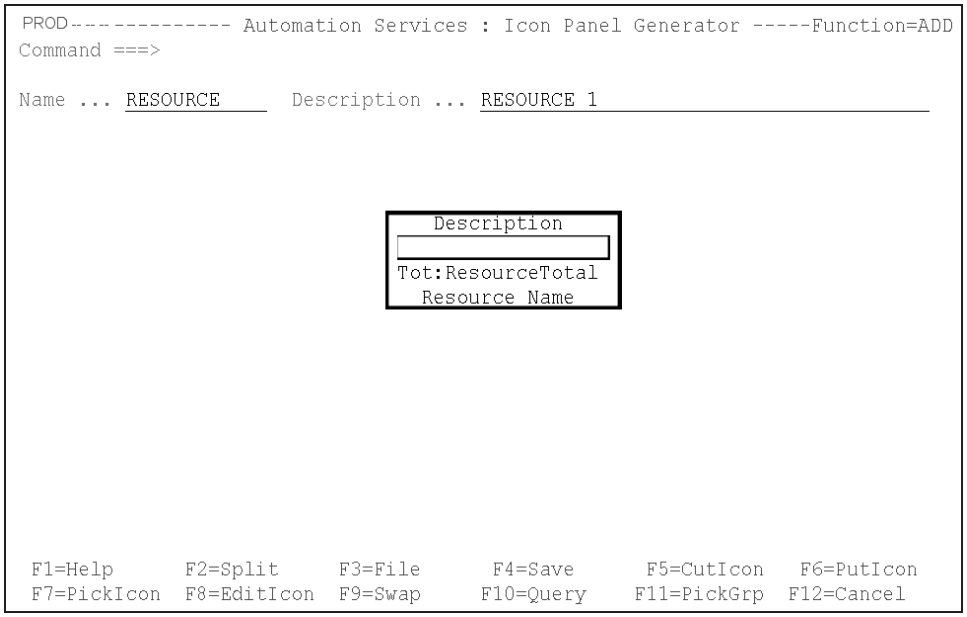
- Name and description of the icon panel
- Actual icon panel representation

The area from Column 2 to the right and from Row 5 down contains the icons you want to display on the graphical monitor.

Use the function keys on the Icon Panel Generator panel to specify what you want to display on the graphical monitor.

Example: Icon Panel Generator Panel

This example defines a panel with one icon.



Add an Icon to the Icon Panel

To build an icon panel for your graphical monitor, you add icons to the panel.

To add an icon to the icon panel

1. Move the cursor to fix the position of the top left corner of your icon. You must place the cursor in an area not already occupied by another icon.
2. Press F7 (PickIcon) to display the list of defined icons.

   The Icon List panel appears.
3. Enter **S** beside the icon you want to add to the icon panel.

   The Icon Panel Generator panel appears. The selected icon is positioned with its top left corner at the cursor.

   Note: After you pick an icon, you can move the cursor to another position and press F6 (PutIcon) to duplicate the icon on the icon panel. You can thus quickly position multiple icons with the same attributes on the panel.
4. Press F11 (PickGrp) to attach a resource group to the icon.

   The Resource Groups List panel appears.
5. Enter **S** beside the group you want to attach to the icon.

   The Icon Panel Generator panel appears. You have added an icon with an attached resource group to the icon panel.

Attach a Resource Group to an Icon on the Icon Panel

You can attach resource groups to icons on the Icon Panel Generator panel. You can change a resource group attachment by attaching another group to the icon.

To attach a resource group to an icon on the icon panel

1. Move the cursor in the icon to which you want to attach a resource group.
2. Press F11 (PickGrp).

   The Resource Groups List panel appears.
3. Enter **S** beside the group you want to attach to the icon.

   The Icon Panel Generator panel appears.

Duplicate an Icon on the Icon Panel

Note: Duplicating an icon on the icon panel copies only the icon, not the attached resource group.

To duplicate an icon on the icon panel

1. Move the cursor inside the icon you want to duplicate.
2. Press F7 (PickIcon).

   The icon is highlighted
3. Position the cursor to where you want to place a copy of the icon and press F6 (PutIcon).

   The icon is placed at the cursor.

   Note: The cursor position fixes the top left corner of the duplicate icon.

Move an Icon on the Icon Panel

To move an icon to another position on the icon panel

1. Move the cursor in the icon you want to move.
2. Press F5 (CutIcon).

   The selected icon is no longer displayed.
3. Move the cursor to fix the position of the top left corner of the icon being moved and press F6 (PutIcon).

   The icon appears at the position of the cursor.

Edit an Icon on the Icon Panel

You can edit an icon from the Icon Panel Generator panel. Editing enables you to update the original icon or create a new copy of the icon.

Updating an icon from the Icon Panel Generator panel updates the icon definition in the knowledge base and the selected icon only. If there are other icons in the panel definition that use the same icon definition, these other icons are not updated as long as you remain in the panel definition. You can, therefore, have several versions of the same icon in the panel definition. When you generate the icon panel, the panel reflects these different versions of the icon (even though there is only one version of the icon definition).

Note: Although a generated icon panel can retain different versions of the same icon, the icon panel definition cannot. The next time you access the panel definition, the definition reflects the latest version of the icon.

To edit an icon on the icon panel

1. Position the cursor in the icon you want to edit.
2. Press F8 (EditIcon).

   The Icon Editor panel appears.
3. Edit the icon, as required.

   Note: If you want to create a new copy of the icon, change the value in the Name field.
4. Press F3 (File).

   The updated definition is saved and the Icon Panel Generator panel appears.

Display Information About an Icon on the Icon Panel

You can display the name of and the resource group attached to an icon on the icon panel.

To display the information, press F10 (Query).

A message displays the required information.

The following example identifies the icon as CVNEW with an attached resource group named ACREC:

```
RM810017 ICON=CVNEW RESOURCE GROUP=ACREC
```

Delete an Icon from the Icon Panel

To delete an icon from the icon panel

1. Position the cursor in the icon you want to delete.
2. Press F5 (CutIcon).

   The selected icon is deleted.

   Note: The CutIcon action temporarily stores the icon that is removed from the icon panel; however, the icon is lost if you use the F7 (PickIcon) or F5 (CutIcon) function key on another icon.

Maintenance of Icon Panel Definitions

You can browse, update, copy, and delete icon panel definitions from the Icon Panel Definition List panel.

Note: You cannot update the definition of an icon panel that a graphical monitor is using.

If an icon in the panel definition does not have an attached resource group, you cannot generate the panel. A message is displayed on your screen to advise you of the fact. You can either attach any missing groups so that you can generate the panel or press F3 (File) again without generating the panel.

How You Edit a Generated Icon Panel

To update an icon panel, you can regenerate the panel by using an updated definition or you can edit the panel description file directly.

Enter the **/GADMIN.E** path to access the list of icon panels. The Panel List panel appears.

The panel displays the list of icon panels in the knowledge base. Some of these panels are generated using icon panel definitions; some of these panels are created by users (for example, by using the Copy or Rename action). If an icon panel definition generates the panel, the Name and Description columns reflect the name and description of the definition.

Consider the following when you edit an icon panel description file:

- If you regenerate an icon panel by using the P - Define Icon Panels option, you lose whatever editing you did in the description file. Use the R action to rename the panel before editing.
- The first line in a description file is the panel description, as displayed on the panel list.
- The #NOTE #ICON statement in a description file associates the specified resource groups with the icon panel.

Note: For information about panels and panel statements, see the *Network Control Language Programming Guide*.

Set Up Default Icon Panel for Your Users

You can add an icon panel to a user profile so that it is displayed automatically each time that user accesses the graphical monitor.

To add an icon panel to a user profile

1. Enter **/ASADMIN.UP** at the prompt.

   The User Profile List appears.

2. Select the user profile.

   The Panel Display List appears.

3. Select Graphical Monitor Profile.

   The Graphical Monitor Profile panel appears.

4. Complete the following field:

   Panel Name

   Specifies the name of the icon panel that you want to appear.

   Note: You can enter ? to display a selection list of icon panels.

5. Press F3 (File).

   The details are saved.

Chapter 15: Setting Up the Alert Monitor

This section contains the following topics:

Access Alert Administration

Alert Monitor administration lets you define Alert Monitor interfaces, filters, and formats that apply to all users.

You perform Alert Monitor administration functions from the Alert Monitor : Administration Menu.

To access Alert Monitor administration functions, enter **/ALADMIN** at the prompt.

The Alert Monitor : Administration Menu appears.

```
PROD---------------- Alert Monitor : Administration Menu --------------/ALADMIN
 Select Option ===>

    I   - Define Trouble Ticket Interface                    ALTTI
    D   - Define Trouble Ticket Data Entry                   -
    F   - Define Filters                                     ALFILT
    L   - Define List Formats                                -
    MIF - Invoke Alert Filter Migration Utility              -
    ST  - Alert Monitor Self Test                            ALTEST
    X   - Exit
```

Alert Monitor Trouble Ticket Interface

The Alert Monitor provides an interface that lets you send alert information in the form of a *trouble ticket* to another interface automatically or manually.

The Alert Monitor supports the following interfaces for raising trouble tickets:

Electronic Mail

Sends an email describing the problem to a problem management application or a particular person. This method can be used to send tickets to multiple problem management applications.

Custom

Lets you write your own NCL procedure to deliver the trouble ticket to an application by whatever means you choose. For example, you can do the following:

- Invoke a REXX procedure, and pass alert variables.
- Send to any external interface, for example, problem-management product.
- Send to MVS system facilities, for example, system console, data sets, SMF user records, or batch jobs.
- Invoke applications, for example, FTP.

Service Desk

Creates a new CA Service Desk request from the alert details.

Note: If your CA Service Desk installation is configured with the optional ITIL application, incidents are created instead of requests.

Note: You can choose one interface only.

If you want the operator to supply information when requesting the creation of a ticket, you also need to set up the trouble ticket data entry definition.

Define a Trouble Ticket Interface

If you want to enable operators to raise trouble tickets on alerts, you must define the trouble ticket interface.

To define a trouble ticket interface between the Alert Monitor and another application

1. From the Alert Monitor Administration Menu, select option **I** - Define Trouble Ticket Interface.

   The Alert Monitor : Interface Definition panel appears.

2. Enter the type of interface that you want to define in the Interface Type field.

   Note: To obtain a selection list of valid values, enter **?** in this field.

3. Press F6 (Action).

   A panel appears where you can define an email (see page 169), custom (see page 171), or CA Service Desk (see page 172) interface. The type of panel displayed varies, depending on the interface type that you specified.

Define an Email Trouble Ticket Interface

This option enables alert details to be sent using email.

Note: To enable this option, you must ensure that your Systems Programmer enables SMTP support on this region's TCP/IP stack.

To define an email trouble ticket interface

1. Enter **/ALADMIN** at the prompt.

   The Alert Monitor : Administration Menu appears.

2. Select option **I** - Define Trouble Ticket Interface.

   The Alert Monitor : Trouble Ticket Interface Definition panel appears.

3. Enter **EMAIL** in the Interface Type field, and press F6 (Action).

   The Email a Trouble Ticket panel appears.

4. Leave the &$USRNAME variable in the Mail Address field. The variable works with the default trouble ticket data entry definition (see page 174) to specify the email address of the trouble ticket system to which you want to send the message. The data entry definition lets operators specify the address.

   If you do not want operators to be able to change the address, specify the address in the Mail Address field and delete the fields in the data entry definition.

   Complete the other fields:

   Host Name

   (IBM's Communications server only) Specifies the host name of this system. This is usually the NJE node name.

   SMTP Node Name

   (IBM's Communications Server only) Specifies the NJE node name on which the SMTP server runs. This is usually the same value as the Host Name.

   SMTP Job Name

   (IBM's Communications server only) Specifies the name of the address space in which SMTP runs. This is usually SMTP.

   SMTP DEST Id

   (CA TCPaccess CS only) Specifies the destination ID in the REMOTE parameter of the SMTP statement in member APPCFG*xx* of the PARM data set.

   Exit Procedure Name

   Specifies the name of an NCL exit routine, in which you can customize the email message sent by this trouble ticket.

   Subject

   Specifies the heading to display as the subject of the email message.

   Enter Mail Text Below

   Specifies the mail message text. Press F1 (Help) for information about variables.

   Press F3 (File).

   The definition is saved.

Define a Custom Trouble Ticket Interface

You use the custom interface if you want to use your own procedure to send trouble tickets.

To define a custom trouble ticket interface

1. Enter **/ALADMIN** at the prompt.

   The Alert Monitor : Administration Menu appears.

2. Select option **I** - Define Trouble Ticket Interface.

   The Alert Monitor : Trouble Ticket Interface Definition panel appears.

3. Enter **CUSTOM** in the Interface Type field, and press F6 (Action).

   The Custom Trouble Ticket panel appears.

4. Complete the following fields:

   Procedure Name

   Specifies the name of your NCL procedure for delivering tickets.

   Important! The NCL procedure must be in the COMMANDS concatenation for your region. To list the concatenation, enter **/ALLOC**.

   Enter Parameters Below

   Specifies any parameters that you want the NCL procedure to receive. Press F1 (Help) for information about variables.

Example: Define a Custom Trouble Ticket Interface

This example shows an interface that uses the distributed CA SOLVE:Central exit, $RMPB06S, to send tickets to a CA SOLVE:Central region with the ACB name SOLVPROB and other required values.

```
PROD----------------- Alert Monitor : Custom Trouble Ticket ----Columns 001 074
Command ===>                                    Function=Update Scroll ===> CSR

Procedure Name  $RMPB06S

                             Enter Parameters Below

**** ****************************** TOP OF DATA ********************************
0001 ACBNAME=solvprob
         parm1=value1
         parm2=value2
**** ***************************** BOTTOM OF DATA ******************************
```

Example: Invoke a REXX procedure

This example shows how you can use the NCL procedure to execute a REXX procedure.

The NCL statement that executes a REXX procedure in your environment has the following format:

REXX *rexx_procedure parm_1 … parm_n*

Define a CA Service Desk Trouble Ticket Interface

The CA Service Desk integration (see page 184) feature must be implemented before you can send alert trouble tickets to it; otherwise, all alert forwarding requests fail.

Note: For more information, see the *CA Common Services for z/OS Service Desk Integration Guide*.

To define a CA Service Desk trouble ticket interface

1. Enter **/ALADMIN** at the prompt.

   The Alert Monitor : Administration Menu appears.

2. Select option **I** - Define Trouble Ticket Interface.

   The Alert Monitor : Trouble Ticket Interface Definition panel appears.

3. Enter **SERVICEDESK** in the Interface Type field, and press F6 (Action).

   The Service Desk Trouble Ticket Setup panel appears.

4. Complete the following fields:

   CA Service Desk Server Web Services HTTP URL

   Specifies the HTTP URL of the web services definitions on the target CA Service Desk server.

   Default: If left blank, the CA Common Services CAISDI/soap component chooses the default server.

   Note: This URL points to the web services definitions that CAISDI/soap invokes to create the requests. This is not the same as the URL that is used to log on to CA Service Desk. Contact your CA Service Desk administrator for the URL.

   CCI Sysid

   Specifies the CCI system ID of the LPAR where the CAISDI/soap task is active. This is the SYSID name specified in the CAICCI startup JCL.

   Default: If left blank, the local CAICCI on this LPAR locates a suitable CAISDI/soap task.

   Request Description Format

   Specifies whether the USD Request Description field is produced with HTML formatting or in plain text (TEXT).

   Default: HTML

   Note: In most cases, leaving the CA Service Desk Server Web Services HTTP URL and CCI Sysid fields blank will suffice. This lets the CAISDI/soap component use its default values.

   Press F3 (File)

   The definition is saved.

Set Up the Trouble Ticket Data Entry Definition

If you want the operator to supply information when creating a trouble ticket, you need to set up the ticket data entry definition.

To set up the trouble ticket data entry definition

1. Enter **/ALADMIN** at the prompt.

   The Alert Monitor : Administration Menu appears.

2. Select option **D** - Define Trouble Ticket Data Entry.

   The Trouble Ticket Data Entry Definition panel appears.

3. In the free-format data entry section of the panel, enter the data entry definition for the panel that the operator will use when creating a ticket.

   You can create multiple field names by replicating the key variables linked by default.

   Note: For more information about completing this section, press F1 (Help).

Example: Data Entry Definition to Prompt Operators for Email Address

The following example shows a definition that prompts the operator to identify the receiver of the ticket.

```
PROD--------- Alert Monitor : Trouble Ticket Data Entry Definition ------------
Command ===>                                    Function=Update Scroll ===> PAGE

**** ****************************** TOP OF DATA ******************************
0001 FIELD NAME=$USRNAME
0002  VALUE="Problem@sydney.enterprise.com"
0003  DESC="Send Email to:"
0004  COMMENT="(name for email)"
0005  REQUIRED=YES
0006  LENGTH=40
**** ***************************** BOTTOM OF DATA ****************************
```

Implement Trouble Ticket Interface for Multiple Email Addressees

You can use an exit procedure, together with the trouble ticket interface and data entry definitions, to implement an interface that prompts operators for more than one email address.

To implement a trouble ticket interface for multiple email addressees

1. Create an NCL procedure with the following statements, and save it to your TESTEXEC:

```
&IF .&$USRNAME1 NE . &THEN +
&$AMTADDRESS1 = &$USRNAME1
&IF .&$USRNAME2 NE . &THEN +
&$AMTADDRESS2 = &$USRNAME2
...
```

   Note: The number of &IF statements sets up the number of addresses you want to provide.

2. Update the trouble ticket data entry definition (see page 174) with the following fields:

```
FIELD NAME=$USRNAME1
VALUE="&$AMTADDRESS1"
DESC="EMAIL ADDRESS #1"
COMMENT=""
REQUIRED=NO
LENGTH=40
FIELD NAME=$USRNAME2
VALUE=""
DESC="EMAIL ADDRESS #2"
COMMENT=""
REQUIRED=NO
LENGTH=40
...
```

   Notes:

   - The number of fields corresponds to the number of email addresses in the procedure you created.
   - The value &$AMTADDRESS1 must be specified.

3. Define the email trouble ticket interface (see page 169) specifying a default address in the Mail Address field and the name of the procedure in the Exit Procedure Name field.

   The trouble ticket interface prompts operators for email addresses when they enter TT next to an alert.

Example: Implement a Trouble Ticket Interface for Two Email Addresses

To create an NCL procedure named EXAMPLE that sends emails to two addresses

1. Create an NCL procedure named EXAMPLE with the following statements, and save it to the TESTEXEC:

```
&IF .&$USRNAME1 NE . &THEN +
&$AMTADDRESS1 = &$USRNAME1
&IF .&$USRNAME2 NE . &THEN +
&$AMTADDRESS2 = &$USRNAME2
...
```

2. Enter **/ALADMIN** at the prompt.
3. Select option **D** - Define Trouble Ticket Data Entry.
4. Complete the panel as follows:

```
PROD-------- Alert Monitor : Trouble Ticket Data Entry Definition -------------
Command ===>                                    Function=Update Scroll ===> CSR

****** ***************************** TOP OF DATA ******************************
000001 FIELD NAME=$USRNAME1
000002  VALUE="&$AMTADDRESS1"
000003  DESC="EMAIL ADDRESS#1"
000004  COMMENT=""
000005  REQUIRED=NO
000006  LENGTH=40
000007 FIELD NAME=$USRNAME2
000008  VALUE=""
000009  DESC="EMAIL ADDRESS #2"
000010  COMMENT=""
000011  REQUIRED=NO
000012  LENGTH=40
****** **************************** BOTTOM OF DATA ****************************
```

5. Enter **/ALTTI** at the prompt.
6. Enter **EMAIL** in the Interface Type field and press F6 (Action).
7. Complete the panel as follows:

```
PROD--------------- Alert Monitor : Email A Trouble Ticket -Columns 00001 00072
Command ===>                                    Function=Update Scroll ===> CSR

Mail Address              defaultaddress@tt.com_____________________________
Host Name      (IBM)      HOSTNAME
SMTP Node Name (IBM)      NODENAME
SMTP Job Name  (IBM)      SMTP____
SMTP DEST Id (TCPaccess)  ________
Exit Procedure Name       EXAMPLE_
Subject                   &$AMDESC_____________________________________________

                             Enter Mail Text Below

****** ***************************** TOP OF DATA ******************************
```

Result

When an operator enters **TT** next to an alert, they are prompted for an email address as follows:

```
PROD--------------- Alert Monitor : Trouble Ticket Details --------------------
Command ===>

Email Address #1 ... defaultaddress@tt.com
Email Address #2 ...
```

Define Alert Monitor Filters

You can filter the alerts displayed on the Alert Monitor by applying a set of criteria to each of the fields in the alert. The filters that you create can be named and stored for later use, using the FILTER command.

To define an Alert Monitor filter

1. Enter **/ALFILT** at the prompt.

   The Alert Monitor : Filter Definition List panel appears.

2. Press F4 (Add).

   The Alert Filter panel appears.

3. Complete the following fields:

   Name

   Specifies the name of the filter.

   Description

   Describes the filter.

   Filter Expression

   Specifies the Boolean expression that determines what alerts are passed by the filter. For more information about creating Boolean expressions, press F1 (Help).

   Press F3 (File)

   The Alert Monitor filter is saved.

Alert Monitor Display Format

The Alert Monitor display format determines the information displayed for the alerts on the Alert Monitor, for example, the columns and the order in which they appear.

You specify the Alert Monitor display format on the List Format panel.

For each type of information you want to display on the Alert Monitor, you need to specify two items: a static heading and a variable that contains the required information.

You can create a multiscreen Alert Monitor display with up to 10 screens, enabling you to display more information on the monitor. The screens can be accessed by pressing the F11 (Right) or F10 (Left) function keys from the monitor.

The variable contains the information you want to display. The name of a variable can sometimes be longer than the data to display. You can enter a shorter name and then make that shorter name an alias of the actual name.

Create the Alert Monitor Display Format

You can create format definitions that can be used to customize the information displayed on the Alert Monitor.

To create the Alert Monitor display format

1. Enter **/ALADMIN.L** at the prompt.

   The List Definition List appears.

2. Enter **C** beside the DEFAULT display format definition.

   A copy of the List Description panel appears.

3. Enter a new value in the List Name field to identify the new definition, and update the Description and Title fields.

   Press F8 (Forward) three times.

   The List Format panel appears.

4. Enter column headings and variables using the text editor to specify the information to display on the Alert Monitor.

   Note: For more information about the text editor, press F1 (Help).

5. (Optional) Press F5 (Fields) to create aliases.

6. Press F3 (File).

   The details are saved.

Enable Alerts from External Applications

You can generate alerts (to view on the Alert Monitor) from external applications such as CA OPS/MVS EMA.

Note: To utilize this feature, the SOLVE SSI must be active.

To enable alerts from external applications

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups list appears.

2. Enter **U** beside the $NM ALERTS parameter group in the Interfaces category.

   The ALERTS - Alert Monitor Interface panel appears.

3. Enter **YES** in the Enable External Alerts? field.

4. Press F6 (Action).

   The changes are activated immediately.

5. Press F3 (File).

   The settings are saved.

Alert Forwarding

Alerts are displayed on the Alert Monitor; however, you can also forward them to the following platforms:

- EM Console in CA NSM
- UNIX platforms as SNMP traps
- CA NetMaster NM for SNA or Tivoli NetView (TME10) systems, as generic alert NMVTs
- CA Service Desk servers (see page 184), as CA Service Desk requests or incidents

You can apply filter criteria to forward different types of alerts to different platforms.

Alert forwarding does not require manual intervention; it occurs automatically when the alert is created.

Implement Alert Forwarding

You implement alert forwarding by using Customizer parameter groups.

Note: TNGTRAP and SERVICEDESK do not have clear alert events. Only alert open and considerations are forwarded.

To implement alert forwarding

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups list appears.

2. Enter **U** in front of the ALERTS parameter group in the Interfaces category.

   The parameter group opens for update.

3. Complete the following field:

   Dest Type

   Specifies the type of alert forwarding to use.

   Press Enter.

   The fields dynamically change to match the specified destination type.

4. Review the fields, and update as required.

   (Optional) Press F8 (Forward), and repeat Step 3 for each Definition ID.

   Note: Press F1 (Help) for information about the fields.

5. Press F6 (Action).

   The changes are applied.

6. Press F3 (File).

   The settings are saved.

SNMP Trap Definition

The MIB definition for alerts forwarded as SNMP traps is provided in member $AMTRAP, supplied in the CC2DSAMP data set. You can download this member to your UNIX system and compile it.

Note: When copying this member to your UNIX system, you can rename it to avoid problems on some UNIX systems where the $ sign has special meaning.

The supplied MIB defines two traps with the following object identifiers:

- $AMTRAP = 1.3.6.1.4.1.1126.1.2.1.2 (for an alert)
- $AMTRAPC = 1.3.6.1.4.1.1126.1.2.1.3 (when an alert is cleared)

Forward to Tivoli NetView

To receive alerts in a Tivoli NetView region, the CNMCALRT task must be defined and active. The alerts are formatted as Operator Notification generic alerts.

To forward alerts to Tivoli NetView

1. Check the DSIDMN (or DSIDMNB) member in the DSIPARM PDS. DSIPARM.PDS is allocated by the Tivoli NetView started task.
2. Ensure that the CNMCALRT task is included and is initialized (INIT=Y). For example:

   ```
   TASK MOD=CNMCALRT,TSKID=CNMCALRT,PRI=6,INIT=Y
   ```

   Note: This statement is necessary for the z/OS software alert forwarding function.

Forward to CA NSM

To format the traps sent to a CA NSM management platform, you must load the rules to reformat the alert messages for display on the EM Console.

To forward alerts to the EM Console in CA NSM

1. Use FTP to download the message definition rules in binary mode from the UNIEMMSG member of your CC2DSAMP data set created at installation. For example, using the Windows FTP client from the prompt:

   ```
   >ftp myhost
   Connected to myhost.mycompany.com.
   User (myhost.mycompany.com:(none)): user01
   331 Send password please.
   Password: xxxxxxxx
   230 USER01 is logged on. Working directory is "/u/users/user01".
   ftp>cd "prefix.ppvv.CC2DSAMP"
   250 The working directory "prefix.ppvv.CC2DSAMP" is a partitioned data set
   ftp>binary
   200 Representation type is Image
   ftp> get uniemmsg uniemmsg.txt
   200 Port request OK.
   125 Sending data set prefix.ppvv.CC2DSAMP(UNIEMMSG) FIXrecfm 80
   250 Transfer completed successfully.
   ftp: 3200 bytes received in 0.67Seconds 4.77Kbytes/sec.
   ftp>quit
   ```

2. From a Windows prompt on the destination CA NSM EM Server, load the message definition rules from the downloaded file. Enter the following command at the prompt to define the rules to event management:

   ```
   cautil -f "uniemmsg.txt"
   ```

3. Enter the following command to load the rules:

   ```
   oprcmd opreload
   ```

4. In your region, set the alert forwarding destination to TNGTRAP.

Alert Forwarding to CA Service Desk

Before you can forward alert details to CA Service Desk to create requests, you implement CA Service Desk Integration.

Note: For more information, see the *CA Common Services for z/OS Service Desk Integration Guide.*

Do not forward any alerts to CA Service Desk until integration is completely and correctly implemented; otherwise, all alert forwarding requests to CA Service Desk fail.

Suppress State Change Alerts

The region automatically generates an alert for a resource that changes state. You can suppress the alerts for selected state changes. You can also specify the severity levels of the generated state change alerts.

To suppress automatically generated state change alerts

1. Enter the **/PARMS** panel shortcut.

   The Parameter Groups panel appears.

2. Enter **F STATECHANGE**.

   The cursor locates the STATECHANGE parameter group.

3. Enter **U** beside the group.

   The group opens for updating.

4. Blank out the fields for the states you want to suppress alerting. For example, if you want to suppress alerting for state changes to UNKNOWN, blank out the Unknown field.

   Press F6 (Action).

   The region stops generating alerts for those state changes.

5. Press F3 (File).

   The group is updated with the changes.

State Change Alerts

State change alerts are based on RMAM001*xx* messages. These messages are defined in CAS, and you can customize them.

You can maintain messages from the Message Definition List panel. The shortcut to the panel is /CASMSG.

Note: For information about how to maintain messages, see the *Managed Object Development Services Guide*.

CA Service Desk Integration

The CA Service Desk Integration feature creates CA Service Desk requests from forwarded alerts and alert trouble tickets, or both.

You can define multiple forwarding destinations to CA Service Desk, with each one pointing to a different CA Service Desk server.

Note: If your CA Service Desk installation is configured with the optional ITIL application, incidents are created instead of requests.

Many CA Technologies mainframe products use this feature to consolidate their problem notification on a specified CA Service Desk server. The feature uses W3C SOAP (Simple Object Access Protocol) to invoke web services provided by CA Service Desk.

Software Requirements

CA Service Desk Integration has the following software requirements:

- CA Service Desk r11 or r11.1
- CA Common Services for z/OS r11, specifically the CAICCI and CAISDI/soap components

How Requests Are Created

To create a CA Service Desk request from an alert, the following internal steps are performed:

1. The CA Common Services for z/OS CAICCI component is used to pass the request to the CA Common Services for z/OSCAISDI soap component. CAISDI/soap is a z/OS-hosted SOAP client.
2. CAISDI/soap sets up an IP connection with the CA Service Desk server, then uses HTTP/HTTPS requests to invoke the necessary web services on the CA Service Desk server to create the new request or incident.
3. The request or incident number is returned and annotated in the alert.

Request Assignment

By default, CA Service Desk requests created by your region appear as *assigned* requests, with an assignee and an end user of System_NetMaster_User.

Your CA Service Desk administrator can customize the product templates to change these assignments to suit your organization.

Request Updating

A CA Service Desk request created from an alert is static. It reflects the alert details that were current at the time it was created.

Note: A CA Service Desk request is not subsequently updated with any changes to the alert, nor closed when the corresponding alert is closed.

Requests are intended for initial problem notification to a wider and more general data center audience. CA Service Desk Integration complements the functions of the Alert Monitor; it does not replace the Alert Monitor.

Every request (if HTML format is used) contains hyperlinks to various WebCenter pages, including the Alert Monitor. You should use the Alert Monitor for real-time dynamic alerting functions.

For recurring alerts, a request is created for the first occurrence only.

Other Ways to Create Requests or Incidents

In addition to Alert Monitor forwarding and trouble tickets, CA Service Desk requests or incidents can also be created from the following functions:

- Operator Console Services (OCS)
- MVS console

Operator Console Services

The OCS command SDCREATE can be used to create a CA Service Desk request from the OCS command line, for example:

```
SDCREATE Problem xxx has occurred
```

This attempts to open a request on the default CA Service Desk server. The request will have a severity of 4, and a summary and description of *Problem xxx has occurred*. Like other requests raised, it is assigned to System_NetMaster_User.

Use the SDTEST command to check if a default server is implemented.

MVS Console

As with any product command, you can also issue SDCREATE from the MVS system console, for example:

```
F rname,SDCREATE Problem xxx has occurred
```

Request Description Format

By default, your region generates CA Service Desk request description content in HTML format.

By default, CA Service Desk does not render embedded HTML directives in the request description field. To support this, you must customize your CA Service Desk server. This task involves customizing the detail_cr.htmpl form to add keeptags and keeplinks support.

Note: For more information, see the *Service Desk Modification Guide*.

Implement the Alert History Function

The Alert Monitor retains data in an alert history file. You can define the time period that alerts are retained.

To specify the time period that alerts are retained

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups list appears.

2. Enter **U** in front of the $NM ALERTHIST parameter group in the Files category.

   The ALERTHIST - Alert History File Specification panel appears.

3. Complete the following fields:

   Days to Retain Alerts

   Specifies the number of days that you want to retain alerts in the history file.

   Limits: 999 days

   Default: 7 days

   Time of Day for Alert Purge

   Specifies the time of day (in the format hh.mm) at which alerts older than the value in the Days to Retain Alerts field are deleted from the history file.

   Press F6 (Action).

   The changes are applied.

4. Press F3 (File).

   The settings are saved.

Reorganize Files and Monitor Space Usage

Over time, the alert history file can become fragmented. You can reorganize the file to improve its efficiency.

To reorganize the Alert History database for optimum space usage

1. Copy (REPRO) the alert history file to a backup file.
2. Delete and redefine the original file.

Use the same attributes that were used when the file was defined at region setup. See the generated S01LCALC member in your INSTALL.JCL data set; this member has the original VSAM definition JCL for the file.

Monitor the amount of disk space used by the data set to estimate the optimal file size and optimal frequency of reorganization.

Example: Back Up Alert History File

This example backs up an alert history file.

```
//BKALERTH EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//IN       DD   DSN=?prefix.ALERTH,DISP=SHR
//OUT      DD   SN=?prefix.ALERTH.BACKUP.SEQ,DISP=OLD
//SYSIN    DD   *
 REPRO INFILE(IN) OUTFILE(OUT)
/*
```

The sequential backup file has the following format:

```
DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760
```

Extract Alert Data for Reporting

You can extract alert data from the Alert History database in a character separated values (CSV) format for processing by external reporting and analysis tools. The default field separator character is comma (,). You can change it in the ALERTHIST parameter group.

To extract alert data for reporting and analysis

1. Allocate a sequential data set with the following attributes:
   - LRECL is greater than or equal to 300 bytes.
   - RECFM is VB.
2. Enter **/ALHIST**.

   The History Menu appears.
3. Type **EX** at the prompt, and specify the data set name that you have allocated in the Extract DSN field.

   (Optional) If you want to limit the extracted data, select an Alert Monitor filter (see page 177) through the Filter Name field.

   Press Enter.

   The data is extracted to the specified data set.
4. Transfer the data set to your personal computer (PC) in ASCII format, and save it with an appropriate extension. (For example, if you plan to use Microsoft Excel to process the data, use the .csv extension.)

   The extracted data is saved in a text file.
5. Open the text file by using your preferred PC application.

   The extracted data is presented in your preferred format for analysis.
6. Analyze your data by applying facilities such as graphs and charts, tables, and macros.

Chapter 16: Implementing EventView

This section contains the following topics:

EventView

EventView performs automation at the event level. It provides event level automation and control, and can handle timed events.

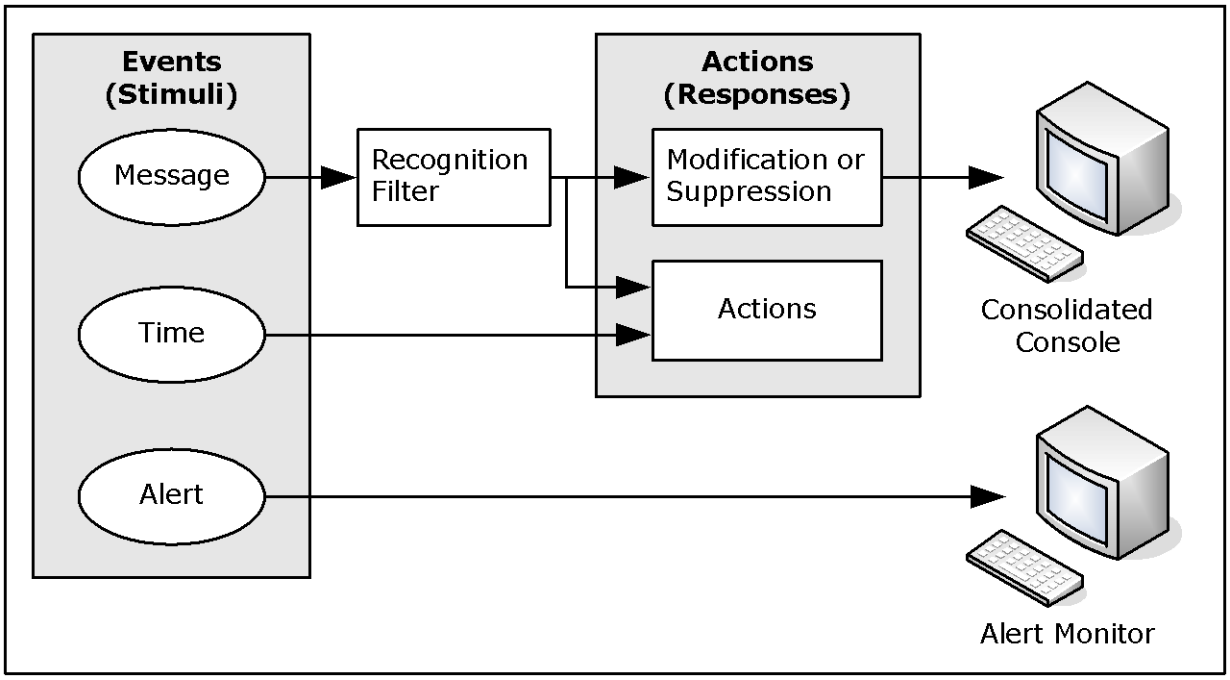
Successful event management relies on the recognition of significant events from the mass of messages generated by a system and the appropriate responses to these events.

EventView Functions

EventView provides the following functions:

- Event-based automation, which relies on the following:
  - The creation of appropriate rules and rule sets
  - The processing of messages
  - The processing of EventView timers
  - Message generation
- Console message consolidation
- Alert generation

The following illustration shows how EventView works.



Event-based Automation

You can define event rules to do the following:

- Suppress messages
- Change message text
- Enhance message presentation (for example, highlighting)
- Set route and descriptor codes
- Perform actions

Rules are grouped logically into rule sets, which define how an event is processed and what actions are taken in response to an event that is *not* related to a resource. (Resource-based events are handled by ResourceView.) An event can be a message or a specified time.

Sample Message Suppression Rule Sets

EventView provides the following samples of message suppression rule sets:

- AGRSUPP, which is based on the aggressive list of suppressible messages recommended by IBM
- CONSUPP, which is based on the conservative list of suppressible messages recommended by IBM

Note: For the IBM recommended lists, see IBM's *MVS Initialization and Tuning Reference* guide.

Console Message Consolidation

You can monitor message flows from multiple systems on a single screen—the consolidated console. Console consolidation controls the way you see messages on the console. In addition, messages displayed on the consolidated console are affected by EventView processing.

For example, message text and message presentation can be modified by EventView, and the consolidated console user sees the modified message. If EventView suppresses a message, that message is not displayed on the consolidated console.

You can define message profiles that customize the view of the message flow. Different users can have different sets of message profiles to suit the functions they perform.

Message profiles enable the meaningful grouping of messages based on criteria such as system, message ID, job name, and system codes.

Benefits of Using EventView

EventView benefits your organization in the following ways:

- Reduces system console message rates; you can filter messages received and suppress unwanted messages
- Produces a standardized response to events or problems
- Enables you to schedule actions to occur at specific times or at regular intervals
- Gathers useful statistics for messages and timers
- Able to learn messages
- Enables you to monitor message flows to multiple consoles on a single screen
- Enables you to generate alerts to remind operators of significant events

How You Implement Event Management Rules

Rule sets consist of a number of rules. The rules define how an event is processed and what actions are taken in response to the event.

Use a number of rule sets to organize your rules in logical and manageable groups.

For a rule set to be in an active state, it must be associated with an active system image. However, only one rule set can be associated with an image. If you want to activate more than one rule set, include the other rule sets in the rule set associated with the active image.

Implementing a rule set consists of the following typical stages:

1. Identify the rule sets you want to create.
2. Create the primary rule set to associate with the active image.
3. Add message rules, message group rules, and timers.
4. Add actions that are performed before normal rule processing occurs.
5. Create the other rule sets, and include them in the primary rule set.
6. Associate the rule set with the appropriate image.

Message Monitoring

Besides responding to resource status, you need to respond to events not handled by resource automation.

The Automation Services components that affect message display are the EventView message rules and the console message consolidation facility. To use the latter facility, you define message profiles.

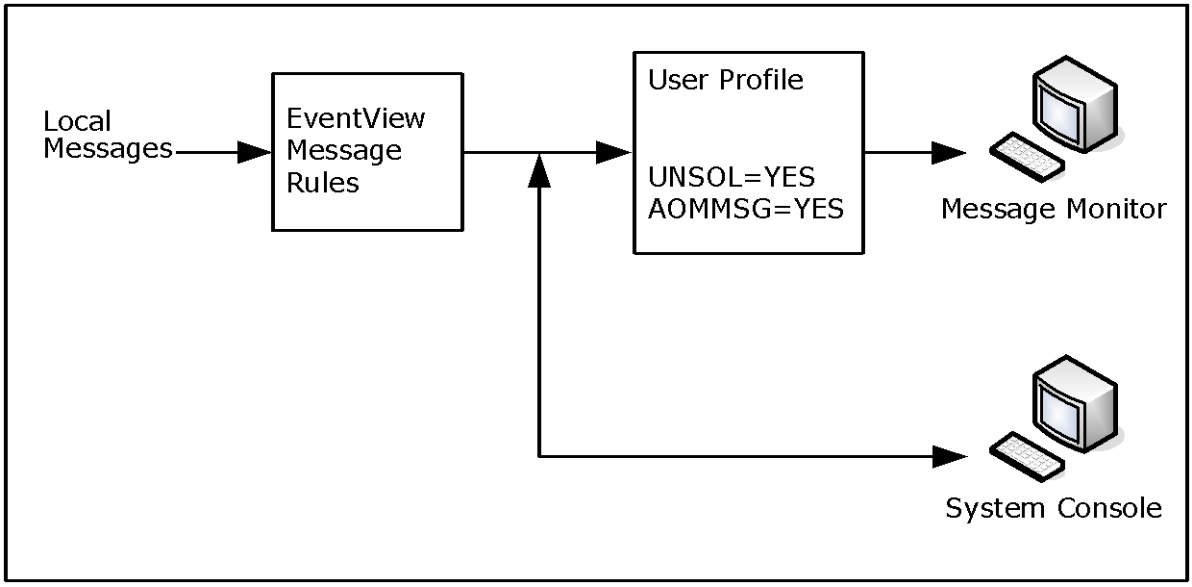
Typically, you use both components and you can monitor messages from multiple systems. However, if you do not want to define message profiles, you can disable the message consolidation facility. In this case, only messages from the local system can be monitored. You control the availability of the facility by using the CCONSOLIDATN parameter group.

Note: For more information about parameter groups, see the *Reference Guide*.

Console Consolidation Disabled

Without the console consolidation facility, you are able to monitor local messages only. Remote messages are not routed to this region, and messages from this system are not routed to remote regions.

The following illustration shows how messages arrive at the message monitor.

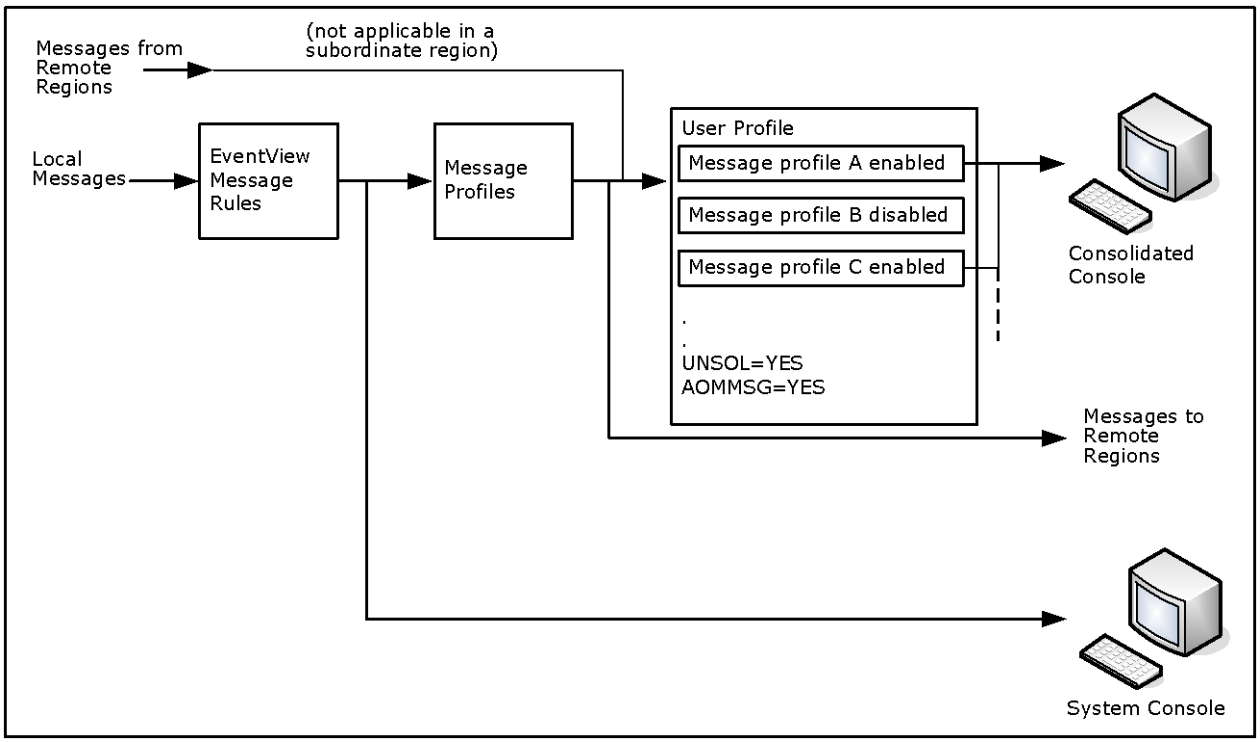


Console Consolidation Enabled

With message consolidation enabled, the message monitor becomes a consolidated console. Using the console consolidation facility, an operator is able to monitor messages from one or more systems on the consolidated console.

You use EventView message rules to preprocess the messages, for example, suppressing or highlighting the messages. You can then use message profiles to select the type of processed information to display on the consolidated console. For example, you can define a message profile that selects messages from a particular system. You can selectively enable profiles to customize the view of monitored events, for example, VTAM messages only.

The following illustration shows how messages arrive at the consolidated console.



How You Implement Message Profiles

Typical stages in implementing message profiles are as follows:

1. Analyze the message flow and the operations tasks to determine the different message views that are required.
2. Create EventView rules to suppress unwanted messages.
3. Create the message profiles, and assign each operator the appropriate message profile IDs in the user definition and user profile.
4. Activate the message profiles.

Alert Generation

Alerts are displayed on the alert monitor. Alerts can be generated from EventView rules. Generate alerts through user-defined processes by using the following macros:

- GENALERT enables a process to generate an alert of a specified severity.
- DELALERT enables a process to remove an alert from the alert monitor.

Use alerts to warn operators of significant events (for example, reminding the operator to perform tasks that cannot be automated).

Chapter 17: Implementing EventView Rule Sets

This section contains the following topics:

EventView Rule Sets

EventView rule sets consist of various members that define how an event is processed and what actions are taken in response to the event. An EventView rule set can include the following:

- Initial actions
- Message rules
- Message group rules
- Timer rules
- Other rule sets

You can create an EventView rule set for each area of responsibility. For example, you can create a CICS rule set, a VTAM rule set, and so on, to organize your rules in logical and manageable groups.

Note: Specify automation that deals with the status of resources in the resource definition, *not* in a message rule.

To activate an EventView rule set, it must be associated with an active system image.

Note: For information about system images, see the *Reference Guide.*

Only one EventView rule set, known as the primary rule set, is associated with a system image. If you want to activate more than one EventView rule set, include the other rule sets in the rule set associated with the active image. For example, you could create a master EventView rule set into which all other EventView rule sets are included.

Add an EventView Rule Set

You must add an EventView rule set before you can add the associated members.

To add an EventView rule set

1. Enter **/EADMIN.R.R** at the prompt.

   The Ruleset List panel appears.

2. Press F4 (Add).

   The Ruleset Description panel appears.

3. Complete the panel, adding comments on the Comments panel if required. See the online help for field descriptions.

   Note: An EventView rule set can be activated only if it has an ACTIVE status.

Specify Control Options for Testing

The control options for the primary EventView rule set override those options specified for the included EventView rule sets.

When setting up an EventView rule set, you can test it without actually triggering any rules.

To set up a rule set for testing, specify the following values:

- **NO** for the Perform Message Modification? and Perform Action? options
- **YES** for the Log Ruleset Activity? option

You can then see from the entries in the general activity log (marked as TEST) what activity would take place if the EventView rule set was, in reality, working as intended.

Example: Messages Logged in Test Mode

This example shows some messages logged in test mode.

```
09.04.49 RE0113 RULESET ACTIVITY LOGGING STARTED
09.04.58 RE0130 (TEST) RULE FOR TESTMSG SET ATTRIBUTES: DELIVER=NO
09.05.12 RE0114 RULESET ACTIVITY LOGGING STOPPED
```

Monitor EventView Rule Set Status

To view the status of the active EventView rule set and all its included rule sets on the current system, enter **/EADMIN.S.R** at the prompt.

The EventView : Ruleset Status panel appears. This panel displays the same information as the Ruleset Description panel, plus it lists loaded EventView rule sets. The primary EventView rule set is the first EventView rule set listed, followed by its included EventView rule sets. Each level of further inclusion is indicated by indentation.

Note: If an EventView rule set has a status of inactive, its included EventView rule sets are not processed.

Statistics

If you specify YES in the Collect Statistics? field on the Ruleset Description panel, then EventView collects statistics relating to messages received and timer schedule items executed. You can use these statistics to measure the effectiveness of your EventView rules.

If the SMFDATA region parameters are configured, the statistics are output to SMF at a user-defined interval.

Note: For information about the SMF record format, see the *Reference Guide*.

Change the EventView Rule Set Associated with a Local System Image

You can change the EventView rule set associated with a local system image. Update the EventView Ruleset to Activate field on the System Image Definition panel.

Note: The associated rule set is activated when the system image is loaded initially. If you change the rule set associated with the system image later, you do not have to reload the system image. The rule set takes effect immediately when the system image definition is saved.

To change the EventView rule set associated with a local system image

1. Enter **/RADMIN.I.L** at the prompt.

   The Local System Image List appears.

2. Enter **U** (Update) next to the system image that you want to update.

   The Local System Image Definition panel appears.

3. Enter the new EventView rule set name in the EventView Ruleset to Activate field.

   Note: You can select an EventView rule set from the prompted field value list.

4. Press F3 (File).

   The updated record is saved.

   Note: By default, EventView rule actions are not executed if the system image is operating in the MANUAL global operation mode. The actions, however, can be enabled by using the Perform Action in Manual Mode? field of the AUTOIDS region parameter group.

Add Associated Rules

After you have created an EventView rule set, you can add associated message, message group, or timer rules.

To add a rule

1. From the Ruleset List, apply the appropriate action, such as **M** (Message List), to the EventView rule set with which you want to associate the new rule.
2. Press F4 (Add).
3. Complete the fields on the initial panel displayed, and on any subsequent panels as required. Press F1 (Help) for help about the fields.

Message Rules

Message rules contain some or all of the following information:

- Message text and filtering criteria
- Message delivery and suppression details
- Required message modification details
- Which actions a message triggers
- Which message groups the current message rule is related to
- User-defined EventView variables

How You Add Message Rules

Message rules are added (see page 201) in the same way as other EventView rule set members. You apply the **M** (Message List) action to the EventView rule set with which you want to associate the new message rule.

Important! Message Text is a mandatory field. If you enter the wildcard character in this field, *all* messages are tested against this rule, which can degrade performance.

Message Execution Conditions

You can specify execution conditions (see page 227). The rule executes only if all of the given conditions apply.

Message Groups

If a message on its own is not significant, but another message increases its significance, create a message group to associate these messages.

Note: The order in which the grouped messages occur is not important, as long as all arrive in the specified time interval.

Message group rules contain the following information:

- The maximum time interval in which all messages in the group must be received, to trigger the rule
- Message text for up to ten messages, on the Message Group Details panel (displayed automatically when a message is associated with a message group (see page 202))
- The text of a message to issue if the group rule is triggered, and where and how to display this message
- The action or actions to perform when a group rule is triggered
- User-defined EventView variables, which you can set to the specified values before or after other rule actions

How You Add Message Group Rules

Message group rules are added in the same way as other EventView rule set members. You apply the **G** (Group) action to the EventView rule set with which you want to associate the new message group.

Associate Message Rules with Message Group Rules

To establish a relationship between a message rule and a message group rule, you must add an entry on the Related Message Group panel (the fifth panel in the sequence of Message Rule panel). The message rule is in turn added on the Message Group Details panel of the message group rule definition. The same message rule can be associated with up to five message group rules.

To associate a message rule with a message group rule

1. Enter **5** from within the message rule definition, for example:

```
PROD------------------ EventView : Message Filter ----------------------TAPEMON
Command ===> 5                                                 Function=UPDATE

Ruleset Name ........ TAPEMON                        Rule Status ...+ ACTIVE__
Short Description ... Mount request processing_____

. Expected Message ------------------------------------------------------------.
|                                        S=ListPanels EV=ExtFilter TV=TestVars |
|      Message Text  ( WildChar = * )                                   ExtFlt |
| ___  IEC501A                                                          NO     |
'-------------------------------------------------------------------------------'
```

   The Related Message Group panel appears.

2. Add the message group rule. Optionally, specify a correlation key for precise recognition purposes.

   The following panel shows an example:

```
PROD---------------- EventView : IEC501A Related Message Group ---------TAPEMON
Command ===>                                                   Function=UPDATE

. Message Group Table ---------------------------------------------------------.
|                                                                              |
| MsgGroupID   CorrelationKey                                                  |
| GROUP1_____ &ZMSGJOBNM_____________________________________________________ |
```

3. Press F3 (File).

   The message rule is updated and added to the message group rule definition, for example:

```
PROD---------------- EventView : Message Group Details -----------------TAPEMON
Command ===>                                                   Function=UPDATE

Ruleset Name ......... TAPEMON
Message Group Name ... GROUP1                        Rule Status ...+ ACTIVE__
Short Description .... Tape mount group_____________
Interval ............. 00.10.00

. Expected Message ------------------------------------------------------------.
|                                                      S/B=Browse U=Update     |
|      Message Rule Text                                                       |
| ___ IEC501A                                                                  |
| ___ IEC509A                                                                  |
| ___                                                                          |
```

The correlation key enables one message group rule to cover numerous different situations, saving you from having to create numerous different rules. The rule is not triggered unless the values of the correlation keys in each of the grouped messages match.

For example, the correlation key as shown in the example, is the name of the variable that contains the job name. The group rule is triggered only if the messages associated with the group:

- All arrived in the specified interval (10 minutes)
- Were all generated by the same job

Timers

If you want a rule triggered on a particular day of the week (or year) and at a particular time, you need to add a timer rule. Timer rules contain the following information:

- Whether the timer rule applies to a specific system
- Up to 99 detailed schedule items
- Which actions are triggered by a timer
- User-defined EventView variables

How You Add Timers

Timer rules are added (see page 201) in the same way as other EventView rule set members. You apply the **T** (Timer) action to the EventView rule set with which you want to associate the new timer.

Initial Actions

Initial actions are actions performed when an EventView rule set is activated (that is, when the associated system image becomes active), and before message processing commences.

How You Add Initial Actions

You add initial actions from the Ruleset List by applying the **IA** (Initial Actions) action to the nominated EventView rule set.

Set variables that are essential to the functioning of an EventView rule set in the initial action rules.

Note: If an EventView rule set has associated included EventView rule sets, the initial actions specified for those EventView rule sets are also performed when the primary EventView rule set becomes active.

If you want to set any EventView variables before or after any of the initial actions are performed (to pass parameter values, for example), press F8 (Forward) to go to the Set Variables panel.

On the Set Variables panel, you supply a name for each EventView variable that you want to set, plus the required variable value. When you use the variable subsequently, you prefix the name with &ZREV, which is the EventView variable identifier.

Example: Log Rule Set Activation Message

This example logs a message to indicate that an EventView rule set is activated.

```
PROD------------------- EventView : Initial Action ----------------------BACKUP
Command ===> forward                                            Function=UPDATE

Ruleset Name ........ BACKUP
Initial Action Name   NOTIFY                          Rule Status ...+ ACTIVE
Short Description ... Log a startup message

System Command ... ____________________________________________________________
                   ____________________________________________________________
MS Command .......... LOG RULESET BACKUP IS NOW ACTIVE
```

How Initial Actions Are Executed

When an EventView rule set is activated, the associated initial actions are executed. When you load a system image that contains an EventView rule set that is already active, the region does not reactivate that EventView rule set and the associated initial actions are not executed (for example, when you switch images that use the same EventView rule set).

If you have several system images that use the same EventView rule set and you want the initial actions associated with the EventView rule set executed every time you load one of those images, you can create a primary EventView rule set for each of the images. Each primary EventView rule set includes the actual EventView rule set you want. Because the primary EventView rule sets are different, it is activated every time you switch between the images, thus executing the initial actions.

Include EventView Rule Sets in Other Rule Sets

To include an EventView rule set in another rule set

1. From the Ruleset List, apply the **IR** (Include) action to the EventView rule set in which you want to include another EventView rule set.

   The Include Ruleset List appears. This list is blank if there is no EventView rule sets included in the current EventView rule set.

2. Press F4 (Add).

   The Eligible Ruleset List appears.

3. Select the EventView rule set to include in the current EventView rule set.

   The selected EventView rule set is added to the Included Ruleset List for the current EventView rule set. This means that the included EventView rule set is active when the parent EventView rule set is active.

Note: Only the control options of the EventView rule set associated to the system image are used. The control options of included rule sets are ignored.

Change the Status of Rule Sets, Rules, or Initial Actions

The status of a rule set, rule, or initial action can be ACTIVE or INACTIVE. It is set by the Status field in the definition. You can change the status of multiple rule sets, rules, or initial actions by applying the A (Activate) and I (Inactivate) actions.

To change the status of rule sets, rules, or initial actions

1. List the required type of definitions.
2. Type **A** or **I** beside the definitions for which you want to change status, and press Enter.

   The selected definitions are updated, and their new status is shown in the Status column.

Maintenance of EventView Rule Sets

You can browse, update, copy, and delete EventView rule set definitions from the Ruleset List panel.

The C and the D action codes enable you to copy and to delete an *entire* EventView rule set. To copy or delete the EventView rule set definition only, use the CO or DO action codes. You can use the DO action code to delete an EventView rule set only if it is empty—that is, it contains no rules.

EventView Variables

The ability to set and use EventView variables in rules lets you create dynamic rules that depend on conditions identified by other rules and EventView rule sets. That is, you use EventView variables to control rule execution.

EventView variables can be used for the following:

- To pass information and data between rules
- To obtain more information about the environment in which the rule is executing
- To record system states

EventView variables can be set on the Set Variables panel of a message rule, a timer rule, or a group rule. Here, you can set values for up to six variables. These values can be literal or you can specify a substitution variable as the source of the variable value for a message rule.

EventView variables can be used by:

- *Rules*, to do the following:
  - Provide a correlation key value to match on the Message Delivery panel and the Related Message Group panel.
  - Provide a value for insertion in replacement text. Replacement text specified on the Message Modification, Set Variables and Test Variables panels can include EventView variable names.
- *Processes*, where the macros EVVARGET and EVVARSET can be used to get and set the values of EventView variables. Variable names can be specified in the Parameters field on the Rule Action panel, as well as on other panels where processes are invoked (see page 111).
- *NCL procedures*, where the $RECALL application program interface (API) can be used to get and set the values of EventView variables. For more information about $RECALL API, see the *Reference Guide*.

You must remember to add the EventView variable indicator prefix, &ZREV, to a variable name when it is specified for evaluation.

View EventView Variables

To view all EventView variables that have been set, enter **/EADMIN.S.V** at the prompt.

The EventView : Active Variables panel appears.

Chapter 18: Configuring Timers

This section contains the following topics:

Timer Rules

A timer rule enables you to schedule an action or actions to perform at a specific time or times of the day, week, month, or year.

A timer rule contains the following information:

- The action or actions to perform
- A schedule that defines when the action or actions are performed
- Whether catchup is required, if the system running the timer is unavailable when the timer is due to be activated

A timer schedule is similar to the availability map used by resources controlled by the region. You can specify up to 99 schedule items per timer rule, each containing the following information:

- The day of the week, date, and time when the action or actions are performed
- Whether the action or actions are performed once only, or at regular intervals during a given time period

Add Timers

If you want to add a timer rule that is very similar to an existing one, you can save yourself having to retype details by copying the existing timer and updating the copy as appropriate.

To add a timer rule

1. Enter **/EADMIN.R**.

   The Define Event Rules panel appears.

2. Type **T** at the prompt and complete the following field:

   Ruleset

   Specifies the name of the rule set with which you want to associate the timer rule.

   Press Enter.

   The Timer Rule List for the specified rule set appears.

3. Press F4 (Add).

   The Timer Description panel appears.

4. Complete the fields on this and subsequent timer rule panels, as required.

   Note: For more information, press F1 (Help).

   Note: If you enter YES in the Delete on Expiry? field, schedule items that have a full date specified are deleted when they pass their expiry date and time.

Note: You can also add or update timers from the Active Timer Display List (**/EADMIN.S.T**).

How Catchup Works

When you define a timer, you specify whether catchup is required if a region running the timer is unavailable when the timer is due to be activated.

Note that if you enter YES in the Catchup Required? field on the Timer Schedule panel, catchup applies to all schedule items entered for this timer.

- If you specify YES, then the scheduled action or actions are performed when the region becomes available, provided that the time specified in the Window field has not elapsed, with the exception of the situation noted below.
- If you specify NO, no belated processing occurs for that timer.

Note: In the case of timers that define actions that are repeated, catchup can be requested. If the specified end time has passed by the time the region running the timer becomes available, the specified action or actions are still performed once. If the region running the timer becomes available part way through the specified time period, the specified action or actions continue at the specified intervals until the specified end time.

Catchup Window

If you specify that catchup is required, you can identify the window in which catchup is performed. You can specify a value between one minute and 24 hours. If the region running the timer becomes available before the catchup window ends, catchup is performed.

Timer Schedule Items

You can enter up to 99 schedule items for a timer. Enter schedule details according to the following definitions:

Day

Specifies the days of the week when the timer is activated. As well as the abbreviated versions, you can enter shorthand values asterisk (*), W/D, or W/E in this field. If an asterisk is entered, an individual schedule item is created for each of the seven days of the week, with all other values duplicated. If you enter W/D, an individual schedule item is created for each of the five working days of the week. Entering W/E results in the creation of individual schedule items for Saturday and Sunday.

Note: The validation procedure does not accept a value in both the Day and the Date fields; enter a value in one of these fields only.

Date

Specifies the date when the timer is activated. If you specify a numeric value between 1 and 31 in this field, the timer is activated on that day of the month each month. For example, if you specify 1, it is activated on the first day of each month. If, in addition to specifying a day, you also specify the first three characters of a month in the format *dd-mmm*, the timer is activated on that day of that month each year. If, in addition to specifying a day and a month, you also specify a four-character year value in the format *dd-mmm-yyyy*, the timer is activated on that day of that month and that year. If you entered YES in the Delete on Expiry? field, schedule items that have a full date specified are purged after execution.

Time

Specifies the time when the action or actions associated with the timer are performed or, if the Every field also contains a value, the time when the action or actions associated with the timer are first performed.

Every

Specifies the period if you want the action or actions associated with the timer performed at regular intervals. If you enter a value in this field, you must also enter a value in either the Num or the End Time field. When you complete one of these fields, the other is calculated automatically when validation occurs.

The first time the action or actions associated with the timer are performed is specified in the Time field—see the preceding field description. To calculate the time when the second occurrence of the action or actions associated with the timer are performed, the value in the Every field is added to the value in the Time field, and so on.

Num

Specifies the number of times that the action or actions associated with the timer are performed. When you enter a value in this field, the value in the End Time field is automatically calculated.

End Time

Specifies the last permissible time when the regular action or actions associated with the timer are performed. When you enter a value in this field, the value in the Num field is automatically calculated.

Status

Specifies the status of a timer schedule item: ACTIVE or INACTIVE. You can disable an individual timer schedule item by changing the status of that item from ACTIVE to INACTIVE.

Add Further Schedule Items

When you have completed the first seven entry lines on the static list displayed initially, you can add further schedule items.

To add further schedule items

1. Press F10 (Scrllst).

   Note: If you are using a 24-line screen, you can type MAX at the prompt to maximize screen use and to display the schedule list only.

2. To add a line to the schedule, apply the **R** (Repeat) action to a listed item.

3. Overtype the repeated line with the new schedule item details.

View the Next Execution of Timer Schedule Items

To view the next execution of the time schedule items, press the F5 (NextTmr) function key on the Timer Schedule panel.

The Next Execution Display panel appears.

This panel displays the next scheduled execution time and date of each timer schedule item in the order that they fall due, as well as all the schedule item details specified in the schedule map.

```
PROD--------------- EventView : Next Execution Display -----------------------
Command ===>                                                  Scroll ===> PAGE

    Item NextDate     NextTime  Day  Date        Time      Every  Num  EndTime
      2  08-AUG-1995  16.00.00  TUE              16.00.00  00.10  12   18.00.00
      3  09-AUG-1995  16.00.00  WED              16.00.00  00.10  12   18.00.00
      4  10-AUG-1995  16.00.00  THU              16.00.00  00.10  12   18.00.00
      5  11-AUG-1995  16.00.00  FRI              16.00.00  00.10  12   18.00.00
      6  12-AUG-1995  16.00.00  SAT              16.00.00  00.30  4    18.00.00
      7  13-AUG-1995  16.00.00  SUN              16.00.00  00.30  4    18.00.00
      1  14-AUG-1995  16.00.00  MON              16.00.00  00.10  12   18.00.00
    **END**

  F1=Help      F2=Split     F3=Exit                  F5=Find      F6=Refresh
  F7=Backward  F8=Forward   F9=Swap
```

Delete Timer Schedule Items

To delete a timer schedule item, apply the **D** (Delete) action to the item.

Timer Actions

On the Timer Actions panel, you can specify what response is made when a scheduled timer item is triggered. You can specify the following:

- System command text, such as: START STC1
- Command text, such as:

  ```
  LOG TEST TIMER RULE EXECUTED
  ```

- A process selected from the list of valid processes—enter a question mark in the field to display a list of valid processes
- An Automation Services command selected from the list of valid commands—enter a question mark in the field to display a list of valid commands

Example: Send Warning Message

The TSO resource is defined to stop automatically at 1900 on weekdays. To warn users of the impending shutdown, you can define a timer that sends a warning message to the users at 1845 on the weekdays.

```
PROD------------------ EventView : GRTIMER1 Rule Actions --------------FOGRULE1
Command ===>                                                    Function=COPY

System Command ... SEND 'TSO WILL BE STOPPED IN 15 MINUTES - PLEASE LOG OFF'
                   ____________________________________________________________
OCS Command ...... ____________________________________________________________
                   ____________________________________________________________
```

Display Active Timer Rules

To display active timer rules

1. Enter **/EADMIN**.

   The Event Administration Menu appears.

2. Enter **S.T**.

   The Active Timer Display appears.

   The displayed list shows the date and time of the next scheduled execution of all timer rules that have a status of active and are associated with the active rule set. If you scroll to the right, the schedule item details, as specified on the schedule map, appear.

   You can browse, update, copy, or delete listed timers.

Chapter 19: Processing Messages

This section contains the following topics:

CICS Messages

A region receives system messages by using its subsystem interface. With CICS support, a region also receives CICS messages by using a PPI connection. The region then processes these messages for use by the ResourceView and EventView features.

An unsolicited CICS message received by using a PPI has the following characteristics:

- It is associated with the job name of the CICS region in which the message is generated.
- It has a routing code of 13 (a solicited CICS message has a routing code of 14).

Message Rules

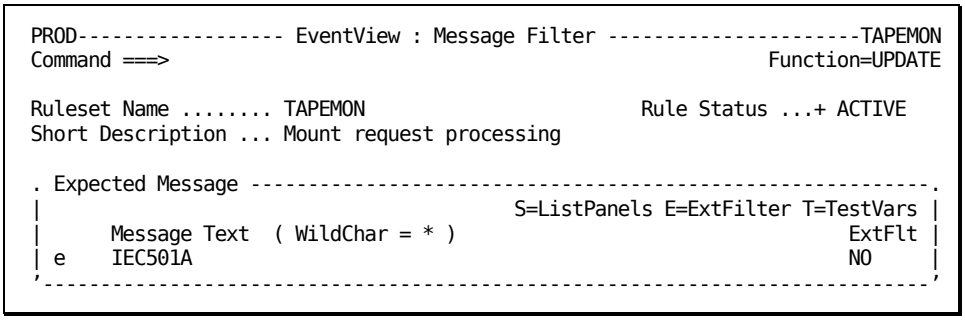
You use EventView message rules to process messages.

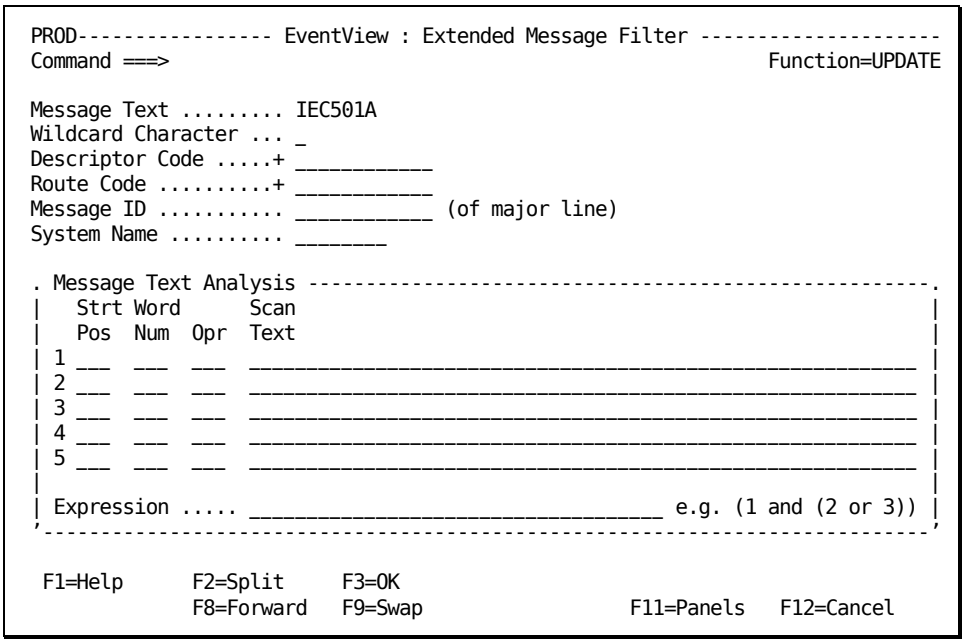
How You Specify Message Filtering Criteria

The text of a received message is compared with the scan text specified on the Message Filter panel. For example, if you specify TESTMSG1 as the scan text, any message starting with those eight characters is considered a match, including TESTMSG12 and TESTMSG1 TESTING.

Note: If you want to capture a message that has leading blanks, do not specify the leading blanks on the message filter panel. However, on the Extended Message Filter panel, absolute position is important so leading blanks must be counted when using start position of text.

This message text can include wildcard characters. The default is the asterisk (*). You can specify the message text that triggers the rule if the execution conditions are met. You can also specify additional filters on further panels to check for various different conditions. To access those panels, enter **E** next to the message text (as shown in the following illustration).





Use Wildcards in Message Text

Typically, you can simply specify enough message text to identify the messages you want the message rule to process.

You can also use wildcard characters to insert character patterns in the message text. If you use a wildcard character, you must also add a wildcard character to the end of the message text if necessary.

The following examples show the correct use of wildcard characters:

EC501A
IEC50*A*
IEC5**A*

The number of characters represented by a wildcard character is dependent on its position in the message text as follows:

- If the wildcard character is at the beginning of, or embedded in the message text, it represents one character.
- If the wildcard character is at the end of the message text, it represents any number of characters.

Extended Filtering Criteria

The Extended Message Filter panel lets you specify precise criteria to match:

Wildcard Character

Specifies a value other than the default value of an asterisk (*). This change is reflected in the Wildcard Character field on the Message Filter panel when you save the extended filtering criteria. This feature is useful if the message actually contains an asterisk.

Descriptor Code

Specifies one or more descriptor codes. A descriptor code determines the color that the operating system uses to display the message on a color console. The code also determines whether the message is a non-roll delete message. The descriptor codes assigned to a message are tested against the specified descriptor codes. A message matches if it contains any of the specified descriptor codes.

Route Code

Specifies one or more route codes. The operating system uses the route code to control message delivery. The route codes assigned to a message are tested against the specified route codes. A message matches if it contains any of the specified route codes.

Message ID

Specifies the first word of the message text (disregarding any flag characters, such as an asterisk, in position 1 or 2). When a secondary line of a multiline WTO message is filtered, the message ID for the line is the same as the ID for the primary line.

System Name

Specifies the name of the system from which the message originated. This field is useful if the local system reissues messages received from other systems. Messages are reissued if the system is part of a sysplex environment.

Message Text Analysis

You can analyze the text of the current message by word, phrase, or string, by specifying any combination of start position, word number, and permitted operator (such as equals, is greater than, and so on). You can specify up to five tests to perform on the message text and link these tests in an expression.

Note: EventView comparisons are *text-based*, that is, they are performed character by character, starting from the leftmost character of the extracted text. Text checking is done using the EBCDIC codes. Numbers are regarded as text. For example, the character string 100 is less than 99.

Message Text Analysis Criteria

You specify the message text analysis criteria on the following panels:

- Define Extended Filter Definitions panel (for a resource definition)
- Extended Message Filter panel (for a message action rule).

The panels enable you to analyze the text of the received message by specifying values in the Strt Pos, Word Num, and Opr fields. You can specify up to five tests, which are then linked in a defined, logical relationship that you specify in the Expression field.

For example, the Expression field has the entry 1 AND (2 OR 3). For the rule to be valid, Test 1 must be true and either Test 2 *or* Test 3 must be true.

A message consists of words. A word is a string of characters delimited by either a space or a comma. You have the option of specifying a word or part of a word for testing, or of extracting a substring for testing.

Important! ResourceView handles numeric comparisons; EventView always performs character comparisons.

Strt Pos

Specifies a position in the message where the text comparison is to start. The presence or absence of a value in the Word Num field determines the actual starting position.

If the start position is 2 and the Word Num field is blank, the comparison is on the partial message starting at the second character.

If the Strt Pos field is blank but the Word Num field has a value, then only that word is compared to the scan text. If the Strt Pos *and* the Word Num field are blank, the entire message is compared to the scan text.

If both Strt Pos and Word Num fields have values, the comparison narrows to a start position in a single word of the message text. The text used for comparison is the partial word. For example, if the word number is 8 and the start position is 2, the comparison starts from the second character of the eighth word.

For example, the following message arrives:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

- If the Strt Pos field has a value of 2, the string tested is as follows:

  ```
  AA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
  ```

- If the Word Num field has a value of 5, the string tested is as follows:

  ```
  MESSAGE
  ```

- If the Strt Pos field has a value of 2 and the Word Num field has a value of 5, the string tested is as follows:

  ```
  ESSAGE
  ```

Default: 1

Values: 1 through 999

Lne Num

Specifies a particular line in a multiline WTO or WTOR. If blank, any value in the Word Num field is treated as if all lines in the multiline message are joined as one string.

Values: Blank and 1 through 999

Note: Lne Num is not supported in EventView message rules; it is supported in ResourceView only.

Word Num

Specifies a particular word in a specific position in the message text string. If this field is blank, the entire message text that occurs after the specified start position is compared to the scan text. If this field contains a value but the Strt Pos field is blank, only the specified word is compared to the scan text. Spaces or commas delimits words.

Values: Blank and 1 through 999

Opr

Specifies a valid operator to control the type of comparison to perform if you enter a value in the Strt Pos or the Word Num field. The following operators are valid:

- CT (ConTain)
- EQ (EQual to)
- GE (Greater than or Equal to)
- GT (Greater Than)
- LE (Less than or Equal to)
- LT (Less Than)
- NE (Not Equal to).

If you enter a question mark (?) in this field, the list of valid operators is displayed.

Scan Text

Specifies the actual text (scan text) you want to test against the message text. You must have a match in the specified position or word for the comparison to be true. If you specify either CT or EQ as the operator, you can use the wildcard in or at the end of the Scan Text field. (You cannot use the wildcard character with the other operators.)

CT Operator

The CT operator tests whether the extracted message text (after the Strt Pos and Word Num fields have been applied) contains the specified scan text. If the Strt Pos and Word Num fields are blank, then the comparison is true if the scan text appears anywhere in the message.

Example: Use CT to Test a Message

This example uses the CT operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
3	5	AGE	SSAGE	TRUE
1	2	THIS	THIS	TRUE

EQ Operator

The EQ operator tests for an exact match. That is, the (extracted) message text string must match the scan text exactly for the test to succeed.

A wildcard can be either in the scan text or at the end of the scan text.

If, for example, the message text is FREDERICK and the scan text is FRED, the test fails. If, however, the scan text is FRED*, the test succeeds.

Example: Use EQ to Test a Message

This example uses the EQ operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
4	5	SAG	SAGE	FALSE
4	5	SAGE	SAGE	TRUE
1	8	10	100	FALSE
1	8	100	100	TRUE

GE Operator

The GE operator tests whether the value of the (extracted) message text is greater than or equal to that of the scan text. For ResourceView, if the characters to test are numeric, a numeric comparison is executed.

Example: Use GE to Test a Message

This example uses the GE operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
1	8	99	100	TRUE (ResourceView)
1	8	99	100	FALSE (EventView)
1	8	100	100	TRUE
4	Blank	99	100A THIS …	FALSE
4	5	LAGE	SAGE	TRUE
4	5	TAGE	SAGE	FALSE

GT Operator

The GT operator tests whether the value of the (extracted) message text string is greater than the value of the scan text. For ResourceView, if the characters to test are numeric, a numeric comparison is executed.

Example: Use GT to Test a Message

This example uses the GT operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
1	8	99	100	TRUE (ResourceView)
1	8	99	100	FALSE (EventView)
1	8	100	100	FALSE
4	Blank	99	100A THIS …	FALSE
4	5	LAGE	SAGE	TRUE
4	5	TAGE	SAGE	FALSE

LE Operator

The LE to operator tests whether the value of the (extracted) message text string is less than or equal to that of the scan text. For ResourceView, if the characters to test are numeric, a numeric comparison is executed.

Example: Use LE to Test a Message

This example uses the LE operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
1	8	99	100	FALSE (ResourceView)
1	8	99	100	TRUE (EventView)
1	8	100	100	TRUE
4	Blank	99	100A THIS …	TRUE
4	5	LAGE	SAGE	FALSE
4	5	TAGE	SAGE	TRUE

LT Operator

The LT operator tests whether the value of the (extracted) message text string is less than the value of the scan text. For ResourceView, if the characters to test are numeric, a numeric comparison is executed.

Example: Use LT to Test a Message

This example uses the LT operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
1	8	99	100	FALSE (ResourceView)
1	8	99	100	TRUE (EventView)
1	8	100	100	FALSE
4	Blank	99	100A THIS …	TRUE
4	5	LAGE	SAGE	FALSE
4	5	TAGE	SAGE	TRUE

NE Operator

The NE to operator tests for a mismatch between the (extracted) message text string and the scan text.

Example: Use NE to Test a Message

This example uses the NE operator to test the following message:

```
AAA100A THIS IS A MESSAGE NUMBERED MESSAGE 100
```

The following table shows the result of the tests:

Strt Pos	Word Num	Scan Text	Extracted Text	Result
1	8	99	100	TRUE
1	8	100	100	FALSE
4	Blank	99	100A THIS ...	TRUE
4	5	LAGE	SAGE	TRUE
4	5	TAGE	SAGE	TRUE

Expression To Link Tests

The tests you specify in the Text Analysis box are linked in a defined relationship in the Expression field. The Boolean operators AND, OR, and NOT link the tests.

For example, you specify 1 and (2 or 3) in the Expression field. This expression indicates that the following conditions must be satisfied before the rule can be triggered:

- Test 1 must always be true.
- Either Test 2 or Test 3 must be true.

Note: If you leave the Expression field blank, all specified conditions must be true.

EventView Variables

You can use the values of EventView variables as a condition to trigger a rule. You specify the values on the Test EventView Variables panel (the second panel in the extended filter sequence). These values are compared with the values of the predefined EventView variables when the rule is validated. To trigger the rule, they must match.

Execution Conditions

If the message text passes the filtering process, further validation is performed to see whether the message received meets the specified execution conditions.

All the execution conditions specified on the Message Filter panel must be met before the message rule can be triggered. The following shows an example.

```
. Execution Conditions ------------------------------------------------------.
|                                                                             |
| Job Name ....            Rule Priority ...........      (1 is best)        |
| Job Type ....            Execute If Not Best Fit?                           |
|                                                                             |
|         Mon Tue Wed Thu Fri Sat Sun       Time       Start       End       |
| On Days NO  NO  YES NO  NO  NO  NO        Range1 ...                         |
|                                           Range2 ...                         |
|                                                                             |
'-----------------------------------------------------------------------------'
```

Important! If you want to detect a message from a started task that runs under the master scheduler (that is, by using the SUB=MSTR operand), do not use the Job Name and Job Type fields.

Overlapping Rules

You need to take into consideration that there may be more than one rule that applies to the same message.

EventView selects and executes the rule considered to be the best fit. This decision is based on how specific the filtering and execution conditions are; the more specific the rule (for example, the more message text specified), the better the fit.

You can override this determination of the best fit by entering a value (in the range 1 to 99) in the Rule Priority field, to indicate the order of importance. Top priority rules are given a ranking of 1, while the least important rule can be ranked 99.

You may want to trigger multiple rules for one message. The Execute if Not Best Fit? field, which can be set to Yes or to No, functions as follows:

- If set to NO (the default), the rule is not executed unless it is the best fit.
- If set to YES, the rule actions are executed whenever validation is successful.

How You Suppress Messages

You can reduce message traffic to the system and the consolidated console by suppressing messages that operators do not require to perform their tasks. Message suppression does not affect the automated resource monitoring and control functions performed by ResourceView and ServiceView.

Use the following methods to suppress messages:

- Set the Deliver flag from the Message Delivery panel of a message rule. For example, you can specify LOG to suppress messages that trigger the rule from the consoles but enables them to be logged.
- Use the threshold criteria on the Message Delivery panel to suppress redundant messages when multiple messages trigger the rule in a specified time.
- When you have implemented rules for all relevant messages, you can suppress all other messages. To suppress these messages, specify NO or Z in the Default Message Delivery field on the Ruleset Description panel.

  Use the message-learning facility to identify any new messages that have been suppressed. You can then decide whether to create rules for them.

Message Delivery

When a message satisfies the filtering criteria of a rule that is the best fit, the rule controls how the message is delivered. Specify the delivery criteria on the Message Delivery panel.

Set the Deliver Flag

To set the Deliver flag on the Message Delivery panel, specify *one* of the following values:

- YES (the default), if you want to deliver the message to the operating system and the consolidated console, and to log to the system log (SYSLOG) and the activity log
- IGN, if you want the region to ignore the message, but deliver it to the operating system and log it to the system log (SYSLOG)
- LOG, if you want the message logged to SYSLOG and the activity log, but not displayed on the console
- NO, if you want the message suppressed everywhere with the exception of SYSLOG
- Z, if you want the message suppressed everywhere, including SYSLOG

Note: Delivery of system messages to the activity log can be suppressed by the LOGFILES parameter group.

Delivery Thresholds

Thresholds determine what actions are taken when multiple messages trigger the rule in a given time period.

You set thresholds on the Message Delivery panel. You can request that the action associated with the rule be performed before these thresholds are reached, after they are reached, or whenever the rule is triggered, by entering a valid value in the Do Action field.

Note: When a threshold is reached, the value of the Deliver flag is effectively reversed. For example, if the flag is set to NO, messages to which the rule applies are suppressed until the threshold is reached, then delivered to the console. If the flag is set to YES, messages are delivered to the console until the threshold is reached, then suppressed. If the flag is set to LOG, messages are sent to the log until the threshold is reached, then delivered to the console.

How You Use Thresholds When Deliver Flag Is YES

You can specify that you do not want to see the same message more than a given number of times within a certain time interval.

For example, if you do not want to see the same message more than ten times within one minute, you enter the following values:

- **10** in the Maximum Number field
- **00.01.00** in the Time Interval field.

How You Use Thresholds When Delivery Flag Is NO

You can specify that you only want a message displayed if it starts occurring more frequently than usual. You enter a value in the Time Interval field. If more messages of the same kind than the number specified in the Maximum Number field are received in the specified time interval, the messages are displayed. Otherwise, the messages are not displayed.

For example, you want to see every fifth occurrence of a message. You set the Maximum Number field to four and leave the Time Interval field blank (or set to 0). This setting specifies that, no matter how long the interval between occurrences of this message, every fifth occurrence of the message is displayed. All other occurrences of the message are suppressed.

Correlation Key

To avoid creating separate rules for different versions of the same message, you can specify a correlation key on the Message Delivery panel. The rule keeps separate threshold counts for each instance of the correlation key. The separate counts avoid the possible suppression of important but uncommon versions of a message.

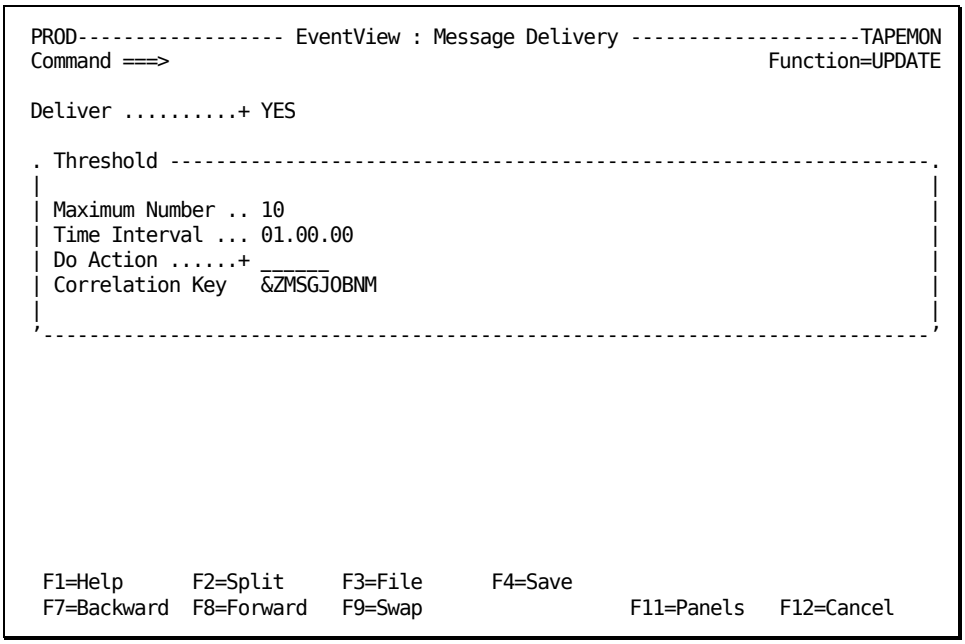
The correlation key can include the following:

- A user-defined EventView variable
- A reference to a ZMSG system variable, such as &ZMSGWORD3

Note: For more information about the system variables, see the *Network Control Language Reference Guide*.

Example: Specify a Correlation Key

This example limits the number of messages (from a given job) that trigger the rule to ten for every hour:



Message Modification

Message presentation and message text can be modified by specifying the requirements on the Message Modification panel.

Message Text Replacement

Using the Replacement Text field on the Message Modification panel, you can replace the entire message text with an alternative text string. The text can include system variables.

Note: For more information about the system variables, see the *Network Control Language Reference Guide*.

System Message Presentation Parameters

To alter how a message is displayed to a system console user, specify the message descriptor code in the Set Descriptor Code field. This code determines the color that the system uses to display the message on a color console, and whether the message is non-roll deletable.

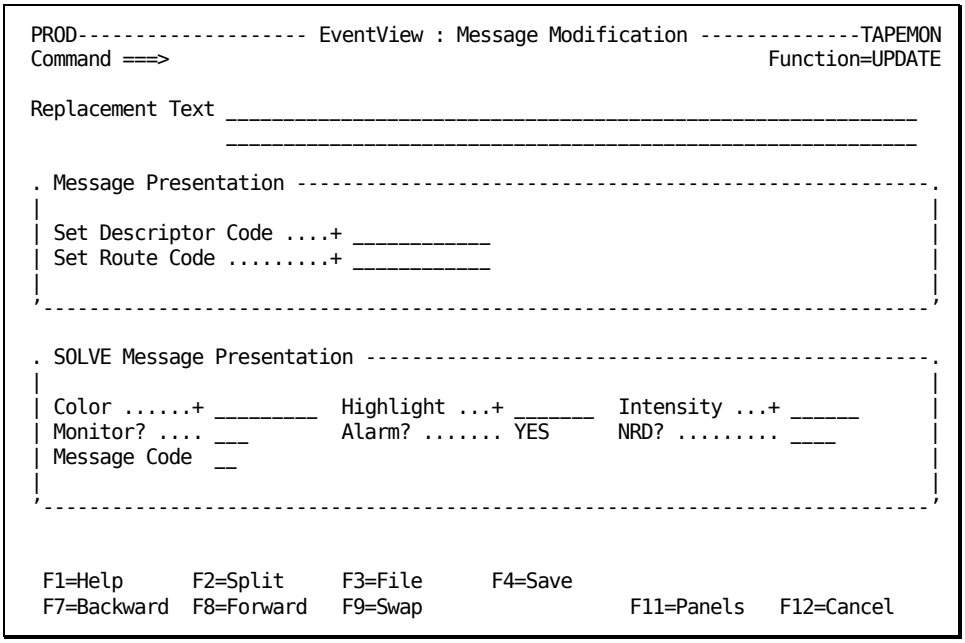
To change the message route code, specify a value in the Set Route Code field, which the system uses to control message delivery.

OCS Message Presentation Parameters

By completing the appropriate fields in the lower box on the Message Modification panel, you can alter how a message is displayed to a user. You can also specify whether a console alarm is sounded when the message is delivered, and whether the message is delivered to monitor class users. The monitor status of a user is set in the user definition and profile.

Example: Sound Alarm on Message Delivery

This example specifies that the console alarm is sounded when the messages that trigger the rule are delivered.



Actions to Take in Response to Messages

Important! Do *not* capture a WTO message and then, using a process or other means, reissue the same WTO message. Reissuing a captured WTO message causes a loop.

On the Message Actions panel, you can specify what response is made to a message. Apart from reply text, you can specify the following:

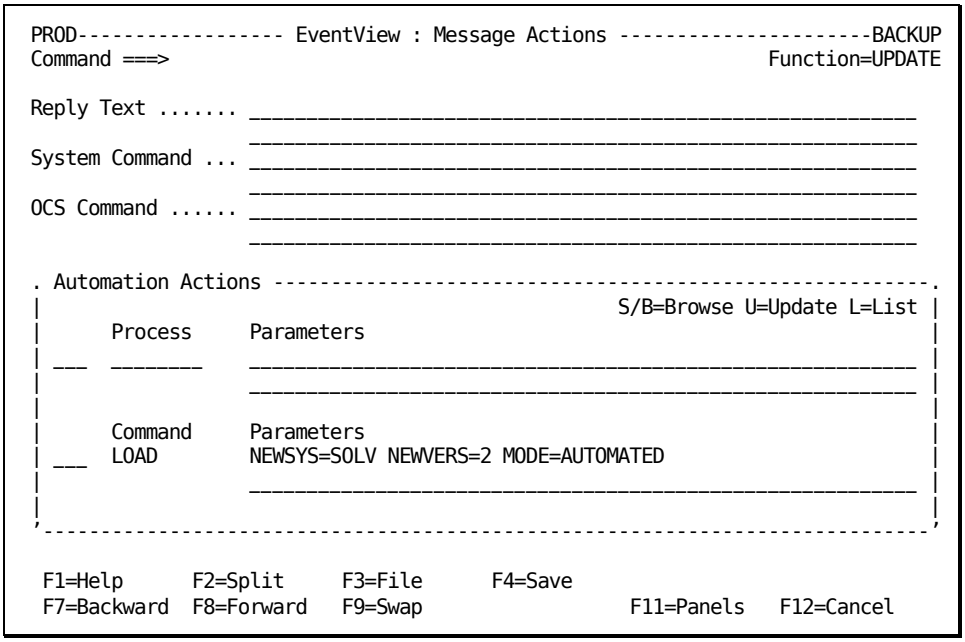
- System command text, such as: START STC1.
- OCS Command text, such as:

  LOG TEST MSG1 ENCOUNTERED-WORD5=&ZMSGWORD5

- A process selected from the list of valid processes—enter a question mark in the field to display a list of valid processes.
- An Automation Services command selected from the list of valid commands—enter a question mark in the field to display a list of valid commands.

Example: Load System Image on Rule Trigger

This example loads a new system image in the local region when the rule is triggered.



Chapter 20: Message Learning

This section contains the following topics:

About Message Learning

The message-learning facility records messages seen by EventView. The facility provides you with a list of all messages encountered during system operation. After you review the initial set of messages, you can reset the new message indicator. Then, when new software is installed, you can easily learn about the new messages.

The facility allows you to do the following:

- List all learnt messages, or all *new* learnt messages
- Display formatted information about listed messages
- Create a message rule from a learnt message
- Use the learnt message list as a prompt list when specifying messages for resource definitions. For more information, see the *Reference Guide*.

Normally, only the first message that starts with a particular word is learnt. However, since some programs issue diverse messages with the same first word, EventView allows for this possibility. EventView also allows you to learn the minor lines of a multiline message. You can enable these features by entering YES in the Learn Multiple Messages? field on the Message Details panel of a learnt message.

Control Message Learning

Message learning can be enabled only if an EventView rule set is loaded with your system image. You control the facility by using the Learn New Messages? field on the Ruleset Description panel of the rule set definition.

To enable message learning for a rule set

1. Enter **/EADMIN.R.R** at the prompt.

   The Ruleset List panel appears.

2. Enter **U** next to the rule set for which you want to enable message learning.

   The panels for rule set definition are listed.

3. Enter **S** next to Ruleset Description.

   The Ruleset Description panel appears.

4. Specify **YES** in the Learn New Messages? field, and press F3 (File).

   Message learning is enabled for the rule set.

Browse and Update Learnt Messages

You can browse and update learnt messages by applying the appropriate action to a listed message.

To display learnt messages

1. Enter **/EADMIN** at the prompt.

   The Event Administration panel appears.

2. Select **L** - Message Learning.

3. Select either **L** - Learnt Messages (to list all learnt messages) or **N** - New Learnt Messages.

4. Apply the **B** (Browse) action to a message you want to browse, or the **U** (Update) action to a message you want to update.

   For example, you may want to update the Learn Multiple Messages? field on the Message Details panel, to indicate that you want EventView to learn multiple messages with the same ID.

5. Select the panel you want to browse or update. Press F1 (Help) for definitions of the fields on the panels.

Generate a Rule for a Learnt Message

If you want to suppress further instances of a message, or to automate the response to the message, you can generate an associated message rule.

To generate a rule for a learnt message

1. From the Learnt Message List, apply the **GR** (Generate Rule) action to a listed item.

   The Ruleset List panel appears.

2. Select the rule set to which you want to add the message rule.

   The initial panel of the generated message rule appears in Add mode. All details stored in the learnt message record that are relevant to message rules have been copied to the message rule record and are displayed in the appropriate fields.

3. Complete the mandatory Short Description field, and complete or update other fields as required.

4. Save the new message rule.

Reset New Message Indicators

If you want to differentiate between messages learnt before and after a certain date, you can reset the new message indicators. You may also want to reset the new message indicators after you review and create rules for the current learnt messages.

Later, you can select the New Learnt Messages option from the Event Message Learning menu to list only those messages that are learnt since you reset the new message indicators (for example, since the last review).

If you list all learnt messages, an asterisk (*) identifies the messages that are flagged as new messages.

To reset new message indicators

1. Enter the **/EADMIN.L** panel path.

   The Event Message Learning menu appears.

2. Select the **R** option.

   The Confirm Database RESET panel appears.

3. Press Enter.

   The new message indicators are reset.

Delete All Learnt Messages

To avoid accumulating too many messages, you can purge all messages after you have viewed those messages that interest you and generated appropriate rules.

Important! Purged messages cannot be recovered.

The AUTOTABLES parameter group controls the size of the table that stores the learnt messages.

To delete all learnt messages, select the **D** option from the Event Message Learning menu.

All learnt messages are purged from the Message Learning database.

Chapter 21: Implementing Message Profiles

This section contains the following topics:

Consolidated Console

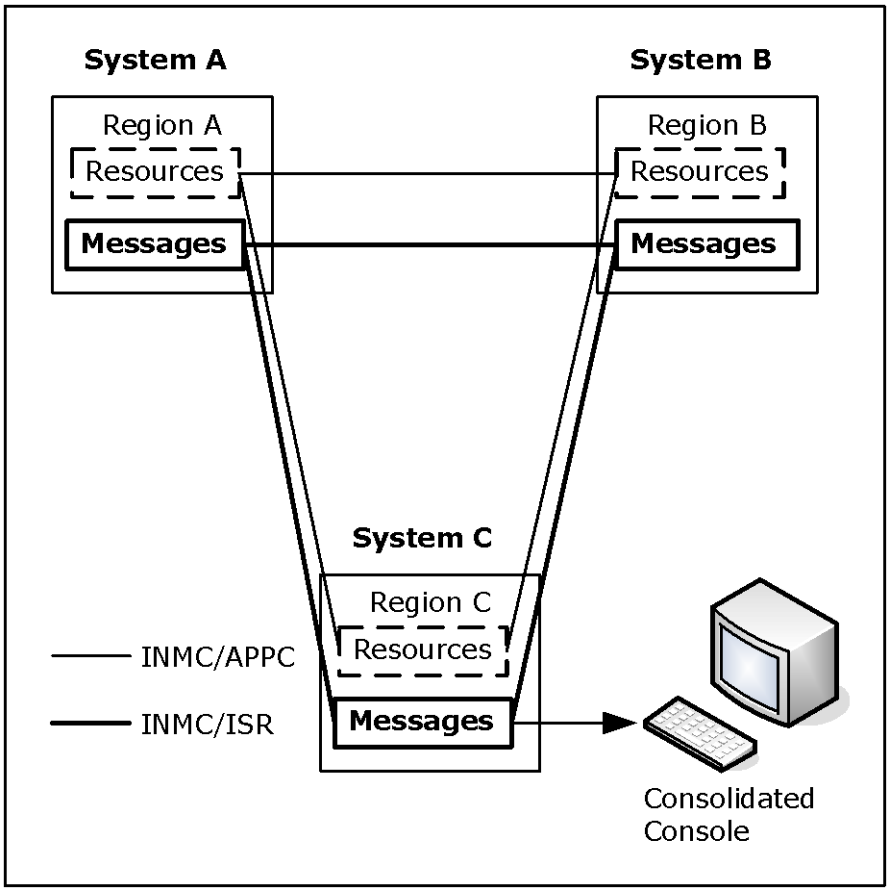
The console consolidation facility consolidates console message traffic from multiple systems onto a single panel (known as a *consolidated console*). Operators can thus view those messages from a single console. You create *message profiles* that contain criteria to identify and classify messages. If a user requests messages for a given message profile, all messages that match the criteria of that profile are displayed on that user's consolidated console.

Note: Multisystem message visibility is available only at consolidated consoles in focal point regions. In subordinate regions, only local messages are visible.

How Console Consolidation Works in a Multisystem Environment

Multisystem support at the message level provides for the distribution of messages to consolidated consoles in focal point regions in the multisystem environment.

The following illustration shows how each region communicates with other connected regions by using Inter-Network Management Connection (INMC)/Inter-System Routing (ISR) links.



Each region has an ISR link manager. The ISR link manager is started up as part of region initialization. The ISR link is active but disabled until a user starts console consolidation. The ISR link manager enables message flow across the link based on requests from users for messages that match specific message profiles.

The user profile determines the messages seen on a consolidated console. The ISR link manager suppresses those messages that are not required, thus reducing the amount of message flow. A user who has not been assigned message profiles or has all the assigned message profiles disabled sees no messages on the consolidated console.

Message Profiles

You can define profiles that capture different types of messages. When you create or change a message profile, the data is automatically distributed to the knowledge bases in connected regions.

Note: You can define message profiles in focal point regions only; however, the defined profiles are available to subordinate regions through knowledge base synchronization.

A message profile contains the following criteria types that determine which messages a consolidated console receives:

- The system from which the message comes
- The ID (or the first word) of the message
- The job for which the message is generated
- The message routing and descriptor codes
- The message types and levels, and the types and classes of job for which the message is generated

A message profile must use at least one criterion from the last four criteria types.

Each profile has a status that determines whether it can be activated for use. Profiles must be activated, either by you or automatically during region startup, before they can be used. After you define the profiles, you activate them for use by the operators.

Rules for Defining and Using Message Profiles

This section contains rules about entering data on panels and about how to get the best results when defining message profiles.

Create New Message Profiles in a Single System First

Create a new message profile to select messages from one system only, using selection criteria that are unique to that system. For example, if each system uses different message classes, specify a message class that is unique to your system. When this profile is working successfully in one system, you can copy it into a new profile for other systems whose messages you want to monitor.

Unique Message Profile Names and IDs

Unique profile names and IDs identify message profiles. When messages are captured, they are associated with a specific profile ID. The profiles replace the message routing codes corresponding to the IDs as the means for the region to direct relevant messages to operators. An operator who wants to receive specific messages on a consolidated console enables the relevant profiles. Alternatively, if the operator always wants to see consolidated messages for certain profiles, the operator can specify this information in the user profile (see page 241).

Important! A profile acts on messages after they are processed by EventView message rules. For example, if a rule changes the routing code and you want to capture the message, use a profile ID that corresponds to the changed routing code.

You cannot include special characters (for example, _, -, (,), and ~) or spaces in a profile name.

Wildcards

Use wildcard characters to represent character patterns at particular positions in a character string. The supported wildcard characters are as follows:

- *, representing any character as follows:
  - If the * is at the beginning of or embedded in a character string, it represents one character.
  - If the * is at the end of a character string, it represents any number of characters.

  You *cannot* use an * by itself. In the following example, messages are selected for any system that starts with the letters EAST:

  Systems to Include
  EAST*
- #, representing one numeric character. In the following example, messages are selected for systems EAST0 through EAST9:

  Systems to Include
  EAST#
- @, representing one alpha character. In the following example, messages are selected for systems EAST0A through EAST9Z:

  Systems to Include
  EAST#@

Type as many characters as necessary to select the required information.

If you want to use a wildcard character in the literal sense, precede the character by a backslash (\), for example:

- ABC### matches any value that starts with ABC followed by three numeric characters.
- ABC##\# enables you to match a value that starts with ABC followed by two numeric characters and ending in a # character.

Ranges

Use a colon (:) to specify ranges.

The character strings on each side of the colon must be of equal length.

Note: The backslash (\) is regarded as one character when the length of the string is calculated.

The asterisk (*) wildcard character can only be used at the end of a string.

Example: Select Messages in a Range of Systems

This example selects messages for systems EAST0, EAST1, and EAST2.

```
Systems to be Included
EAST0:EAST2
```

Inclusion and Exclusion Criteria

In each profile, you specify the criteria that determine the messages to display on the consolidated console. Most panels have *inclusion fields* and *exclusion fields*, or allow you to specify N(o) or Y(es), according to whether you want to include or exclude messages with certain attributes. The rules for including and excluding messages are as follows:

- If you leave all the fields on a panel blank, the criteria specified on the other panels determine what messages are displayed. For example, if you leave the System Specification panel blank, messages from all connected regions are potentially available for display.

  However, if you do *not* specify any criteria (that is, if you leave the criteria fields on all the panels blank), the profile receives *no* messages.

- If you specify inclusion and exclusion values for a particular criterion, the inclusion values take priority. The exclusion values are then applied to the resulting set of included messages.

  For example, using message ID as a criterion, if you want to include all message IDs except the IDs starting with AAA111, you can use the following values:

  - A*:9* as inclusion values
  - AAA111* as exclusion values

- Messages are selected for display only if they meet the criteria specified on all panels. For example, YES in the Sess field on the Message Job Specification panel specifies that only SESS type messages are displayed, even if messages of other types meet the criteria specified on the other panels.
- Items selected with N or Y have an OR relationship. For example, if you include routing codes 1, 2, and 11, messages that have a routing code of 1 *or* 2 *or* 11, *or* a combination of these codes, are displayed if they satisfy the other criteria.
- If you complete an exclusion field or specify N, a message that meets this criterion is not displayed, even if it satisfies all the inclusion criteria.

Note: The consolidated console does not receive messages suppressed by EventView rules.

Access the Message Profile Definitions

To access the message profile definitions, enter **/EADMIN.C.M** at the prompt.

The Message Profiles panel appears.

This panel lists all the message profiles in the knowledge base. You can enter action codes to perform actions on existing message profiles, or press F4 (Add) to add a new profile.

How You Define a Message Profile

To add a message profile, press F4 (Add) from the Message Profiles panel. You define the profile by using the following panels:

Profile Details

Enables you to identify the profile. Complete this panel.

System Specification

Enables you to use the system associated with a message as a selection criterion.

Message Specification

Enables you to use the message ID as a selection criterion.

Job Name Specification

Enables you to use the job associated with a message as a selection criterion.

OS Codes Specification

Enables you to use the routing and descriptor codes associated with a message as selection criteria.

Message Job Specification

Enables you to use the message type and level, and the job type and class associated with a message as selection criteria.

You can create a profile to capture particular messages (for example, tape mount messages) or messages for particular jobs (for example, production CICS jobs). You do not need to complete every panel for most profile definitions. However, complete at least one of the criteria panels. If you leave all the criteria panels blank, the profile blocks all messages.

Profile Details

Use the Profile Details panel to identify the message profile. Specify the profile name, ID, and description. All profile panels contain this information.

Note: You cannot use the value 2 as the profile ID.

Only profiles that have IDs corresponding to those set for a parameter in the CCONSOLIDATN parameter group are available for use in the local region. The parameter can exclude certain IDs. To display the value of the parameter, enter the **/PARMS** shortcut to access the list of parameter groups and browse the CCONSOLIDATN parameter group.

The Profile Details panel also contains the following information:

- Profile status
- Whether to profile for solicited messages
- History of when the profile was created and last updated

Only profiles with an ACTIVE status can be activated for use.

System Criteria

From the Profile Details panel, press F8 (Forward) to display the System Specification panel. You can specify the systems for which messages are captured.

The values you use in the Systems to be Included or Excluded fields are the system management facilities (SMF) ID or the region domain ID. The value type is indicated at the bottom of the panel as SMFID or NMDID respectively, and is set in the CCONSOLIDATN parameter group.

The criteria can be specific, generic, or in a range.

Leave the fields blank to allow messages for all the connected systems to be captured. These messages are captured only if other criteria are specified in the profile and the messages satisfy those criteria.

Message ID Criteria

From the System Specification panel, press F8 (Forward) to display the Message Specification panel. You can specify the IDs (or generic IDs, for example, $HASP*) of the messages you want to capture. The message ID is the first word of a message.

The values can be specific, generic, or in a range.

Leave the fields blank to capture messages with any ID. These messages are captured only if other criteria are specified in the profile and the messages satisfy those criteria.

Job Criteria

From the Message Specification panel, press F8 (Forward) to display the Job Name Specification panel. You can name the jobs (and started tasks) for which messages are captured.

The values can be specific, generic, or in a range.

Leave the fields blank to capture messages for all jobs. These messages are captured only if other criteria are specified in the profile and the messages satisfy those criteria.

System Codes Criteria

From the Job Name Specification panel, press F8 (Forward) to display the OS Codes Specification panel. You can specify the route and descriptor codes assigned to messages that are captured. Messages can contain one or a combination of the codes you specify. If a message contains codes that you exclude specifically, the message is not selected.

You can exclude certain codes by typing **N** under the codes, include certain codes by typing **Y** under the codes, and leave the other code fields blank. A message containing any of the included codes is selected unless the message also contains an excluded code.

Leave the fields blank to capture messages that contain any route and descriptor codes. If other criteria are specified in the profile and the message satisfies those criteria, the message is displayed on the consolidated console.

Message Type, Level, and Job Criteria

From the OS Codes Specification panel, press F8 (Forward) to display the Message Job Specification panel. You can specify the message types, and message levels, job types, and job classes assigned to messages that are captured.

You can include or exclude certain items in each criteria type, but not both (except for the Broadcast field under Message Levels). For example, if you want to accept immediate action messages but not broadcast messages, specify Y in the Immediate Action field and N in the Broadcast field.

Leave the fields blank to allow messages of any type, level, job type, or job class. If other criteria are specified in the profile and the message satisfies those criteria, the message is displayed on the consolidated console.

Message Types

Message types correspond to the operands of the MONITOR or STOPMN system commands. For example, messages generated because of the MONITOR SESS command have the SESS type.

Message Levels

Message levels indicate the relative importance of a message.

Note: The broadcast level has precedence over all other message criteria. If broadcast messages are allowed, the message profile passes all broadcast messages irrespective of the other criteria.

Job Types

Job types are as follows:

Job

Indicates a batch job.

STC

Indicates a started task.

In a JES3 environment, a started task has a job type of Job.

TSU

Indicates a TSO user.

Unknown

Indicates a job type that is not one of the previous types.

Job Classes

The job class is assigned by the CLASS parameter of the JOB JCL statement.

Example: Profile Specific Messages

In this example, the organization has two branches: an eastern branch and a western branch. You want to create a profile to capture all tape mount messages for all the production systems running in the eastern data center, but do not want to capture messages for development jobs. The job classes assigned to tape mount requests are 1, 2, and 3.

From the Message Profiles panel, press F4 (Add) to add a profile. A Profile Details panel appears. You can press F8 (Forward) to scroll forward through the panels or enter the panel index number at the Command prompt to select a specific panel. (Pressing F11 (Panels) displays of a list of panels and panel index numbers.)

Use the following panels in this example:

- Profile Details
- System Specification
- Message Specification
- Job Name Specification
- Message Job Specification

On the Profile Details panel, type a unique profile name (TAPEMOUNTS), a unique ID (127), a description of the profile (Tape Mounts for Eastern Production Jobs), and assign a status. Assign a status of ACTIVE so that the profile can be activated.

The following panel shows the completed Profile Details.

```
PROD------------------- EventView : Profile Details -------------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile -----------------------+
| Name ... TAPEMOUNTS       ID ..... 127 (1 - 128)                            |
| Description .... Tape Mounts for Eastern Production Jobs_____________________ |
+--------------------------------------------------------------------------------+
+ Profile Status ------------------------------------------------------------+
| Profile Status ... ACTIVE__ (Active/Inactive)                              |
+--------------------------------------------------------------------------------+
+ Include Solicited Messages? -----------------------------------------------+
| Solicited Type ... NO___    (No, Other, Nothr, Yes, All)                   |
+--------------------------------------------------------------------------------+

+ History -------------------------------------------------------------------+
|                                                                            |
| Profile Created           Profile Last Updated      Profile Status Updated |
| Userid  USER01            Userid                    Userid                 |
| Date .. THU 25-MAY-2006   Date ..                   Date ..                |
| Time .. 14.48.27          Time ..                   Time ..                |
+--------------------------------------------------------------------------------+

 F1=Help      F2=Split     F3=File      F4=Save
              F8=Forward   F9=Swap                 F11=Panels   F12=Cancel
```

Press F8 (Forward) to scroll forward to the System Specification panel. You do not want to capture messages for any western branch systems, so you complete the exclusion fields. All western branch systems start with the letters WST, so WST* is typed to exclude all western branch systems.

The following panel shows the completed System Specification.

```
PROD---------------- EventView : System Specification -----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... TAPEMOUNTS       ID ..... 127 (1 - 128)                           |
| Description .... Tape Mounts for Eastern Production Jobs                   |
+----------------------------------------------------------------------------+
+-----------------------------------+ +-----------------------------------+
| Systems to be Included            | | Systems to be Excluded            |
+-----------------------------------+ +-----------------------------------+
| _______________ _______________ | | WST*____________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
+-----------------------------------+ +-----------------------------------+

Note : Identify the systems by using SMFID


 F1=Help      F2=Split     F3=File      F4=Save
 F7=Backward  F8=Forward   F9=Swap                F11=Panels   F12=Cancel
```

Press F8 (Forward) to scroll forward to the Message Specification panel. You only want to display IEF233A messages, which are requests for tape mounts, so you complete the inclusion fields.

The following panel shows the completed Message Specification.

```
PROD---------------- EventView : Message Specification -----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... TAPEMOUNTS       ID ..... 127 (1 - 128)                           |
| Description .... Tape Mounts for Eastern Production Jobs                   |
+----------------------------------------------------------------------------+
+-----------------------------------+ +-----------------------------------+
| Message IDs to be Included        | | Message IDs to be Excluded        |
+-----------------------------------+ +-----------------------------------+
| IEF233A_________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
| _______________ _______________ | | _______________ _______________ |
+-----------------------------------+ +-----------------------------------+




 F1=Help      F2=Split     F3=File      F4=Save
 F7=Backward  F8=Forward   F9=Swap     F10=Scrllst  F11=Panels   F12=Cancel
```

Press F8 (Forward) to scroll forward to the Job Name Specification panel. You do not want to capture messages for development jobs. All development jobs in the eastern branch start with the letters DEV, so DEV* is typed in an exclusion field.

The following panel shows the completed Job Name Specification.

```
PROD---------------- EventView : Job Name Specification ----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile -----------------------+
| Name ... TAPEMOUNTS       ID ..... 127 (1 - 128)                            |
| Description .... Tape Mounts for Eastern Production Jobs                    |
+-----------------------------------------------------------------------------+
+-----------------------------------+ +-----------------------------------+
| Job Names to be Included          | | Job Names to be Excluded          |
+-----------------------------------+ +-----------------------------------+
| ________________ ________________ | | DEV*_____________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
+-----------------------------------+ +-----------------------------------+



 F1=Help      F2=Split     F3=File      F4=Save
 F7=Backward  F8=Forward   F9=Swap     F10=Scrllst  F11=Panels   F12=Cancel
```

Enter **6** at the prompt to display the Message Job Specification panel. Here you want to capture messages for jobs only, in job classes 1 (for jobs that need one tape mounted), 2 (for jobs that need two tapes mounted), and 3 (for jobs that need three tapes mounted).

The following panel shows the completed Message Job Specification.

```
PROD-------------- EventView : Message Job Specification ---------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile -----------------------+
| Name ... TAPEMOUNTS       ID ..... 127 (1 - 128)                            |
| Description .... Tape Mounts for Eastern Production Jobs                    |
+-----------------------------------------------------------------------------+
+-----------------------------------------------------------------------------+
| Message Types        ( Y =Include, N =Exclude, Blank =Don't care )           |
|  Jobnames .. ___   Status .. ___   Active .. ___   Sess .. ___               |
|                                                                             |
| Message Levels       ( Y =Include, N =Exclude, Blank =Don't care )           |
|      WTOR .. ___  Immediate Action .. ___  Critical Eventual .. ___          |
|  Eventual .. ___     Informational .. ___          Broadcast .. ___          |
|                                                                             |
| Job Types            ( Y =Include, N =Exclude, Blank =Don't care )           |
|       Job .. YES   STC .. ___   TSU .. ___   Unknown .. ___                 |
|                                                                             |
| Job Classes          ( Y =Include, N =Exclude, Blank =Don't care )           |
|              ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789                           |
| (A-Z,0-9) ..                          YYY                                   |
+-----------------------------------------------------------------------------+
 F1=Help      F2=Split     F3=File      F4=Save
 F7=Backward              F9=Swap                 F11=Panels    F12=Cancel
```

Example: Profile CICS Messages

In this example, you want to create a profile to capture all messages that satisfy the following conditions:

- The message identifier starts with DFH, indicating a CICS message.
- The message comes from CICS regions CICSA through CICSI, excepting CICSD, which is a development region.

From the Message Profiles panel, you press F4 (Add) to add a profile. A Profile Details panel appears. You can press F8 (Forward) to scroll forward through the panels or enter the panel index number at the Command prompt to select a specific panel. (Pressing F11 (Panels) displays a list of panels and panel index numbers.)

Use the following panels in this example:

- Profile Details
- Message Specification
- Job Name Specification

The following panels show the completed message profile:

```
SOLVPROD------------------ EventView : Profile Details -----------MCPROFIL-0000
Command ===>                                                  Function=BROWSE

.---------------------- Message Classification Profile ----------------------.
| Name ... CICSMSG          ID ..... 128 (1 - 128)                            |
| Description .... All CICS messages except those from CICSD                  |
'------------------------------------------------------------------------------'

. Profile Status ------------------------------------------------------------.
| Profile Status ... ACTIVE   (Active/Inactive)                               |
'------------------------------------------------------------------------------'
```

```
SOLVPROD--------------- EventView : Message Specification --------MCPROFIL-0000
Command ===>                                                  Function=BROWSE

.---------------------- Message Classification Profile ----------------------.
| Name ... CICSMSG          ID ..... 128 (1 - 128)                            |
| Description .... All CICS messages except those from CICSD                  |
'------------------------------------------------------------------------------'
.-----------------------------------. .-----------------------------------.
| Message IDs to be Included        | | Message IDs to be Excluded        |
+-----------------------------------+ +-----------------------------------+
| DFH*                              | |                                   |
|                                   | |                                   |
```

```
SOLVPROD-------------- EventView : Job Name Specification --------MCPROFIL-0000
Command ===>                                                  Function=BROWSE

.---------------------- Message Classification Profile ----------------------.
| Name ... CICSMSG          ID ..... 128 (1 - 128)                            |
| Description .... All CICS messages except those from CICSD                  |
'------------------------------------------------------------------------------'
.-----------------------------------. .-----------------------------------.
| Job Names to be Included          | | Job Names to be Excluded          |
+-----------------------------------+ +-----------------------------------+
| CICSA:CICSI                       | | CICSD                             |
|                                   | |                                   |
```

Example: Profile Messages for Specific Jobs

In this example, you want to create a profile to capture messages for certain CICS jobs on the production systems in the eastern and the western branches. The branches use only one test system, ETST. You assign a status of INACTIVE, as you do not want the profile to be used immediately. You only want to capture messages that have routing codes of 1, 2, or 11.

From the Message Profiles panel, you press F4 (Add) to add a profile. A Profile Details panel appears. You can press F8 (Forward) to scroll forward through the panels or enter the panel index number at the Command prompt to select a specific panel. (Pressing F11 (Panels) displays a list of panels and panel index numbers.)

Use the following panels in this example:

- Profile Details
- System Specification
- Job Name Specification
- OS Codes Specification

The following panels show the completed message profile:

```
PROD------------------- EventView : Profile Details -------------MCPROFIL-0000

Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... CICSMESSAGES     ID ..... 126 (1 - 128)                           |
| Description .... Messages for all production CICS Jobs___________________ |
+------------------------------------------------------------------------------+

+ Profile Status -------------------------------------------------------------+
| Profile Status ... INACTIVE (Active/Inactive)                              |
```

```
PROD----------------- EventView : System Specification -----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... CICSMESSAGES     ID ..... 126 (1 - 128)                           |
| Description .... Messages for all production CICS Jobs                     |
+------------------------------------------------------------------------------+
+-----------------------------------+ +-----------------------------------+
| Systems to be Included            | | Systems to be Excluded            |
+-----------------------------------+ +-----------------------------------+
| ________________ ________________ | | ETST____________ ________________ |
```

```
PROD---------------- EventView : Job Name Specification ----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... CICSMESSAGES     ID ..... 126 (1 - 128)                           |
| Description .... Messages for all production CICS Jobs                     |
+------------------------------------------------------------------------------+
+-----------------------------------+ +-----------------------------------+
| Job Names to be Included          | | Job Names to be Excluded          |
+-----------------------------------+ +-----------------------------------+
| CICSPRD1:CICSPRD9 CICSTST*_______ | | CICSPRD4_________ ________________ |
```

```
PROD--------------- EventView : OS Codes Specification -----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... TAPEMOUNTS       ID ..... 127 (1 - 128)                           |
| Description .... Messages for all production CICS Jobs                     |
+------------------------------------------------------------------------------+
+------------------------------------------------------------------------------+
| Routing Codes    ( Y =Include, N =Exclude, Blank =Don't care )               |
|                    1         2         3         4         5         6     |
|           12345678901234567890123456789012345678901234567890123456789012345678  |
|   1-64 => YY        Y                                                      |
```

Example: Profile All Messages

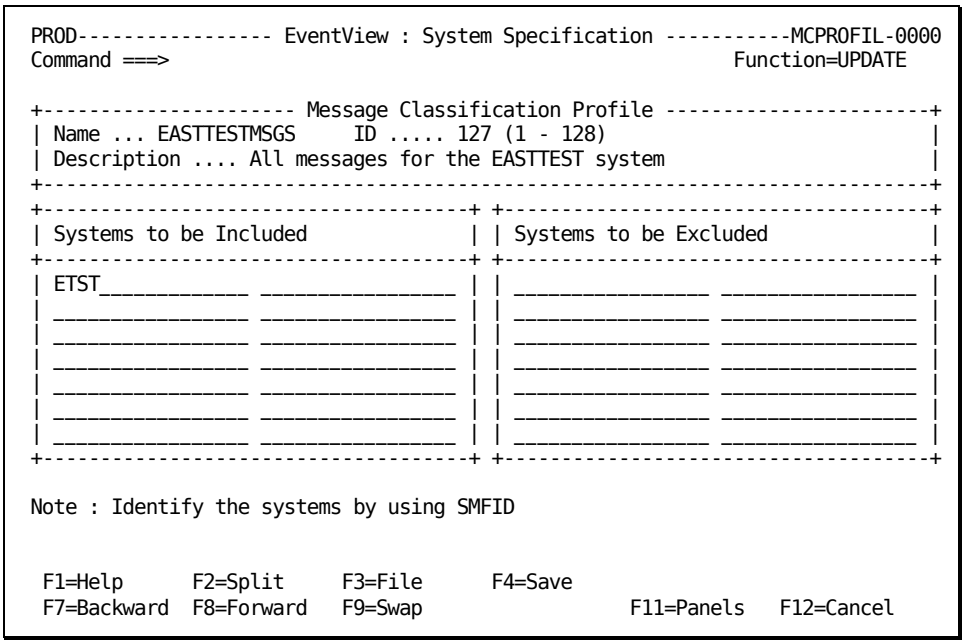
Note: This example is for illustration only. In a multisystem environment, if you have not implemented EventView message rules to provide a high level of message suppression, using this message profile can result in a very high volume of message flow to the consolidated console.

In this example, you want to create a profile to capture the messages for all connected systems. You allow all messages by excluding a system that is not part of the network. The following shows an example where DMMY is the excluded system.

```
PROD---------------- EventView : System Specification -----------MCPROFIL-0000
Command ===>                                                  Function=UPDATE

+---------------------- Message Classification Profile ----------------------+
| Name ... ALLMESSAGES      ID ..... 127 (1 - 128)                           |
| Description .... All messages                                              |
+----------------------------------------------------------------------------+
+-----------------------------------+ +-----------------------------------+
| Systems to be Included            | | Systems to be Excluded            |
+-----------------------------------+ +-----------------------------------+
| ________________ ________________ | | DMMY____________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
| ________________ ________________ | | ________________ ________________ |
+-----------------------------------+ +-----------------------------------+

Note : Identify the systems by using SMFID


 F1=Help      F2=Split     F3=File      F4=Save
 F7=Backward  F8=Forward   F9=Swap                F11=Panels   F12=Cancel
```

Example: Profile Messages for a Particular System

In this example, you want to create a profile to capture the messages for a particular system. The following shows an example where ETST is the system whose messages you want to monitor.



Change the Activation Status of a Message Profile

A message profile must have an ACTIVE status before it can be activated for use. If you only want to change the Profile Status Field for a profile, change the value directly from the Message Profiles panel.

To change the value to ACTIVE, type **A** next to all the profiles you want to update and press Enter.

The value in the Status column for the profiles changes to ACTIVE.

To change the value to INACTIVE, type **I** next to all the profiles you want to update and press Enter.

The value in the Status column for the profiles changes to INACTIVE.

Activate Message Profiles

Note: The message profile activation process can halt the region for a short period of time. After this period, the region continues from where it left off, without loss of control or data. However, delays might occur in responses to system activities. Unless the activation of the message profiles is of a high priority, perform this task when the system is not busy.

After you have created or updated message profiles, you must activate (load) them in each of the linked regions before they can be used.

To activate message profiles, use *one* of the following methods:

- Select the **A** option on the System Console Consolidation panel, or enter the **/EADMIN.C.A** path (available to focal point regions only).
- Enter **ACTIVATE** at the prompt on the Message Profiles panel (available to focal point regions only). To display the panel, enter the **/EADMIN.C.M** path.
- Action the CCONSOLIDATN parameter group (available to focal point and subordinate regions). To display the list of parameter groups, enter the **/PARMS** shortcut.

A region only activates profiles with a status of ACTIVE.

Profiles with a status of ACTIVE also become active automatically whenever the region is started. Profiles with a status of INACTIVE are not activated when the region is started.

Message Profile Size Considerations

If the total size of the profiles loaded is too large, activation of message profiles can fail. If the problem occurs, a message is generated to indicate by how much the size should be reduced. The ID of the message is either RMCCST11 or RMINWI36.

Note: For information about how to correct the problem, see the message online help.

Maintenance of Message Profile Definitions

In a focal point region, you can browse, update, copy, and delete message profile definitions from the Message Profiles panel.

Note: For information about how to assign message profiles to individual users, see the *Security Guide*.

Chapter 22: Configuring the Event Simulator

This section contains the following topics:

Event Simulator

The event simulator enables you to correctly assess the impact of a loaded system image on the operations of the local system. The MSGAWARENESS parameter group controls the availability of the simulator.

By using the simulator, you can generate simulated events and review the returned results. A simulated event returns the expected results. It does not invoke the actual actions. The results of the simulation identify the following affected active definitions:

- Resource definitions
- EventView rules
- Consolidated console message profiles
- Other product-specific definitions and records

Generate Simulated Events

To generate simulated events

1. Enter **/EADMIN.E** at the prompt.

   The Simulated Events List appears.

2. Do *one* of the following:
   - If the required event definition is not on the list, press F4 (Add) to define and generate the event.
   - If the required event definition is on the list, do one of the following:
     - Use the SV or SI action code to simulate one or more defined events.
     - Enter **ALL SI** at the prompt to simulate all defined events.

Define a Simulated Event

To define a simulated event

1. From the Simulated Event List, enter **/EADMIN.E** at the prompt.
2. The simulated event definitions appear.
3. Press F4 (Add). You can also use the C action code to open a copy of an existing definition that you can modify.

   The Simulate Message panel appears.
4. Specify the message you want to simulate and the type of information you want returned.

   You can enter a question mark (?) in the Message Text field to display the list of messages learned by the region. If you select a message from the list, the panel is automatically updated for any associated job name, routing codes, and descriptor codes.
5. Do *one* of the following:

   - If you want to generate the simulated event, press F6 (Simulate). To save the results, press F3 (File) or F4 (Save).
   - If you do not want to generate the simulated event now, press F3 (File) to save the definition for later use.

   Note: Filed message definitions are *not* retained across region restarts.

Results of Event Simulation

The results of event simulation are returned on the Simulation Results List panel.

```
PROD-------- Automation Services : Simulation Results List --------------------
Command ===>                                                  Scroll ===> CSR

                 S/B=Browse Definition U=Update Definition C=Collapse E=Expand
    Simulated Message Details:
     Message Text ... $HASP170 PRT1     INTERRUPTED
     Jobname ........ JES2           Jobtype ....... JOB  Message Type ... WTO
     Route Codes .... 7              Desc. Codes ... 4

    ************************** Simulation Results ****************************
    Dflt EventView Ruleset ..... $$$$$URS
         Default ruleset processing performed as per:
            Message Delivery ... YES           Perform Mods.? .. YES
            Perform Actions? ... YES           Log Activity? ... NO
            Collect Statistics?  YES           Learn New Msgs?   NO

    Miss No Consolidated Console profiles hit for the following reason:
            No Consolidated Console Profiles Hit

    Hit  PRT Resource Name ..... PRT1         JES Printer PRT1
         Monitor Message ....... $HASP170 PRT1*
         Extended Actions:
```

For the previous example, the results indicate that the:

- Messages are passed on by the $$$$$URS rule set but no rules are triggered
- PRT1 resource becomes degraded but no actions are invoked

If the results are not satisfactory for a displayed definition, you can use the U action code to update it. For example, if you enter U next to the PRT resource line, the Status Monitor Message Details panel displays. You can then update the appropriate resource message rule.

Summarize the Results

When a simulated event affects many definitions, the results are displayed over several panels; however, you can summarize the results.

To summarize the results, enter **ALL C** at the prompt.

The results appear as a list of affected definitions.

Note: You can use the ALL E command to display all details of all the results. You can use the C and E action codes to change the view of selected results.

Maintenance of Simulated Event Definitions

You can browse, update, copy, and delete simulated event definitions. To delete all definitions, enter **ALL D** at the prompt.

Note: If you update the message attributes, you are creating a message. Previously stored simulation results are not retained.

Chapter 23: Implementing Activity Logs

This section contains the following topics:

Activity Logs

The activity logging facility records all the activity in your region. You can use the activity logs to help determine the cause of problems.

Two separate activity log formats exist:

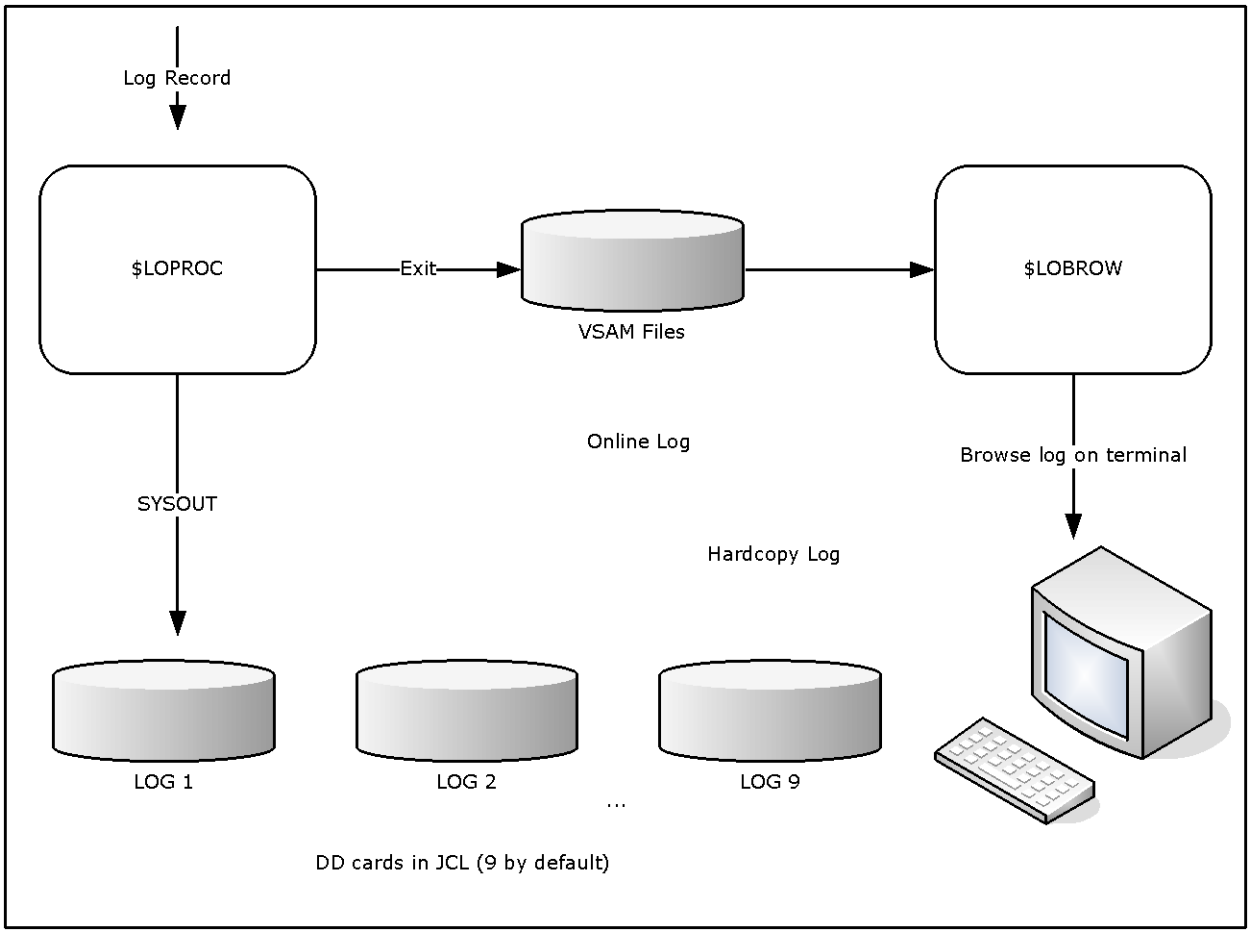
- Online
- Hardcopy

Log records are written to both formats.

By default, activity logs contain the following information:

- All commands entered
- All responses to commands entered
- Any unsolicited messages received from VTAM or the operating system, provided the related interfaces are available
- All messages explicitly written to the log by NCL procedures

The following illustration shows the path that the log record takes in the system.



The online activity log is supplied by the distributed procedure $LOPROC. The $LOPROC procedure writes log data to VSAM files (three by default). The VSAM files are accessed by a second procedure, $LOBROW, which allows online browsing of the log.

Note: $LOPROC and $LOBROW are the default procedure names. You can change these names by using the LOGFILES parameter group in Customizer (**/PARMS**).

Customize Activity Log Settings

The activity logs record system messages and messages that occur in the region. You can customize the LOGFILES parameter group to do the following to suit the requirements of your site:

- Disable the logging of system messages
- Allocate additional activity log files

To customize the LOGFILES parameter group

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups panel appears.

2. Enter **U** next to the LOGFILES parameter group.

   Fields appear that let you customize the parameter group.

3. Complete the fields, as required, and press F6 (Action).

   The new settings are activated.

4. Press F3 (File).

   The information is saved.

Disable System Message

By default, system messages are delivered to the activity log.

To disable system message logging, set the value in the Log Operating System Messages? field to **NO**.

Disable Command Logging

By default, system commands received by the region are delivered to the activity log.

To disable system command logging, set the value in the Log Commands? field to **NO**.

Allocate Activity Log Files

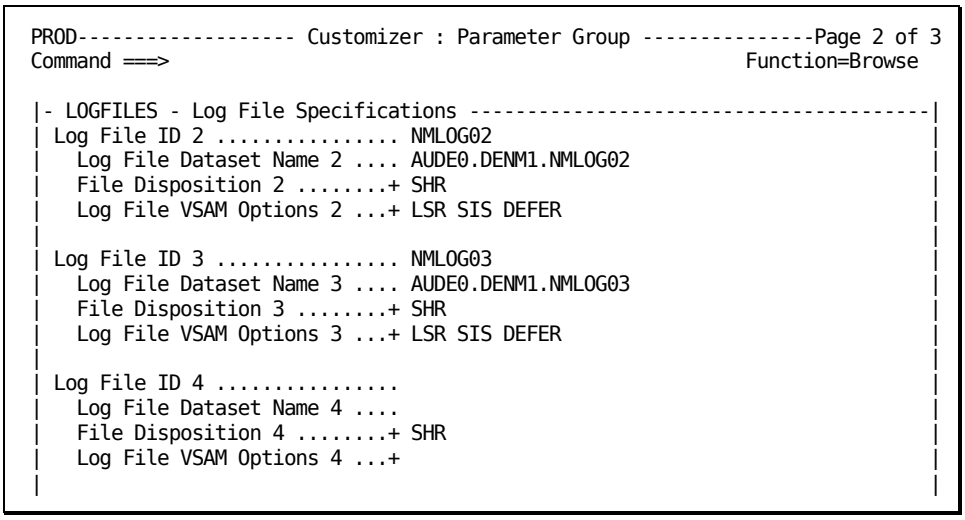
During initialization, the region is allocated three activity log files. However, you can allocate up to seven files.

Note: As supplied, the log file IDs and data set names are, respectively, NMLOG*nn* and *dsnpref*.NMLOG*nn*. (*dsnpref* is the data set prefix used during the setup of the region).

To make more than three files available to the region

1. Define additional logging data sets.
2. After the first Parameter Group panel opens, press F8 (Forward).

   The next panel appears.
3. Complete the fields for each file you want to make available. The following is an example:



To allocate more files

1. Press F8 (Forward).
2. Press F6 (Action)

   The new settings are applied.
3. Press F3 (File).

   The information is saved.

Administer Online Activity Log Files

From the Activity Log : Administration menu, you can do the following:

- Swap active activity logs
- List all days contained in log files and browse logs for a particular date
- List all log files and browse a particular file

To administer online activity log files, enter **/LOADMIN** at the prompt.

The Activity Log : Administration menu appears.

Note: For information about the options available on this menu, press F1 (Help).

Swap the Online Log

The online activity log automatically swaps to a fresh VSAM file when each file fills up.

You can manually swap your currently active VSAM file if you want to free a particular log file (for example, for backups).

Important! Swapping the current VSAM log causes the $LOPROC procedure to write all subsequent activity log records to the next VSAM log. If this log was previously used, it is reset. Therefore, you can no longer browse the old records that it contained.

To swap the online activity log

1. Enter **/LOGSWAP** at the prompt.

   The Activity Log Services : Confirm Swap Log panel appears.

2. Press F6 to request the log swap, or F12 to cancel your request.

   Note: If the $LOPROC procedure encounters a VSAM error when it is logging activity to an online log file, it automatically swaps to the next log file.

Online Log Exit

You can create an NCL procedure to intercept, analyze, and react to the messages that are sent to the activity log.

Use the LOGFILES parameter group in Customizer to specify the name of your exit.

The exit is executed every time a message is sent to the log. Using the exit to perform complex functions can degrade the performance of the region.

Note: Ensure that your log exit procedure is well-tested before you put it into production.

Variables Available to the Activity Log Exit

The following variables are available to the activity log exit:

&#LO$RECORD

Contains records of the following formats:

time_generated user_id terminal_id message_text

The text of the message starts at the fourth word of the record.

arrival_time origin region* $$AOMTIME$$ *aom_time message_text

The text of the message starts at the sixth word of the record. This format lets you identify AOM-sourced messages.

You can change the contents of this variable. To suppress the message from the log, set the variable to NOLOG.

Note: For more information, see the &LOGREAD verb in the *Network Control Language Reference Guide*.

$LOG

Specifies a Mapped Data Object (MDO) that contains the message attributes. The MDO is mapped by the $MSG map.

You can use the &ASSIGN verb to query the MDO.

Note: For information about querying MDO components and additional variables, see the *Network Control Language Programming Guide*.

Example: Remove Messages from the NCL Log

The following shows an example procedure:

```
&CONTROL
-*-----------------------------------------------*
-* TO REMOVE IKJ56247I MESSAGES FROM THE NCL LOG. *
-*-----------------------------------------------*
&PARSE DELIM=' ' VARS=#LO$WORD* DATA=&#LO$RECORD
&IF .&#LO$WORD4 EQ .IKJ56247I &THEN +
        &#LO$RECORD = NOLOG
```

Enable the Log Exit

To enable the log exit

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups list appears.

2. Enter **U** beside the LOGFILES parameter group.

   The Customizer : Parameter Group panel for the LOGFILES parameter group appears.

3. Enter the name of your activity log exit in the Log Exit Name field.

4. Press F6 (Action).

   The changes are applied.

5. Press F3 (File).

   The changes are saved.

Online Logging Procedure

The default online logging procedure is $LOPROC. This procedure is designed to work with the online browsing procedure $LOBROW.

You can replace the $LOPROC and $LOBROW procedures with your own customized NCL procedures. Alternatively, you can write a customized log browsing procedure to present the supplied data files (from $LOPROC) in your own format.

Structure of Supplied Log Files

The supplied log files (NMLOG01, NMLOG02, and NMLOG03) have the following physical file structure:

- The record key has the following format:

  YYYYMMDDHHMMSSHS*nnnn*

  nnnn=1000 + (reset every 100th of a second) and key length=20 bytes

- A record has the following contents

  ORIGIN

  Contains the terminal name.

  REGION

  Contains the user ID.

  TEXT

  Contains the message text to display in the activity log.

  MSGATTR

  Contains the 2-byte color/highlight indicator. Colors are R=red, Y=yellow, W=white, B=blue, G=green, T=turquoise, or P=pink. Highlight values are R=reverse, B=blink, U=underscore, or N=none.

  ORIGTIM

  Contains the time at the remote domain.

  ORIGDMN

  Contains the name of the originating domain.

  ORIGSRC

  Contains the ID of the remote terminal.

Note: For more information, see the following references:

- The description of the &FILE OPEN verb in the *Network Control Language Reference Guide*.
- The *Network Control Language Programming Guide*.

How You Write Logging and Browsing Procedures

To write your own customized NCL procedure to replace $LOBROW, use the &FILE OPEN statement with FORMAT=DELIMITED.

You can store your log records in whatever file format you want. Your log browsing procedure must match this file format.

Note: For more information, see the descriptions of the following verbs in the *Network Control Language Reference Guide*:

- &LOGREAD
- &LOGCONT
- &LOGDEL

Implement Logging and Browsing Procedures

After you write your own browsing procedure or your own logging and browsing procedures, you implement them for use.

To implement your procedures

1. Enter **U** next to the LOGFILES parameter group in Customizer.
2. Update the relevant fields.
3. Press F6 (Action).

   Your procedures are used for logging and browsing.
4. Press F3 (File).

   Your changes to the parameter group are saved.

Hardcopy Activity Log

A region can have more than one hardcopy activity log, of which only one is open for logging.

Your region can be configured to perform logging to disk, tape, or hard copy. From one to nine logs can be specified by including the required number of DD statements in the execution JCL. Logging can be specified to wrap when the last log is full or is swapped.

To obtain the status of these logs, use the SHOW LOGS command.

Note: When logging to disk the following DCB attributes should be used:

```
DSORG=PS,RECFM=VBA,LRECL=137,BLKSIZE=15476
```

Format of Logged Information

Each entry recorded on the log has the following format:

```
12.04.23.12   SMITH      TERM54     +V NET,ACT,ID=NCP001
```

This entry consists of the following information:

- A time stamp in the format *hh.mm.ss.hs* (where *hh* is the hour, *mm* is the minute, *ss* is the second, and *hs* is the hundredth of a second)
- The user ID that entered the command or logged the message
- The terminal from which the command was entered or to which a message is sent
- The text of the message or command

Commands are highlighted with a plus sign (+) prefixed to the text to make it easier to distinguish commands from messages when browsing the log. If the command entered is an unsolicited VTAM command, it is highlighted and prefixed with an equals sign (=).

Format of Logged Timer-initiated Commands

Commands executed as the result of a timer-initiated command are prefixed by a plus sign, followed by the identity number of the timer command responsible. This identity number has the following format: #*nnnn*.

Example: Logged Timer-initiated Command

This example shows the log record of a command initiated by a timer:

```
15.00.00.01   NETOPER   CNTL01    +#0005 D BFRUSE
```

Format of Logged Commands Executed in Background Environments

Commands executed under the control of background environments are identified by the following keywords in the user ID field for the command text and any resulting messages:

BG-SYS

Background System Processor

BG-MON

Background Monitor

BG-LOG

Background Logger

Format of Logged Commands from NCL Procedure-dependent Environment

If a command is executed from an NCL procedure-dependent environment (&INTCMD), the node field on the log contains the NCL ID of the process issuing the command.

Format of Log After Time Change

If a time change causes the time to go backward, the activity log differentiates the records that overlap in time by adding a plus sign (+) after the time for the newer records. The feature is only available when you are viewing the log in the default or NORMAL format.

Format of the Hardcopy Log

The hardcopy log data set has the following format:

- A heading on each page—contains the day and date on which the log was created and the system identifier (NMID) of the originating region.
- A log identifier on the right side of the page. The log identifier is the ddname under which the log was created. This log identifier assists log collation after printing.
- 60 lines on each page—this format can be altered to suit your requirements using the SYSPARMS LOGPAGE operand.

  Note: For information about LOGPAGE, see the *Reference Guide*.

Swap the Hardcopy Log

Swapping the current log frees the log for immediate printing. Swapping the log is possible only when another unused log remains to which logging can continue. You can specify up to nine logs. Logs do not need to be consecutive.

To swap the log, enter the LOGSWAP command.

When a log is swapped, the log status, the requesting user ID, and the reason for the swap are recorded. You can display these details with the SHOW LOGS command.

Each of the logs is identified in the JCL member by the LOG*n* ddname. *n* is in the range one to nine.

Example: Log Name

This example defines the LOG4 ddname:

```
//LOG4 DD SYSOUT=A,FREE=CLOSE
```

Mixing of device types is valid. Inclusion of FREE=CLOSE prints the log when it is released by the LOGSWAP command.

Reuse of Hardcopy Log Data Sets

Wrapping lets you reuse a LOG data set when all of the available LOG data sets have been used.

The LOGWRAP SYSPARM determines whether log data set wrapping is allowed. You set the value of this SYSPARM in the Are Activity Logs to Wrap? field when you customize the LOGFILES parameter group in Customizer (**/PARMS**).

If you specify NO (the default) in the Are Activity Logs to Wrap? field, then wrapping is not permitted. When all the LOG data sets have been used due to successive LOGSWAP commands, the previous LOG data sets cannot be reused. After the last LOG data set is used, any further LOGSWAP commands are rejected.

If you specify YES in the Are Activity Logs to Wrap? field, log wrapping is allowed according to the following rules:

- If you direct your LOG data sets to SYSOUT, then, as each LOG*n* DD statement is used, the data set is unallocated because FREE=CLOSE. In this case, you can reissue an ALLOC command to reallocate another SYSOUT file under the same ddname. For example:

  ```
  ALLOC DD=LOG3 SYSOUT=A FREE=CLOSE
  ```

  This ddname is now available for use as another LOG data set. Subsequent LOGSWAP operations can now reuse this LOG data set rather than rejecting the command when the last LOG data set is used.
- If the LOG DD statements point to sequential data sets, log wrapping overwrites the earlier LOG data held in these data sets. Archive the existing data before allowing the wrap to occur.

Cross-Reference of Hardcopy Logs

To help operations staff to piece the full log together, certain information is recorded on the last and first lines of swapped LOG data sets.

The first line of a new log contains the reason for the swap, or the initiating user ID.

The last message printed on a swapped log is the ddname of the new log. Also printed at the start of the new log is the ddname or logical ID for the previous log.

I/O Errors on the Hardcopy Log

If an I/O error occurs on a log, the log is closed and the next available log is automatically swapped to, if one is available, and logging continues. This also applies to data set full conditions when logging to disk.

If the I/O error occurs on the last available log, a warning message is sent to all monitor terminals informing them that logging has ceased. The STATUS command also includes a warning message if logging is stopped. All log messages are passed to LOGPROC for analysis even if no log output is possible.

Write to the System Log

You can use the SYSPARMS SYSLOG operand to write all logged output or all VTAM PPO messages received to the system log.

To write all logged output to the system log also, enter the **SYSPARMS SYSLOG=YES** command.

To write all VTAM PPO messages to the system log also, enter the **SYSPARMS SYSLOG=PPO** command.

Note: For more information about the SYSPARMS SYSLOG operand, see the *Reference Guide*.

Chapter 24: Customizing WebCenter

This section contains the following topics:

Update WebCenter Parameters

The WEBCENTER parameter group contains information about the WebCenter interface. Through this parameter group, you can define the port number, the access URL, and so on.

Note: The SOCKETS parameter group must be enabled for WebCenter to function.

To access the WEBCENTER parameter group

1. Enter **/PARMS** at the prompt.

   The Customizer : Parameter Groups panel appears.

2. Enter **U** beside $NM WEBCENTER.

   The WEBCENTER Customizer Parameter Group panel appears.

3. Define the WebCenter Web interface and press F6 (Action).

   Note: For more information about the fields, press F1 (Help).

   The changes are applied.

4. Press F3 (Exit).

   The changes are saved.

WebCenter SSL Security

Secure Sockets Layer (SSL) for WebCenter provides support for encrypted conversations between the WebCenter Web server and client Web browsers.

If you want to enable this facility, you must obtain a certificate from a Certificate Authority and define it using the WEBCENTER parameter group.

Note: For more information, see the *Security Guide*.

How You Control Access to WebCenter Menu Options

You can control access to WebCenter menu options programmatically by using the variables in the CC2DEXEC($W3MH01X) WebCenter menu user exit.

The exit contains a menu control variable for each menu option in WebCenter. These variables are initially set to TRUE. To prevent access to a particular option, change the value of the control variable for that option to FALSE.

The exit is called when a user logs in to WebCenter. You can update this exit to suit your requirements.

Important! If modifications are required, copy the distributed member to the TESTEXEC data set for the region for modification.

To control access to a menu option for a user, use the &SECCALL NCL verb to query the user's access as defined in UAMS. Then, set the appropriate variables to TRUE or FALSE. The exit contains some sample code.

Note: For more information about the verb, see the *Network Control Language Reference Guide*.

To control access to a menu option for all users, update the value of its control variable but do not perform an &SECCALL query.

Changes to this exit are not dynamic. For changes to the exit to take effect, the user must log in to WebCenter with a new advanced program-to-program communications (APPC) connection. That is, if a user is logged in when the changes occur, the changes are not evident until they log out and log in again.

Important! Preventing access to a menu option does not mean preventing access to the underlying function. What it does is to prevent users from using the menu option to access the function.

Log On to WebCenter

You can use Internet Explorer or Firefox to access WebCenter.

Important! If you use an upgraded product for the first time, delete the temporary internet files or clear the cache of your browser before you log in to WebCenter.

To log on to WebCenter

1. Start your web browser, and enter the web access URL in the Address text box.

   The WebCenter logon window appears.

   Note: The web access URL is defined when you enable the SOCKETS parameter group. This task can be performed during installation or afterwards. You can find the value on the primary menu of the 3270 interface.

2. Enter your user ID and password, and click the Log In button.

   The WebCenter menu appears.

Chapter 25: Implementing Print Services

This section contains the following topics:

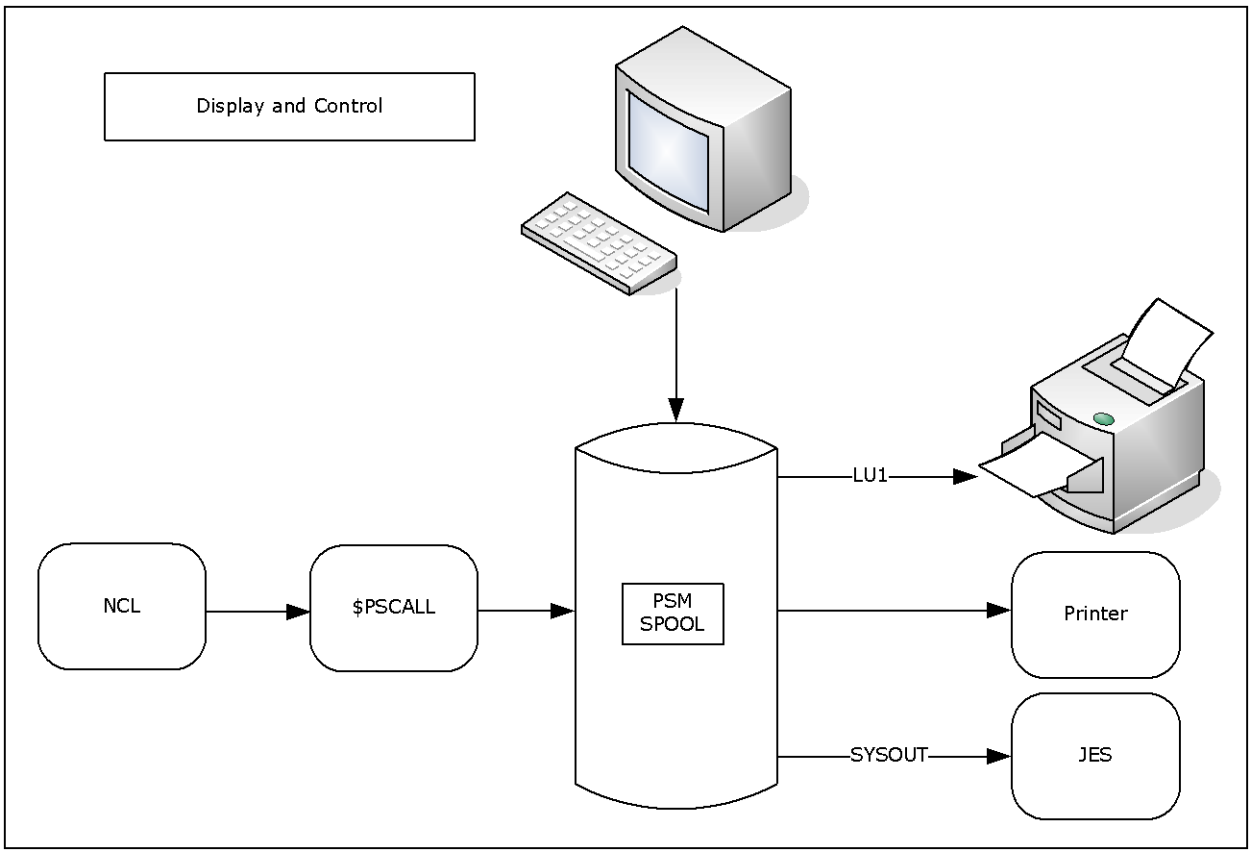
Print Services Manager

Print Services Manager (PSM) allows you to specify the format of a print request and on which printer it is printed. Print requests can be viewed online before or after printing and can be redirected to files rather than printers.

PSM provides the following features, which can be customized to suit your requirements:

- Printer definition facilities
- Form definition maintenance
- Setup definition maintenance
- Default printer assignment maintenance
- Alias printer name definitions
- Banner page customization on output
- Spooled print request browsing, retention, and redirection to a different printer
- Integration with NCL-based components

The following illustration shows the different ways that PSM can be used to control printing requirements.



Access PSM

The customizable functions of PSM are accessed from the PSM : Primary Menu.

To access PSM, enter **/PSM** at the prompt.

Note: You can also access PSM directly by invoking the $PSCALL NCL procedure from OCS or an installation written NCL procedure. The PSM NCL interface is described in the *Network Control Language Reference Guide*.

Add a Printer Definition

A printer definition defines where, how, and on what paper output is printed. A printer definition is required for each printer at which output is printed.

To add a printer definition

1. Enter **/PSMPRTR** at the prompt.

   The PSM : Printer Definition List appears.

2. Press F4 (Add).

   The PSM : Printer Definition panel appears.

3. Complete the fields, as required.

   Note: For information about the fields, press F1 (Help).

4. Press F3 (File).

   The definition is saved.

List Printer Definitions

You can display a list of all the printer definitions defined for your region. This lets you browse and perform maintenance on the listed definitions.

To list all printer definitions, enter **/PSMPRTR** at the prompt.

Add a Form Definition

A form definition is required for each type of paper on which output is printed. The Form Definition Menu is used to set up and administer these form definitions.

To add a form definition

1. Enter **/PSMFORM** at the prompt.

   The PSM : Form Definition List appears.

2. Press F4 (Add).

   The PSM : Form Definition panel appears.

3. Complete the fields and press F3 (File).

   The form definition is saved.

   Note: For information about the fields, press F1 (Help).

List Form Definitions

You can list all of the form definitions defined for your region and then browse and perform maintenance on them.

To list all form definitions, enter **/PSMFORM** at the prompt.

Add Control Characters

Control characters are sent to a printer before or after (or both) the output is printed. They are defined in setup definitions.

To add control characters

1. Enter **/PSMSET** at the prompt.

   The PSM : Setup Definition List appears.

2. Press F4 (Add).

   The PSM : Setup Definition panel appears. To access the second panel of the setup definition, press F8 (Forward).

   Complete the fields, as required.

   Note: For information about the fields, press F1 (Help).

3. Press F3 (File).

   The setup definition is saved.

List Control Characters

You can display a list of all the setup definitions defined for your region. This list lets you browse and perform maintenance on the listed definitions.

To list control characters, enter **/PSMSET** at the prompt.

Add a Default Printer for a User ID

Each user ID in your region can be assigned a default printer. Default printer assignments let you define the printer to which output is sent whenever a user ID does not specify a printer.

To add a default printer for a user ID

1. Enter **/PSMDFTP** at the prompt.

   The PSM : Default Printer Assignment List appears.

2. Press F4 (Add).

   The PSM : Default Printer Assignment panel appears.

3. Complete the following fields:

   User ID

   Specifies the User ID of the user to whom the printer is assigned a default.

   Printer Name

   Specifies the name of the printer to which this user's printing is sent.

   Press F3 (File).

   The default printer assignment is saved.

List Default Printers

You can display a list of all the default printer assignments defined for each user ID. This list lets you browse and perform maintenance on the listed definitions.

To list default printers, enter **/PSMDFTP** at the prompt.

Clear the Printer Spool

Print requests are retained on the print spool if an error occurs during printing or if HELD is specified on the PSM : Print Request panel. The PSM clear spool panel is used to clear print requests from the print queue.

Note: This function is available to authorized users only.

To clear the print spool

1. Enter **/PSMADMN** at the prompt.

   The PSM : Administration Menu appears.

2. Enter **CS** at the prompt.

   The PSM : Clear Spool panel appears.

3. Complete the following field:

   Date

   Specifies that all print requests added to the spool before or on this date are deleted.

   Press F6 (Action).

   The print requests are deleted.

Exits to Send Print Requests to a Data Set

Two printer exit procedures are distributed with your product. Each writes the output for a print request to a data set. The procedure $PSDS81X can be customized to specific site requirements. The procedure $PSDS81Z offers the same functionality with improved performance, but cannot be customized. The target data sets for both procedures can be sequential or partitioned.

Parameters that control the operation of the exit are defined in the Exit Data portion of the printer definition. Procedures that pass data to PSM for printing can override the exit data specified in the PSM printer definition.

The procedures use the parameters contained in the exit data to do the following:

- Determine the target data set
- Determine how to process a data line with a skip amount of zero
- Set the length of the lines print

How the Procedures Process a Print Request

The procedures read each line of print data and write it directly to the nominated data set. Each print line is analyzed according to skip control before processing. This continues until all lines of data for the print request have been received from PSM and written to the nominated data set.

$PSDS81X and $PSDS81Z Parameters

The $PSDS81X and $PSDS81Z exits have the following keyword parameters:

```
  DSN=datasetname
[ DISP={ SHR | OLD | NEW | MOD } ]
[ LRECL={ n | 80 } ]
[ SKIP0={ NEWLINE | DISCARD | DESTRUCTIVE |
          NONDESTRUCTIVE } ]
[ CYL= pri [,sec] [,dir] ]
[ TRK={ pri [,sec] [,dir] | 15,5 } ]
[ BLKSZ= n ]
[ STORC= storclas ]
[ MGMTC= mgmtclas ]
[ DATAC= dataclas ]
[ VOL= volser ]
[ UNIT={ unit | SYSALLDA} ]
[ RECFM={ F | FB | V | VB } ]
```

DSN=*datasetname*

Specifies the target data set name. If the data set is partitioned, the member name must be included or the data set is corrupted.

You can use the following symbolics in the *datasetname* parameter:

- &DAY is the day of the week (for example, MON).
- &YY is the two-digit representation of the year (for example, 11).
- &YYYY is the four-digit representation of the year (for example, 2011).
- &MM is the two-digit representation of the month (for example, 02).
- &MON is the three-character representation of the month (for example, JAN and FEB).
- &DD is the day of the month.
- &HHMMSS is the time.
- &HH is the hour.
- &MIN is the minute.
- &JOBID is the job ID.
- &JOBNAME is the job name.
- &NMID is the region ID.
- &NMDID is the region domain ID (DID).
- &GRPNAME is the sysplex name.
- &SYSID is the system ID.
- &SYSNAME is the system name.
- &USERID is the requesting user ID.

Symbolics are delimited by a period (.) or another symbolic (that is, &YY&MM. is the same as &YY.&MM.). Symbolics are also allowed in a member name.

Example:

```
DSN=NM.&SYSID..&USERID..D&YY&MM&DD..T&HHMMSS..DATA
```

For example, this specification can resolve to the following data set name:

```
DSN=NM.SYSA.MYUSER.D040915.T144505.DATA
```

DISP={ SHR | OLD | NEW | MOD }

Specifies the disposition of the output data set.

- SHR specifies shared use of the data set.
- OLD specifies exclusive use of the data set.
- NEW allocates a new data set.
- MOD appends the output in the file.

Default: SHR

LRECL={ *n* | 80 }

Specifies the output record length.

Limits: 1 through 250

Default: 80

SKIP0={ NEWLINE | DISCARD | DESTRUCTIVE | NONDESTRUCTIVE }

Specifies how to process a data line with a skip amount of zero.

- NEWLINE creates a line of data.
- DISCARD discards the line of data.
- DESTRUCTIVE causes the data to replace the existing data line.
- NONDESTRUCTIVE overlays the data on the existing data line, but only where blanks were present on the existing data line. No existing non-blank characters are modified.

Note: The procedures ignore the following PSM print options: NEWPAGE and USCORE.

Default: NEWLINE

The following additional parameters are applicable when DISP=NEW is specified:

CYL=*pri,sec,dir*

Specifies the primary and secondary space allocation values in cylinders. If a partitioned data set is used, the parameter specifies the number of directory blocks.

TRK=*pri,sec,dir*

Specifies the primary and secondary space allocation values in tracks. If a partitioned data set is used, the parameter specifies the number of directory blocks.

Default: TRK=15,5

BLKSZ=*blocksize*

Specifies the block size.

STORC=*storclas*

Specifies the storage class.

MGMTC=*mgmtclas*

Specifies the management class.

DATAC=*dataclas*

Specifies the data class.

VOL=*volser*

Specifies the volume serial number.

UNIT= { *unit* | SYSALLDA }

Specifies the unit.

Default: SYSALLDA if volser is specified

RECFM= { F | FB | V | VB }

Specifies the record format.

Default: FB

Printer Exit Definition Example

This example directs the output for a PSM print request, assigned to the printer named DSEXIT, to the member TEST1 in the data set PROD.PSM.DATA. The record length of this data set is 80. Overlay lines in the data are removed.

```
PROD1--------------------- PSM : Printer Definition ---------------------------
Command ===>                                                  Function=BROWSE

  Printer Name ... DSEXIT
  Type ........... EXIT              (JES, VTAM, ALIAS, EXIT)
  Description .... Print to a data set
  Lower Case? .... YES               (Yes or No)
  Line Limit ..... 0                 (0 to 999999)
  Form Name .....+ FORM0
ALIAS Printer
  Real Name .....+                   (Real printer name)
JES Printer
  Destination ....                   (destid.userid)
  Output Class ...                   (A to Z, 0 to 9)
VTAM Printer
  LU Name ........
  Logmode ........
EXIT
  Exit Name ...... $PSDS81Z
  Exit Data ...... DSN=PROD.PSM.DATA(TEST1) LRECL=80
                   SKIP0=DISCARD
```

Note: Previous references to parameters WKVOL, CYL, and LIST in the exit data are no longer required. Remove them from the printer definition before using $PSDS81Z or $PSDS81X, or the print request fails.

Print-to-Email

The $PSEMAIL printer definition lets you email the output of a printing request. The request can be either an attachment or in the body of the email. When the output is sent as an attachment, the email uses the PS8803 message as its body and the PS8804 message as its salutation:

`Data attached for` *`email_subject`*

`Yours,`
`user_name`

user_name

Displays the sender name defined in UAMS.

You can maintain these messages from the Message Definition List panel. The shortcut to the panel is /CASMSG.

Note: For information about how to maintain messages, see the *Managed Object Development Services Guide*.

Chapter 26: Implementing a CA SOLVE:Central Automatic Problem Recording Environment

This section contains the following topics:

How Automatic Problem Recording Works

Your region can add problem records to the SOLVE:Problem application in CA SOLVE:Central.

The $RMPB06S and $RMPB07S NCL procedures enable a region to record problems automatically in the SOLVE:Problem application.

The region uses the $RMPB06S procedure to raise problem tickets and starts the $RMPB07S procedure in the CA SOLVE:Central region to record them in the SOLVE:Problem application.

Set Up SOLVE:Problem

You set up the SOLVE:Problem application for it to receive and record problems.

To set up SOLVE:Problem

- Copy the $RMPB07S NCL procedure from the CC2DEXEC data set to the NCL procedures library (typically TESTEXEC) in the region in which the SOLVE:Problem application is running.

  dsnpref

  Is the data set prefix used during the installation of CA SOLVE:Operations Automation.

  vv

  Is the release code.

- Define the following user IDs to the SOLVE:Problem region:
  - *xxxx*BSYS BSYS background user IDs, where *xxxx* is the domain ID of the CA SOLVE:Operations Automation region from which you want to receive problem tickets
  - IDs of the users who can raise problem tickets manually from the alert monitor in the CA SOLVE:Operations Automation region
- Define a link, in the SOLVE:Problem region, to each region from which you want to receive problem tickets:

  `DEFLINK TYPE=APPC LUNAME=`*`acb_name`* `LINK=`*`link_name`*

  acb_name

  Specifies the ACB name of the CA SOLVE:Operations Automation region.

  link_name

  Defines a name that identifies the link.

$RMPB06S Procedure—Exit to the SOLVE:Problem Application

The $RMPB06S NCL procedure is an exit to the SOLVE:Problem application.

Syntax

This procedure has the following format:

```
$RMPB06S ACBNAME=acb_name
```

acb_name

Specifies the link to the region in which the SOLVE:Problem application is running. You can issue the SHOW PARMS command from that region to determine its ACB name. The PRI parameter in its RUNSYSIN startup JCL member specifies the name.

Return Codes

A zero return code indicates that the NCL procedure was successful. A nonzero return code indicates that the NCL procedure failed.

If an error occurs during the execution of the $RMPB06S NCL procedure, the error is logged as a severity 1 message. (The message appears in red in a transient log). The message contains one of the following return codes:

4

Indicates failed execution.

8

Indicates syntax error.

Example: PROBSOLV Process

CA SOLVE:Operations Automation provides a sample process, PROBSOLV, which invokes this procedure. The process uses the ACB parameter to pass *acb_name* to the procedure.

Appendix A: CICSCMD Command

This section contains the following topics:

Overview

The CICSCMD command lets you issue CICS commands to manage your CICS resources.

You can issue a CICS command from various places, including the following locations:

- COMMAND macro
- Command Entry panel
- Message monitor

CICSCMD INQ ADABAS Command—Inquire ADABAS Database Availability

The CICSCMD INQ ADABAS inquires about the availability of an ADABAS database. It returns the RCIQ02 message that indicates the status of the selected database.

This command has the following format:

```
CICSCMD cics_region_name INQ ADABAS[(database_file_number)]
```

cics_region_name

Defines the CICS region for which the availability information is required.

database_file_number

(Optional) Defines the file number of the ADABAS database to which the inquiry applies. If no file number is specified, the inquiry applies to the default database.

Example:

```
CICSCMD CICSA INQ ADABAS(100)
```

CICSCMD INQ CONNECTION—Inquire Connection Status

The CICSCMD INQ CONNECTION command inquires about the status of CICS-defined connections. It returns the RCIW08 messages that indicate the status of the selected connections. The returned information is similar to that returned by the CICS CEMT INQUIRE CONNECTION transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ CONNECTION[(connection_name_mask)]
```

cics_region_name

Defines the CICS region from which the information is required.

connection_name_mask

(Optional) Specifies the connections to which the inquiry applies. The value can be either the name or a mask. If you use a mask, it must be of the following form: *connection_name_prefix**. If it is not specified, the inquiry applies to all connections defined in the CICS region.

Example:

```
CICSCMD CICSA INQ CONN

CICSCMD CICSA INQ CONN(CICB)

CICSCMD CICSA INQ CONN(CIC*)
```

CICSCMD INQ DB2—Inquire CICS-DB2 Connection Status

The CICSCMD INQ DB2 command inquires about the status of the CICS-DB2 connection. It returns the RCIQ05 or RCIQ06 message that indicates the status of the CICS-DB2 connection.

This command has the following format:

```
CICSCMD cics_region_name INQ DB2
```

cics_region_name

Defines the CICS region from which the information is required.

Example:

```
CICSCMD CICSA INQ DB2
```

CICSCMD INQ FILE—Inquire File Status

The CICSCMD INQ FILE command inquires about the status of CICS files. It returns the RCIW05 messages that indicate the status of the selected files. The returned information is similar to that returned by the CICS CEMT INQUIRE FILE transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ FILE[(file_name_mask)]
```

cics_region_name

Defines the CICS region from which the information is required.

file_name_mask

(Optional) Specifies the files to which the inquiry applies. The value can be either the name or a mask. If you use a mask, it must be of the following form: *file_name_prefix**. If it is not specified, the inquiry applies to all files defined in the CICS region.

Example:

```
CICSCMD CICSA INQ FILE

CICSCMD CICSA INQ FILE(CUSTOMER)

CICSCMD CICSA INQ FILE(CUST*)
```

CICSCMD INQ IRC—Inquire IRC Status

The CICSCMD INQ IRC command inquires about the status of the interregion communication (IRC) facility. It returns the RCIW12 message that indicates the status of IRC. The returned information is similar to that returned by the CICS CEMT INQUIRE IRC transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ IRC
```

cics_region_name

Defines the CICS region from which the information is required.

Example:

```
CICSCMD CICSA INQ IRC
```

CICSCMD INQ PSBNAME—Inquire PSB Availability

The CICSCMD INQ PSBNAME command inquires about the availability of PSB resources. It returns the RCIQ03 or RCIQ04 message that indicates the status of the DL/I database resources described by the selected PSB.

This command has the following format:

```
CICSCMD cics_region_name INQ PSBNAME(psb_name)
```

cics_region_name

Defines the CICS region from which the information is required.

psb_name

Defines the PSB to which the inquiry applies.

Example:

```
CICSCMD CICSA INQ PSBNAME(CICSPGM1)
```

CICSCMD INQ TASK—Inquire Task Status

The CICSCMD INQ TASK command inquires about the status of CICS tasks. It returns the RCIW11 messages that indicate the status of all tasks in the CICS region. The returned information is similar to that returned by the CICS CEMT INQUIRE TASK transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ TASK
```

cics_region_name

Defines the CICS region from which the information is required.

Example:

```
CICSCMD CICSA INQ TASK
```

CICSCMD INQ TERMINAL—Inquire Terminal Status

The CICSCMD INQ TERMINAL command inquires about the status of CICS terminals. It returns the RCIW10 messages that indicate the status of the selected terminals. The returned information is similar to that returned by the CICS CEMT INQUIRE TERMINAL transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ TERMINAL[(terminal_name_mask)]
```

cics_region_name

Defines the CICS region from which the information is required.

terminal_name_mask

(Optional) Specifies the terminals to which the inquiry applies. The value can be either the name or a mask. If you use a mask, it must be of the following form: *terminal_name_prefix**. If it is not specified, the inquiry applies to all terminals defined in the CICS region.

Example:

```
CICSCMD CICSA INQ TERM

CICSCMD CICSA INQ TERM(S201)

CICSCMD CICSA INQ TERM(S*)
```

CICSCMD INQ TRANSACTION—Inquire Transaction Status

The CICSCMD INQ TRANSACTION command inquires about the status of CICS transactions. It returns the RCIW07 messages that indicate the status of the selected transactions. The returned information is similar to that returned by the CICS CEMT INQUIRE TRANSACTION transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ TRANSACTION[(transaction_name_mask)]
```

cics_region_name

Defines the CICS region from which the information is required.

transaction_name_mask

(Optional) Specifies the transactions to which the inquiry applies. The value can be either the name or a mask. If you use a mask, it must be of the following form: *transaction_name_prefix**. If it is not specified, the inquiry applies to all transactions defined in the CICS region.

Example:

```
CICSCMD CICSA INQ TRANS

CICSCMD CICSA INQ TRANS(ACNT)

CICSCMD CICSA INQ TRANS(AC*)
```

CICSCMD INQ VTAM—Inquire ACB Status

The CICSCMD INQ VTAM command inquires about the status of the CICS VTAM ACB. It returns the RCIW09 message that indicates the status of the CICS-VTAM connection. The returned information is similar to that returned by the CICS CEMT INQUIRE VTAM transaction.

This command has the following format:

```
CICSCMD cics_region_name INQ VTAM
```

cics_region_name

Defines the CICS region from which the information is required.

Example:

```
CICSCMD CICSA INQ VTAM
```

CICSCMD PERFORM SHUTDOWN—Shut Down Region

The CICSCMD PERFORM SHUTDOWN command shuts down a CICS region. The RCIW17 message acknowledges the shutdown request. The function is similar to that provided by the CICS CEMT PERFORM SHUTDOWN transaction.

This command has the following format:

```
CICSCMDcics_region_name PERFORM SHUTDOWN [IMMEDIATE | TAKEOVER] [DUMP]
```

cics_region_name

Defines the CICS region to be shut down.

IMMEDIATE | TAKEOVER

(Optional) Specifies the shutdown option. If you do not specify any value, the region (and the alternate CICS region if it exists) is shut down normally.

IMMEDIATE

Shuts down the region immediately.

TAKEOVER

Shuts down the region to let the alternate CICS region take over.

DUMP

(Optional) Produces a storage dump when shutdown completes.

Example:

```
CICSCMD CICSA PERFORM SHUTDOWN

CICSCMD CICSA PERFORM SHUTDOWN IMMEDIATE DUMP
```

CICSCMD SET CONNECTION—Set Connection Status

The CICSCMD SET CONNECTION command controls the status of CICS-defined connections. The RCIW06 or RCIW20 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET CONNECTION transaction.

This command has the following format:

```
CICSCMD cics_region_name SET CONNECTION(connection_name_mask) {INS | OUT} {ACQ | REL}
```

cics_region_name

Defines the CICS region that owns the connections whose status is to be set.

connection_name_mask

Specifies the connections whose status is to be set. The value can be either the name or a mask. If you use a mask, it must be of the following form: *connection_name_prefix**.

INS | OUT

Specifies whether the CICS region can receive and send data.

INS

Puts the connections into service. The CICS region can receive and send data.

For APPC connections, INS enables them to be established subsequently.

OUT

Puts the connections out of service. The CICS region can neither receive nor send data.

For APPC connections, OUT is valid only for links with no acquired sessions.

ACQ | REL

(APPC only) Specifies whether to acquire or release the sessions on the links.

ACQ

Acquires sessions with the CICS regions at the other end of the links that are in service.

REL

Releases the sessions with the CICS regions at the other end of the links.

Example:

```
CICSCMD CICSA SET CONN(CICB) INS

CICSCMD CICSA SET CONN(CICB) ACQ
```

CICSCMD SET FILE—Set File Status

The CICSCMD SET FILE command controls the status of CICS files. The RCIW06 or RCIW20 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET FILE transaction.

This command has the following format:

```
CICSCMD cics_region_name SET FILE(file_name_mask) {ENA | DIS} {OPEN | CLOSED}
```

cics_region_name

Defines the CICS region that owns the files whose status is to be set.

file_name_mask

Specifies the files whose status is to be set. The value can be either the name or a mask. If you use a mask, it must be of the following form: *file_name_prefix**.

ENA | DIS

Specifies whether CICS transactions can access the files.

ENA

Enables the files. Transactions can access an enabled file.

DIS

Disables the files. Transactions cannot access a disabled file. However, transactions that are already using the file can continue using the file.

OPEN | CLOSED

Specifies whether to open or close the files.

OPEN

Opens the files for data access. If the UNENABLED file attribute has been set by a previous CLOSED request, the OPEN request enables the file.

CLOSED

Closes the files to data access. If the file was enabled, the CLOSED request sets the UNENABLED attribute.

Example:

```
CICSCMD CICSA SET FILE(CUSTOMER) CLOSED

CICSCMD CICSA SET FILE(CUSTOMER) OPEN
```

CICSCMD SET IRC—Set IRC Status

The CICSCMD SET IRC command controls the status of IRC. The RCIW06 or RCIW20 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET IRC transaction.

This command has the following format:

```
CICSCMD cics_region_name SET IRC {OPEN | CLOSED | IMMCLOSE}
```

cics_region_name

Defines the CICS region for which you want to affect the IRC status.

OPEN | CLOSED | IMMCLOSE

Specifies whether to initialize or terminate IRC.

OPEN

Initializes IRC for connections to be established.

CLOSED

Terminates IRC after all the tasks that use IRC sessions have ended.

IMMCLOSE

Terminates IRC after all the tasks that use IRC sessions have ended or abended. The IMMCLOSE request causes an existing task to abend when it tries to use an IRC session.

Example:

```
CICSCMD CICSA SET IRC CLOSED

CICSCMD CICSA SET IRC OPEN
```

CICSCMD SET TASK—Terminate Task

The CICSCMD SET TASK command terminates a CICS task. The RCIW15 or RCIW14 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET TASK transaction.

This command has the following format:

```
CICSCMD cics_region_name SET TASK(task_number) {PUR | FORCE}
```

cics_region_name

Defines the CICS region that owns the task to be terminated.

task_number

Defines the task to be terminated.

PUR | FORCE

Terminates the task.

PUR

Terminates the task only if region and data integrity can be maintained.

FORCE

Terminates the task irrespective of whether region and data integrity can be maintained.

Example:

```
CICSCMD CICSA SET TASK(28) PUR

CICSCMD CICSA SET TASK(128) FORCE
```

CICSCMD SET TERMINAL—Set Terminal Status

The CICSCMD SET TERMINAL command controls the status of CICS terminals. The RCIW06 or RCIW20 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET TERMINAL transaction.

This command has the following format:

CICSCMD*cics_region_name* SET **TER**MINAL(*terminal_name_mask*) {INS | OUT} {ACQ | REL}

cics_region_name

Defines the CICS region that owns the terminals whose status is to be set.

terminal_name_mask

Specifies the terminals whose status is to be set. The value can be either the name or a mask. If you use a mask, it must be of the following form: *terminal_name_prefix**.

INS | OUT

Specifies whether CICS transactions can use the terminals.

INS

Puts the terminals into service. CICS transactions can use the terminals.

For VTAM terminals, INS enables them to be acquired subsequently.

OUT

Puts the terminals out of service. CICS transactions can no longer use the terminals.

For VTAM terminals, OUT causes them to be released.

ACQ | REL

(VTAM only) Specifies whether to acquire or release the terminals.

ACQ

Acquires sessions with the logical units represented by the terminals.

REL

Releases the sessions with the logical units represented by the terminals.

Example:

```
CICSCMD CICSA SET TERM(S201) INS

CICSCMD CICSA SET TERM(S201) ACQ
```

CICSCMD SET TRANSACTION—Set Transaction Status

The CICSCMD SET TRANSACTION command controls the status of CICS transactions. The RCIW06 or RCIW20 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET TRANSACTION transaction.

This command has the following format:

```
CICSCMD cics_region_name SET TRANSACTION(transaction_name_mask) {ENA | DIS}
```

cics_region_name

Defines the CICS region that owns the transactions whose status is to be set.

transaction_name_mask

Specifies the transactions whose status is to be set. The value can be either the name or a mask. If you use a mask, it must be of the following form: *transaction_name_prefix**.

ENA | DIS

Specifies whether to enable or disable the transactions.

ENA

Enables the transactions. The transactions are available for use.

DIS

Disables the transactions. The transactions are not available for use. However, transaction instances that are already started can continue until completion.

Example:

```
CICSCMD CICSA SET TRANS(ACNT) ENA

CICSCMD CICSA SET TRANS(ACNT) DIS
```

CICSCMD SET VTAM—Set ACB Status

The CICSCMD SET VTAM command controls the status of the CICS VTAM ACB. The command sets the status of the CICS-VTAM connection. The RCIW06 or RCIW20 message indicates the success or failure of the operation. The function is similar to that provided by the CICS CEMT SET VTAM transaction.

This command has the following format:

```
CICSCMD cics_region_name SET VTAM {OPEN | CLOSED | IMMCLOSE | FORCECLOSE}
```

cics_region_name

Defines the CICS region for which you want to affect the IRC status.

OPEN | CLOSED | IMMCLOSE | FORCECLOSE

Specifies whether to establish or terminate the CICS-VTAM connection.

OPEN

Opens the CICS VTAM ACB to establish the connection.

CLOSED

Waits for transaction tasks that use VTAM-connected terminals to complete before releasing the terminals, and then closes the CICS VTAM ACB to terminate the connection.

IMMCLOSE

Cancels transaction tasks that use VTAM-connected terminals, closes the terminal sessions, and then closes the CICS VTAM ACB to terminate the connection.

FORCECLOSE

Closes the CICS VTAM ACB to terminate the connection, causing all terminal sessions to terminate immediately.

Example:

```
CICSCMD CICSA SET VTAM CLOSED

CICSCMD CICSA SET VTAM OPEN
```

Appendix B: NCL Exits

The NCL exits provided by Automation Services are standardized to ensure that the following are constant:

- The storage of exit parameters in the knowledge base
- The substitution of variables in parameter strings for exits
- The passing of variables to exit code

This section contains the following topics:

NCL Exit Procedures

An exit is a piece of user-written code that is executed at a predefined point. Numerous exit points are supplied by your product, to enable you to perform site-specific functions as required. For example, wherever you can specify a process in Automation Services, you can replace the process name with the name $NCL and specify the name of an NCL exit to be called at that point.

Note: For more information about $NCL, see the *Reference Guide*.

Standardized Structure

For ease of maintenance and debugging, an NCL exit should have the following structure (as discussed in this appendix):

- A commented introductory section
- A procedure that validates and interprets the parameters passed to the exit and converts them to variables
- The core processing code that performs the main function or functions of the exit
- A final piece of code that sets a return code and passes control back to the caller

A sample NCL exit procedure, $RMPB06S, is supplied with Automation Services for your reference. This procedure creates a SOLVE:Problem record.

Introduction Section

The commented introductory section should contain text that describes the following:

- The purpose of the exit
- The parameters passed to the exit
- The return codes set by the exit

Types of Parameters

There are two types of parameters available to an NCL exit when it receives control (that is, starts executing):

- Parameters passed to the exit as NCL variables
- Parameters passed directly to the exit (that is, as a parameter string)

Parameters Passed as NCL Variables

These parameters are created by the exit controller.

Different variables are passed to an exit, depending on the type of exit. For example, the variable &ZRMDBNAME (which contains a resource name) is passed to a state change exit.

Note: For information about the variables available to an NCL exit procedure, see the *Reference Guide*.

Parameters Passed Directly to the Exit Procedure

These parameters are in the form of a user-controlled parameter string that is stored, with the exit name, in the knowledge base. Parameters passed directly to the exit, such as the ones shown in the following example, are stored in the system variable &ALLPARMS:

```
CUSTACT=NOTIFY USER=FRED TEXT="Fred, the problem's fixed!"
```

Parameters passed directly to the exit can be in the keyword or the positional format. However, if $NCL calls the exit, positional parameters are not accepted. The positional format is not recommended.

The following code fragment processes the parameters stored in the system variable &ALLPARMS and sets a return code. This code fragment can only process parameters that are in the *keyword* format:

```
&SETVARS PREFIX=PARM ERROR=CONTINUE DATA=&ALLPARMS
&IF &RETCODE NE 0 &THEN +
   &DO
      &SYSMSG = &STR &0-001 PARM ERROR MSG=&SYSMSG
      &RETURN &SYSMSG
   &DOEND
```

If the parameters are not in the keyword format, the system variable &RETCODE is set to eight. An error message loaded into the system variable &SYSMSG. Using information from the &SETVARS statement, the &DO group of statements set an error message that:

- Identifies the exit procedure
- Supplies the identification number of the error within the exit
- Identifies the error condition

Keyword Format

The standard keyword has the following format:

keyword=value

The keyword is a one to eight characters long, alphanumeric string. If the value contains blanks, then quotes are required, according to normal quoting rules. For example, a value of ACT IMM is specified as follows:

```
KEYWORD='ACT IMM'
```

A value of NOT ACT'D is specified in one of the following ways:

```
KEYWORD='NOT ACT''D'
KEYWORD="NOT ACT'D"
```

If the initial &SETVARS statement is successful, keywords are converted into variables which have the format: PARM*keyword.*

Example: Keyword Variables

This example passes the following parameters:

```
CUSTACT=NOTIFY USER=FRED TEXT="Fred, the problem's fixed!"
```

These parameters create the following variables:

- PARMCUSTACT, containing the value: NOTIFY
- PARMUSER, containing the value: FRED
- PARMTEXT, containing the value: Fred, the problem's fixed!

Positional Parameter Format

Positional parameters are named according to their position in a parameter string, such as &1, &2, and so on. A space separates these parameters.

Example: Keyword Variables

This example passes the following parameters:

```
CUSTACT=NOTIFY USER=FRED TEXT="Fred, the problem's fixed!"
```

The positional parameters create the following variables:

- &1, containing the value: CUSTACT=NOTIFY
- &2, containing the value: USER=FRED
- &3, containing the value: TEXT=“Fred,
- &4, containing the value: the
- &5, containing the value: problem’s
- &6, containing the value: fixed!”

Important! Using the positional parameter format with variables in a parameter string can produce unexpected results. For example, if the variable happens to have a null value, the remaining variables all shift across one argument. That is, if positional parameter &5 contains a variable value and that variable value is blank, &6 becomes positional parameter &5.

Main Processing Section

This section of the exit contains the processing code that performs the main exit functions (that is, the functions for which reason the exit has been written).

How the Procedure Exits Back to the Caller

After successful processing, the procedure sets the variable &RETCODE to zero, leaves &SYSMSG empty, and exits to the caller.

If processing is not successful, the variable &RETCODE must be set to eight, and a meaningful error message loaded into the variable &SYSMSG.

Appendix C: Sysplex Support

This section contains the following topics:

Clone Regions

The z/OS system enables you to clone systems by sharing data set members. This product uses this feature to enable you to clone regions on different systems in a sysplex environment.

To clone regions on different systems, update the RUNSYSIN member for sharing by the regions as follows:

1. Update appropriate parameter values and data set names to use defined system symbols to enable the member to be shared between regions on different systems. Useful system symbols include &SYSCLONE, &SYSNAME, and &SYSPLEX. You can use the VAR*xxx* control statement to set up your own variables.
2. Add **SUBS=YES** before the statements in which system symbols or defined variables are used, to enable substitution.

Register a Region with the Sysplex Automatic Restart Manager

The sysplex automatic restart manager (ARM) restarts a registered region automatically if that region fails.

To register a region with ARM, specify the following parameters in a PPREF statement in the RUNSYSIN member:

XOPT=(...,{NOARM|ARM},...)

Specifies whether the region should (ARM) or should not (NOARM) register with ARM.

ARMNAME=*element_name*

Specifies the name used to register the region with ARM.

Default: SVM_*acb_name*, where *acb_name* is the value of the PRI parameter.

The type of the registered element is SOLVEMS.

Restart Status Messages

Restart status messages provide information about the restart status of resources that are registered with ARM.

An SSI region can use the event notification facility (ENF) to listen for ARM events and generate WTO messages for these events. All regions on the same system receive these generated messages.

You can use these messages, and system messages IXC392I and IXC807I through IXC813I, to track the status of the regions that are under the control of ARM.

Enable the Generation of the Restart Status Messages

To generate the restart status messages, an SSI region must be authorized to listen to ARM ENF events. Set up only one SSI region per system to listen to these events.

To set up an SSI region as an ENF listener, add the following parameters in the SSIPARM(SSIPARMS) member:

```
ENF=YES
ENFARMWTO=YES
```

Restart Status Message Syntax

A generated restart status message has the following format:

```
NS4Unn ARM element_status R: restart_flag J: resource_name
       EL: element_type OS: old_system NS: new_system
       RG: restart_group_name
```

NS4U*nn*

Identifies the message.

nn

Identifies the message for a specify element status as indicated in the following table:

nn	*element_status*	Description
01	RG	The job or started task has registered as an element of ARM.
02	RD	The element is ready.
03	DR	The job or started task has deregistered from ARM.

element_status

Indicates the status of the element by which the job or started task is registered with ARM.

restart_flag

Indicates whether the job or started task has been restarted.

Limits: N (no) or Y (yes)

resource_name

Identifies the job or started task.

element_name

Identifies the element by which the job or started task is registered with ARM.

old_system

Identifies the previous system on which the job or started task was active.

new_system

Identifies the current system on which the job or started task is active.

restart_group_name

Identifies the restart group, if applicable, to which the job or started task belongs.

Appendix D: Message Attributes

This section contains the following topics:

Message Routing Codes

One or more routing codes are usually associated with a system message. The operating system uses routing codes to deliver messages to a particular console. When a system console is generated, it is assigned one or more routing codes. Routing codes can be used to send messages to a consolidated console.

The operating system, by convention, applies certain meanings to specific routing codes. You are encouraged to work within these conventions. System-sourced messages follow these conventions.

Note: MSP and VOS3 systems support routing codes 1 through 16 only.

The following values for message routing codes are defined:

- 1 is for master console action.
- 2 is for master console information.
- 3 is for tape pool.
- 4 is for direct-access pool.
- 5 is for tape library.
- 6 is for disk library.
- 7 is for unit record pool.
- 8 is for teleprocessing control.
- 9 is for system security.
- 10 is for system error, system maintenance, or system programmer information.
- 11 is for programmer information.
- 12 is for emulators.
- 13 through 20 are reserved for customer use.
- 21 through 28 are reserved for subsystem use.
- 29 through 41 are reserved for IBM use.
- 42 is for JES2 or JES3 information.
- 43 through 64 are reserved for JES use.
- 65 through 96 are for processor-related messages.
- 97 through 128 are for device-related messages.

Message Descriptor Codes

One or more descriptor codes are usually associated with a system message. The operating system uses descriptor codes to determine the display attributes of a message. Descriptor codes 1, 2, and 11 indicate that the message is non-roll deletable (NRD). If the message is also a WTOR, descriptor code 7 indicates NRD.

Descriptor codes 1 through 6 and 11 are mutually exclusive, whereas codes 7 through 10 can be used in combination with any other code.

The following values for message descriptor codes are supported:

- 1 indicates system failure.
- 2 indicates that immediate action is required.
- 3 indicates that eventual action is required.
- 4 indicates a system status message.
- 5 indicates an immediate command response.
- 6 indicates a job status message.
- 7 indicates an application program or processor message.
- 8 indicates an out-of-line message.
- 9 indicates an operator request.
- 10 is for dynamic status displays.
- 11 indicates that a critical eventual action is requested.
- 12 through 16 are reserved for future use.

Message Levels

Message levels can be used to limit the system messages that can be delivered to a specific system console or consolidated console user.

When a user ID has authority to receive messages captured by the Automation Services subsystem interface, those messages can be limited by specifying message levels.

The following values for message levels are supported:

- BC displays console broadcast messages.
- CE displays critical eventual action messages (descriptor code 11), which indicate a potential system problem.
- E displays eventual action messages (descriptor code 3), which do not require immediate action.
- I displays system failure and immediate action messages (descriptor codes 1 and 2), which indicate a task is waiting for operator action.
- IN displays informational messages.
- WTOR displays write-to-operator with reply messages, which usually require an operator reply.

Appendix E: Health Checks

This section contains the following topics:

CA Health Checker

The CA Health Checker provides a simple and consistent method for CA products to create health checks to run under the IBM Health Checker for z/OS. The IBM Health Checker for z/OS helps you identify potential problems in your z/OS environment by checking system or product parameters and system status against recommended settings. CA SOLVE:Operations Automation for CICS health checks are automatically activated on the target system when the product is started on a system with IBM Health Checker for z/OS installed and configured.

The CHECK_OWNER for all CA SOLVE:Operations Automation for CICS health checks is CA_NM.

Use either CA SYSVIEW or SDSF Health Checker displays to list and view the checks. View messages generated by CA health checks in the MVS System Log.

NM_ACB

Description

This CA SOLVE:Operations Automation for CICS health check checks that the primary ACB of the region is open. This check runs every 5 minutes.

Best Practice

VTAM is required to access the 3270 interface. If you primarily use the WebCenter interface to access you region, you can lower the priority of this health check.

Parameters accepted

None.

Debug Support

No.

Verbose Support

No.

Reference

None.

Non-exception Messages

The following messages can appear in health checker:

- This region's primary ACB, *acbname*, is open.
- The region is shutting down. Check is not relevant at this time.

Exception Messages

If an exception occurs, the following messages are issued as WTOs and written to the SYSLOG:

- CAH0001E The check timed out while waiting for a response to a command.
- NMH0106E This region's primary ACB, *acbname*, is not open.

NM_INITIALIZATION

Description

This CA SOLVE:Operations Automation for CICS health check checks region initialization. The check runs once at region startup. If an exception occurs, the check repeats every 5 minutes until initialization is successful.

Best Practice

Follow the Install Utility procedures in the *Installation Guide* to set up your region, and ensure that the parameters are specified correctly.

Parameters Accepted

None.

Debug Support

No.

Verbose Support

No.

Reference

See the online help for region parameter groups.

Non-exception Messages

The following messages can appear in health checker:

- The region has initialized successfully.
- The region is initializing. Check is not relevant at this time.
- The region is shutting down. Check is not relevant at this time

Exception Messages

If an exception occurs, the following messages are issued as WTOs and written to the SYSLOG:

- CAH0001E The check timed out while waiting for a response to a command.
- NMH0104E Initialization errors have occurred in region *regionname*.

NM_SOCKETS

Description

This CA SOLVE:Operations Automation for CICS health check checks that the sockets are available to support IP connections. The check runs every 15 minutes.

Best Practice

To help ensure IP connections, the port number for the connection must be specified and not in use by another task.

Parameters Accepted

None.

Debug Support

No.

Verbose Support

No.

Reference

None.

Non-exception Messages

The following messages can appear in health checker:

- Sockets are configured and active. HTTP port is *nnnn* URL is http://*nnn.nnn.nnn.nnn:nnnn*
- The region is initializing. Check is not relevant at this time.
- The region is shutting down. Check is not relevant at this time

Exception Messages

If an exception occurs, the following messages are issued as WTOs and written to the SYSLOG:

- CAH0001E The check timed out while waiting for a response to a command.
- NMH0110E TCP/IP interface is not active, status is *cccccccc*.
- NMH0111E No port number has been specified for this region.

NM_SSI

Description

This CA SOLVE:Operations Automation for CICS health check checks that the SOLVE SSI SSID is defined and connected. The check runs every 15 minutes.

Best Practice

Ensure that the following conditions are met:

- The SOLVE SSI started task is active.
- The SOLVE SSI SSID value for the region matches the SSID= parameter for the SOLVE SSI started task.
- The SOLVE SSI SSID and the AOM SSID are different.

Parameters Accepted

None.

Debug Support

No.

Verbose Support

No.

Reference

None.

Non-exception Messages

The following messages can appear in health checker:

- SOLVE SSI SSID correctly defined and connected. SSID is *ssidname*.
- The region is initializing. Check is not relevant at this time.
- The region is shutting down. Check is not relevant at this time.

Exception Messages

If an exception occurs, the following messages are issued as WTOs and written to the SYSLOG:

- CAH0001E The check timed out while waiting for a response to a command.
- NMH0108E SSID error, no SSID specified.
- NMH0108E SSID error, *ssidname* is not connected.
- NMH0108E SSID error, SSID matches AOM SSID(*ssidname*).

NM_WEB

Description

This CA SOLVE:Operations Automation for CICS health check checks that the WebCenter interface is available. This check runs every 15 minutes.

Best Practice

Use the Install Utility to set up the region. During the process, specify the web interface port.

Parameters Accepted

None.

Debug Support

No.

Verbose Support

No.

Reference

None.

Non-exception Messages

The following messages can appear in health checker:

- The region is initializing. Check is not relevant at this time.
- The region is shutting down. Check is not relevant at this time.
- The WebCenter interface is active. HTTP port is *nnnn* URL is http://*nnn.nnn.nnn.nnn:nnnn*

Exception Messages

If an exception occurs, the following messages are issued as WTOs and written to the SYSLOG:

- CAH0001E The check timed out while waiting for a response to a command.
- NMH0113E The WebCenter interface is not [active | configured].

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W