

MAINVIEW® AutoOPERATOR™

Solutions Guide

Version 6.2

March 15, 2002



Copyright © 2002 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc. IBM and DB2 are registered trademarks of International Business Machines Corp. All other registered trademarks or trademarks belong to their respective companies.

THE USE AND CONTENTS OF THIS DOCUMENTATION ARE GOVERNED BY THE SOFTWARE LICENSE AGREEMENT ENCLOSED AT THE BACK OF THIS DOCUMENTATION.

Restricted Rights Legend

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b), or DFARS 227.7202, as amended from time to time. Send any contract notices to Contractor/Manufacturer:

BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827
USA

Contacting BMC Software

You can access the BMC Software Web site at <http://www.bmc.com>. From this Web site, you can obtain general information about the company, its products, special events, and career opportunities. For a complete list of all BMC Software offices and locations, go to <http://www.bmc.com/corporate/offices.html>.

USA and Canada

Address BMC Software, Inc.
2101 CityWest Blvd.
Houston TX 77042-2827

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside USA and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

You can obtain technical support by using the Support page on the BMC Software Web site or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software,” below.

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, seven days a week by accessing the technical support Web site at <http://www.bmc.com/support.html>. From this site, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support via Telephone or E-mail

In the USA and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the USA and Canada, please contact your local support center for assistance. To find telephone and e-mail contact information for the BMC Software support center that services your location, refer to the Contact Customer Support section of the Support page on the BMC Software Web site at www.bmc.com/support.html.

Before Contacting BMC Software

Before you contact BMC Software, have the following information available so that a technical support analyst can begin working on your problem immediately:

- product information
 - product name
 - product version (release number)
 - license number and password (trial or permanent)
- operating-system and environment information
 - machine type
 - operating system type, version, and service pack or program temporary fix (PTF)
 - system hardware configuration
 - serial numbers
 - related software (database, application, and communication) including type, version, and service pack or PTF
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
 - product error messages
 - messages from the operating system, such as `file system full`
 - messages from related software

Contents

Part 1. Introduction	1
Chapter 1. Introduction to MAINVIEW AutoOPERATOR Solutions	3
EXEC Distribution	3
Rules Distribution	3
Documentation Boxes	3
Documentation Box Fields	5
Notes About Solutions	6
Part 2. MVS Solutions	7
Chapter 2. JES2 Solution	11
JES2 Monitoring Solution	11
Variables	11
Invocation	11
Processing Flow	12
EXECs	12
Rules	12
Chapter 3. VTAM/NCP Solution	13
TCAS Reply Solution	13
Variables	13
Invocation	13
Processing Flow	13
EXECs	13
Rules	14
Chapter 4. Storage Subsystems Solutions	15
Shared DASD Control Solution	15
Variables	15
Invocation	15
Processing Flow	16
EXECs	16
Rules	16
TLMS Solution	17
Variables	17
Invocation	17
Processing Flow	17
EXECs	17
Rules	17
Storage Reply Solution	18
Variables	18
Invocation	18
Processing Flow	18
EXECs	18
Rules	18

Chapter 5. Monitoring Solutions	19
Dump Data Sets Monitoring Solution	19
Variables	19
Invocation	19
Processing Flow	20
EXECs	20
Rules	20
WTO Buffers Monitoring Solution	21
Variables	21
Invocation	21
Processing Flow	21
EXECs	21
Rules	22
SMF Data Set Monitoring Solution	23
Variables	23
Invocation	23
Processing Flow	23
EXECs	24
Rules	24
LOGREC Data Set Monitoring Solution	25
Variables	25
Invocation	25
Processing Flow	25
EXECs	26
Rules	26
RMF Monitoring Solution	27
Variables	27
Invocation	27
Processing Flow	27
EXECs	27
Rules	27
Chapter 6. Access Methods Solution	29
VSAM Failures Solution	29
Variables	29
Invocation	29
Processing Flow	29
EXECs	29
Rules	30
Chapter 7. Job Scheduling Solution	31
CONTROL M Solution	31
Variables	31
Invocation	31
Processing Flow	31
EXECs	31
Rules	31
Chapter 8. MVS Performance Management Solutions	33
Load Balancing Solution	33
Variables	33
Invocation	34
Processing Flow	34
EXECs	34

Rules	34
DASD Reserve Analysis Solution	35
Variables	35
Invocation	35
Processing Flow	35
EXECs	35
Rules	36
Exception Monitoring Solution	37
Variables	37
Invocation	37
Processing Flow	38
EXECs	38
Rules	38

Part 3. CICS Solutions 39

Chapter 9. Error Recovery Solutions	41
Storage Violation Solution	42
Variables	42
Invocation	42
Processing Flow	42
EXECs	43
Rules	43
CICS Abnormal Termination Solution	44
Variables	44
Invocation	44
Processing Flow	44
EXECs	44
Rules	44
VSAM Subtask Abnormal Termination Solution	45
Variables	45
Invocation	45
Processing Flow	45
EXECs	46
Rules	46
CICS Temporary Storage Suspensions Solution (CICSTSS)	47
Variables	47
Invocation	47
Processing Flow	47
EXECs	48
Rules	48
Temporary Storage (TS) Data Set Extension Failure Solution	49
Variables	49
Invocation	49
Processing Flow	49
EXECs	49
Rules	50
Terminal Errors Solution	51
Variables	51
Invocation	51
Processing Flow	51
EXECs	51
Rules	52

Chapter 10. CICS Performance Management Solutions	53
File Degradation Analysis Solution	54
Variables	54
Invocation	54
Processing Flow	55
EXECs	55
Rules	56
File Allocation - Deallocation Solution	57
Variables and JCL	57
Invocation	57
Processing Flow	58
EXECs	58
Rules	58
VSAM Control Area Split Monitor Solution	59
Variables and JCL	59
Invocation	59
Processing Flow	59
EXECs	59
Rules	60
Transaction Response Time Monitor Solution	61
Variables and JCL	61
Invocation	61
Processing Flow	61
EXECs	61
Rules	62

Part 4. DB2 Solutions

Chapter 11. Initial Customization Steps	65
Steps	65
Chapter 12. DB2 Global Operations Solutions	69
DB2 Major Messages (DB2ALRT Solution)	69
Variables	69
Invocation	69
Processing Flow	71
EXECs	72
Rules	72
DB2 Environment Set-Up (DB2INIT Solution)	73
Variables	73
Invocation	74
Processing Flow	74
EXECs	74
Rules	74
DMR Major Messages (DMRALRT Solution)	75
Variables	75
Invocation	75
Processing Flow	76
EXECs	76
Rules	76
DMR Utilities (DMRUTIL Solution)	77
Variables	77
Invocation	77

EXECs	77
Rules	78
Chapter 13. DB2 Resource Contention Analysis Solutions	79
Deadlock/Timeout Analysis (DB2DLOK Solution)	79
Variables	79
Invocation	80
Processing Flow	80
EXECs	83
Rules	83
DB2 Table Space Filling Up (DB2TFUL Solution)	84
Variables	84
Invocation	85
Processing Flow	85
EXECs	85
Rules	85
DB2 Thread Control (DB2THRD Solution)	86
Variables	86
Invocation	86
Processing Flow	86
EXECs	86
Rules	87
Chapter 14. DB2 Performance Management Solutions	89
DB2 Runaway Query Control (DB2RNWY Solution)	89
Variables	89
Invocation	90
Processing Flow	90
EXECs	90
Rules	91
DB2 Response Time (DB2RESP Solution)	92
Variables	92
Invocation	92
Processing Flow	93
EXECs	93
Rules	93
<hr/>	
Part 5. Communication Solutions	95
Chapter 15. E-mail Solution	97
EXECs	97
Chapter 16. SNMP Solution	101
EXECs	101
Chapter 17. Using the Paging Sample	105
EXECs	105
QAOSNPP1	105
QAOSNPP2	105
Processing Flow	107
Variables Returned from QAOSNPP2	108
Chapter 18. Using the AutoOPERATOR Web Sample	111

Introduction	111
What AutoOPERATOR Web Is	111
How AutoOPERATOR Web is Used	112
What the Sample Web Pages Look Like	112
Prerequisites	112
What You Need to Know	112
Warning about Case Sensitivity	113
What Information to Gather	113
What Software Must Be Installed	114
Installing and Customizing the Samples	114
Running the Installation EXEC	114
Configuring the HTTP Server	115
Making BBSAMP Available to AutoOPERATOR	115
Meeting AOAnywhere Requirements	115
Reviewing Security	116
Accessing Multiple Systems	116
Accessing the Samples	116
Viewing the Entry Page	117
Viewing the About Page	118
Viewing the CSM Page	119
Viewing the ALERTs Page	122
Viewing the Automation Reporter Page	124
Viewing the Commands Page	126
Debugging the Distributed CGI Scripts	128
Understanding CGIs	128
Debugging the CGI Script in a TSO Environment	128
Debugging the CGI Script without HTML Decoding	128
Avoiding Frames in the Display	128
Displaying the HTML Source in a Frame	129
Refreshing HTML Source Code	129
Common Error Messages	129

Part 6. Appendixes 131

Appendix A. Sample EXECs	133
@TIMER	133
JESDOWN	144
Appendix B. Sample REXX EXECs	151
REXX Compound Variable Initialization EXECs	152
REXX DASD Reserve Analysis EXEC	154
REXX QUEUE/PULL and VPUT/VGET Usage	157
Appendix C. MVS Solution Variables	159
Appendix D. CICS Solution Variables	161
Appendix E. DB2 Solution Variables and Parameters	163
Data Sets Containing Variables	163
Data Sets Containing Threshold Parameters	163
Tables That Describe Variables and Parameters	164

Part 7. Glossary and Index	173
Glossary	175
Index	187

How To Use This Manual

The *MAINVIEW AutoOperator Solutions Guide* documents the solutions distributed by BMC Software to run under AutoOPERATOR. This edition of the guide contains MVS, CICS, and DB2 solutions.

Who Should Read This Manual

This manual is designed for use by installers who install MAINVIEW AutoOPERATOR solutions and by administrators who maintain MAINVIEW AutoOperator solutions as site requirements change.

How This Manual Is Organized

This guide is divided into six parts:

Part 1. Introduction

Introductory material that applies to all solutions

Part 2. MVS Solutions

Information on solutions that can affect the MVS environment

Part 3. CICS Solutions

Information on solutions that can affect the CICS environment

Part 4. DB2 Solutions

Information on solutions that can affect the DB2 environment

Part 5. Communication Solutions

Information on solutions that can affect the e-mail and SNMP environments

Part 6. Appendixes

Customer support information, examples of EXECs, and DB2 variables and parameters

Part 7. Glossary and Index

Contains the glossary and the index for this guide

MAINVIEW AutoOPERATOR Product Library

MAINVIEW AutoOPERATOR is available with seven options:

- MAINVIEW AutoOPERATOR for OS/390
- MAINVIEW AutoOPERATOR for IMS
- MAINVIEW AutoOPERATOR for CICS
- MAINVIEW AutoOPERATOR Access NV
- MAINVIEW AutoOPERATOR TapeSHARE
- MAINVIEW AutoOPERATOR for MQSeries
- MAINVIEW AutoOPERATOR Elan Workstation

The base product and these options are documented in the following MAINVIEW AutoOPERATOR manuals:

- *MAINVIEW AutoOPERATOR Customization Guide*
- *MAINVIEW AutoOPERATOR Basic Automation Guide*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*
- *MAINVIEW AutoOPERATOR Options User Guide*
- *MAINVIEW AutoOPERATOR for MQSeries Installation and User Guide*
- *MAINVIEW AutoOPERATOR Reference Summary*
- *MAINVIEW AutoOPERATOR Solutions Guide*

- Related Reading

To customize some products, you may need additional information not documented in this manual. Refer to the following publications:

- *MAINVIEW for DB2 User Guide*, which describes how to use the online monitor and analyzer services for use by the DB2 data base administrator, system programmer, or other performance analyst.
- *MAINVIEW Common Customization Guide*
- *MAINVIEW Administration Guide*
- *Using MAINVIEW*
- *IBM Advanced Communications Function for VTAM (ACF/VTAM)*, SC38-0256, which describes VTAM parameters for subsystem communication
- *IBM System Modification Program Extended*, SC28-1107, which gives a complete explanation of IBM's System Modification Program (SMP)

Product Requirements

You must have AutoOPERATOR installed to run MAINVIEW AutoOPERATOR solutions. In addition, the following MAINVIEW components are required for RULE SETs as shown in [Table 1](#).

Table 1. MAINVIEW Components Required for RULESETs

MAINVIEW Components	FMID: RULESET: TYPE	BBOA021 AAORUL00 MVS	BBOMS11 AAORULM1 MVS	BBOMCS11 AAORULM1 CICS	BBODS11 AAORULD1-D8 DB2
MAO	Yes	Yes	Yes	Yes	Yes
CAO			Yes	Yes	
IAO					Yes
RES + (Full RESOLVE product)			Yes		
IMF					Yes
CMR				Yes	
DMR					Yes

Naming Conventions for EXECs

The naming convention for the MAINVIEW AutoOPERATOR solutions EXECs uses the format:

gffnnnl

where:

g Is the first letter of the group (for example, M for MVS, I for IMS, C for CICS, D for DB2)

ff Is a two-character abbreviation for the functional area within the group:

Code	Functional Area
AM	ACCESS METHODS
ER	ERROR RECOVERY
IN	INFO SYS/MANAGEMENT
J2	JES2
J3	JES3
JS	JOB SCHEDULING
MN	MONITORS
PE	PERFORMANCE
SD	SHUT-DOWN
ST	STORAGE SUBSYSTEMS
SU	START-UP
UT	UTILITIES

VT VTAM/NCP

nnn Is a number from 001 to 999

l Is the language of the particular EXEC (for example, X for REXX, C for CLIST)

The following two categories of EXECs do not follow this naming convention:

- Operator-initiated EXECs, which are executed as a result of an operator command.
- Text ID-driven EXECs, which are executed as a result of a Write-To-Operator (WTO) or a Write-To-Operator Reply (WTOR). The names for text-ID-driven EXECs are the text-IDs.

What's New or Changed in MAINVIEW AutoOPERATOR Solutions 6.2

This section briefly describes the changes and enhancements to this document. The changes documented in this book are the result of PTF BPO2240 and are noted by revision bars (|). PTF BPO2240 removed specific support for DB2 releases earlier than 2.3 and MAINVIEW for DB2 releases earlier than 2.1. The remaining solutions support MAINVIEW for DB2 2.1 and 3.1 and DB2 2.3 and 3.1.

New and Changed EXECs

Some EXECs have been added or modified. They are described under “[DMR Utilities \(DMRUTIL Solution\)](#)” on page 77.

Changes to the AAORULD8 Rule Set

New messages have been added to the AAORULD8 rule set and are described under “[DB2 Major Messages \(DB2ALRT Solution\)](#)” on page 69.

Part 1. Introduction

The information in the introductory chapter is applicable to the solutions in [Part 2, “MVS Solutions”](#) on page 7, [Part 3, “CICS Solutions”](#) on page 39, and in [Part 4, “DB2 Solutions”](#) on page 63.

Chapter 1. Introduction to MAINVIEW AutoOPERATOR Solutions	3
EXEC Distribution	3
Rules Distribution	3
Documentation Boxes	3
Documentation Box Fields	5
Notes About Solutions	6

Chapter 1. Introduction to MAINVIEW AutoOPERATOR Solutions

The information in the introductory chapter is applicable to the solutions in:

- [Part 2, “MVS Solutions” on page 7](#)
- [Part 3, “CICS Solutions” on page 39](#)
- [Part 4, “DB2 Solutions” on page 63](#)

BMC Software provides a platform for developing automated solutions to problems within a data center. This platform consists of Rules, EXECs, timer facilities, and SYSPROG Services commands.

Using this platform, BMC Software has developed solutions to problems that are common across many data centers. These solutions assist you in accomplishing many of the initial automation tasks quickly and efficiently after you install the MAINVIEW AutoOPERATOR product.

EXEC Distribution

All EXECs for solutions are distributed in the BBPROC data set allocated to the SYSPROC DD statement in the AutoOPERATOR subsystem.

Rules Distribution

All Rules for MAINVIEW AutoOPERATOR Solutions are distributed DISABLED in the BBPARM data set. The distribution of Rules within this data set differs depending on the environment of the solution.

Table 2. Rules Distribution in Different Environments

Solution Environment	Distribution of Rules
MVS and CICS	One set of Rules for all solutions in one member (AAORULM1 and AAORULC1, respectively)
DB2	One set of Rules for each solution in a separate member (AAORULM*)

For information on how to enable any of these sets of Rules, any Rules within the individual solutions, or other sets of Rules in the AutoOPERATOR product, see the Rules chapters of the *AutoOPERATOR Basic Automation Guide*.

Documentation Boxes

The MAINVIEW AutoOPERATOR Solutions EXECs distributed with this release have been renamed and enhanced with a documentation box. [Figure 1 on page 4](#) shows an example of a documentation box.

```

/*****
/* DOC GROUP(MVS) FUNC(MONITORS) CODE(MN) */
/* DOC DISP(YES) AUTHOR(B&B) */
/* DOC DESC(RESOLVE WTO BUF SHORTAGE) */
/*-----*/
/* NAME: */
/* IEE249I */
/* */
/* DESCRIPTION: */
/* RESOLVE THE WTO BUFFER SHORTAGE SITUATION */
/* */
/* SEE ALSO: */
/* IEA404A */
/* IEA405E */
/* */
/* INVOKED BY: */
/* IEA404A */
/* IEA405E */
/* */
/* INPUT PARAMETERS: */
/* NONE */
/* */
/* OUTPUT: */
/* NONE */
/* */
/* OUTBOARD CALLS: */
/* NONE */
/* */
/* EXTERNAL ROUTINES CALLED: */
/* MVS "K" OR "CONTROL" COMMANDS ARE ISSUED. */
/* NOTE: FOR THE "K" COMMANDS TO BE PROPERLY AUTHORIZED, THEY */
/* ---- MUST BE ISSUED FROM A NON-SUBSYSTEM CONSOLE. EDIT */
/* THE MEMBER, BBISSPOO, IN YOUR BBPARM DATA SET AND */
/* ADD THE KEY WORD, CMDCON=1; THIS WILL DIRECT ALL COM- */
/* MANDS WHICH DO NOT REQUIRE A RESPONSE TO CONSOLE */
/* ID 1. */
/* */
/* ALERTS ISSUED: */
/* NONE */
/* */
/* VARIABLES USED: */
/* NAME: DESCRIPTION: DEFAULT VALUE: */
/* MSG MESSAGE ID IEE249I */
/* P1 WORD 1 CUA OF CONSOLE OR SYSLOG*/
/* P2 WORD 2 CONSOLE ID */
/* P3 WORD 3 COND= */
/* P4 WORD 4 AUTH= */
/* P5 WORD 5 NBUF= */
/* */
/* TEST SCRIPT INFO: */
/* GENERATE A WTO FROM A TEST EXEC TO TRIGGER A CALL TO */
/* EITHER OF THE DRIVER EXECs: IEA404A OR IEA405E */
/* */
/* RETURN CODES: */
/* NONE */
/* */
/* CHANGE LOG: */
/* 03-05-92 BY MDR: ADDED DOCUMENTATION BOX */

```

Figure 1. Example of a Documentation Box

The information contained at the top of the documentation box is used by the MAINVIEW AutoOPERATOR EXEC Management Application to display information about the EXECs in the SYSPROC libraries. GROUP, FUNC, CODE, DISP, AUTHOR, and DESC are displayed on one line by the MAINVIEW AutoOPERATOR EXEC Management Application (refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide* for a detailed explanation). The rest of the information in the documentation box is for self-documentation and easy reference.

Documentation Box Fields

The following is a brief description of the title fields shown in [Figure 1 on page 4](#).

Field	Description
GROUP	Is the group to which this EXEC belongs (for example, MVS, CICS, DB2, or IMS)
FUNC	Is the functional category within the group
CODE	Is the two-character function code
DISP	Shows whether information about this EXEC should be displayed under the MAINVIEW AutoOPERATOR EXEC Management Application (refer to the <i>MAINVIEW AutoOPERATOR Basic Automation Guide</i>)
AUTHOR	Is the author of the EXEC
DESC	Is a 25 or fewer word description of what the EXEC does
NAME	Is the name of the EXEC
DESCRIPTION	Is a detailed description of what the EXEC does
SEE ALSO	Is a cross-reference listing
INVOKED BY	Is a list of other EXECs that call this one
INPUT PARAMETERS	Indicates the parameter name, description, and default value required for input parameters
OUTPUT	Is the information or names of variables that are returned
OUTBOARD CALLS	Indicates whether any pager calls have been placed
EXTERNAL ROUTINES CALLED	Other routines or services called, along with any special considerations for those calls
ALERTS ISSUED	If this is a long EXEC and ALERTS are issued, this supplies brief text; otherwise, a YES or NO is returned
VARIABLES USED	Indicates the variable name, description, and default value of any variables used
TEST SCRIPT INFO	Indicates the necessary environment and steps needed to test the EXEC if changes are required
RETURN CODES	Shows what return codes (if any) are returned
CHANGE LOG	Is a chronology of changes

Notes About Solutions

- Some national characters do not translate correctly when you translate from EBCDIC format to ASCII format and then back to an EBCDIC format. In particular, the national characters #, !, %, and @ translate differently depending on the country in which the translation takes place. BMC Software recommends that you avoid using national characters in your naming conventions.
- You might want to customize or modify the solutions. If you do modify a solution, BMC Software recommends that you retain the original EXEC or rule for reference. To do so:
 1. Copy the solution's components into the UBBPROC data set
 2. Make the modifications
 3. Concatenate the UBBPROC data set to the front of the BBPROC data set in your BBI-SS JCL

Part 2. MVS Solutions

The distributed MVS Solutions fall into seven functional categories:

- JES2
- VTAM/NCP
- Storage Subsystems
- Monitors
- Access Methods
- Job Scheduling
- Performance Management

The following chapters discuss each function, their automation subjects, and the distributed solutions.

Chapter 2. JES2 Solution	11
JES2 Monitoring Solution	11
Variables	11
Invocation	11
Processing Flow	12
EXECs	12
Rules	12
Chapter 3. VTAM/NCP Solution	13
TCAS Reply Solution	13
Variables	13
Invocation	13
Processing Flow	13
EXECs	13
Rules	14
Chapter 4. Storage Subsystems Solutions	15
Shared DASD Control Solution	15
Variables	15
Invocation	15
Processing Flow	16
EXECs	16
Rules	16
TLMS Solution	17
Variables	17
Invocation	17
Processing Flow	17
EXECs	17
Rules	17
Storage Reply Solution	18
Variables	18
Invocation	18
Processing Flow	18
EXECs	18
Rules	18
Chapter 5. Monitoring Solutions	19
Dump Data Sets Monitoring Solution	19

Variables	19
Invocation	19
Processing Flow	20
EXECs	20
Rules	20
WTO Buffers Monitoring Solution	21
Variables	21
Invocation	21
Processing Flow	21
EXECs	21
Rules	22
SMF Data Set Monitoring Solution	23
Variables	23
Invocation	23
Processing Flow	23
EXECs	24
Rules	24
LOGREC Data Set Monitoring Solution	25
Variables	25
Invocation	25
Processing Flow	25
EXECs	26
Rules	26
RMF Monitoring Solution	27
Variables	27
Invocation	27
Processing Flow	27
EXECs	27
Rules	27
Chapter 6. Access Methods Solution	29
VSAM Failures Solution	29
Variables	29
Invocation	29
Processing Flow	29
EXECs	29
Rules	30
Chapter 7. Job Scheduling Solution	31
CONTROL M Solution	31
Variables	31
Invocation	31
Processing Flow	31
EXECs	31
Rules	31
Chapter 8. MVS Performance Management Solutions	33
Load Balancing Solution	33
Variables	33
Invocation	34
Processing Flow	34
EXECs	34
Rules	34
DASD Reserve Analysis Solution	35

Variables	35
Invocation	35
Processing Flow	35
EXECs.....	35
Rules	36
Exception Monitoring Solution	37
Variables	37
Invocation	37
Processing Flow	38
EXECs.....	38
Rules	38

Chapter 2. JES2 Solution

The Job Entry Subsystem (JES) is critical to the processing flow of your MVS environment. There might be jobs that execute under the control of JES that are critical to your data center.

BMC Software provides the JES2 Monitoring Solution, which alerts data center personnel of any JES2-related problems.

JES2 Monitoring Solution

The Job Entry Subsystem is a critical MVS subsystem. If JES2 is not functioning correctly, the throughput on the MVS system is adversely affected. The JES2 Monitoring Solution ensures that JES2 system problems are recognized immediately.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

SYSPROG	TSO user ID of the primary system programmer who is to receive warning messages.
SYSBEEP	Information to be placed on pager.
MVSCALL	Name of support person for Elan to page. This name must be defined on the Elan Workstation.
SYSJES	Version, release, and modification level of JES in the form <i>SPn.n.n</i> .
SYSMVS	Version, release, and modification level of MVS in the form <i>SPn.n.n</i> .

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing variables and default settings.

Invocation

This solution is invoked by Rules for the following JES2 messages:

\$HASP050	JES2 Resource Shortage
\$HASP093	nn% Spool Utilization (JES2 V2)
\$HASP095	JES2 Catastrophic Error Or Abend
\$HASP646	nn% Spool Utilization (JES2 V3)

Processing Flow

When any of the following messages are received, an ALERT is issued with an alarm. If a value is specified for the SYSPROG variable, a message is sent to the system programmer's TSO user ID.

\$HASP050	JES2 Resource Shortage
\$HASP093	nn% Spool Utilization (with greater than 80% utilization)
\$HASP095	JES2 Catastrophic Error Or Abend
\$HASP355	Spool Volumes Are Full
\$HASP646	nn% Spool Utilization (with greater than 80% utilization)

For the \$HASP095 and \$HASP355 messages, an outboard pager call is also made if all the following conditions are met:

- The outboard component is installed.
- A value is supplied for MVSCALL, specifying the person who is paged.
- A value is supplied for SYSBEEP, specifying the information that is sent to the pager.

For \$HASP355, the command to drain the initiators is issued.

EXECs

The EXECs for the JES2 Monitoring Solutions are:

\$HASP050	Handles \$HASP050 message
\$HASP093	Handles \$HASP093 message
\$HASP095	Handles \$HASP095 message
\$HASP355	Handles \$HASP355 message
\$HASP646	Handles \$HASP646 message

Rules

The Rules for the JES2 Monitoring Solution are:

\$HASP050	Text-ID=\$HASP050
\$HASP093	Text-ID=\$HASP093
\$HASP095	Text-ID=\$HASP095
\$HASP355	Text-ID=\$HASP355
\$HASP646	Text-ID=\$HASP646

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

Chapter 3. VTAM/NCP Solution

Much of the work in data centers is performed interactively using online systems and terminal networks. However, the commands and replies used to communicate with the network software can be cumbersome to use.

BMC Software provides the TCAS Reply Solution, which provides automatic replies to TCAS WTORs.

TCAS Reply Solution

When TCAS ends either normally or abnormally, WTORs are issued. If these WTORs are not replied to correctly and in a timely manner, the down time for TSO can be extended unnecessarily. This solution ensures that the TCAS WTORs are replied to correctly and immediately.

Variables

This solution does not require any values to be set for variables.

Invocation

This solution is invoked by Rules for the following TCAS messages:

IKT001D	nnn Users Active Reply U, SIC or FSTOP
IKT010D	nnn Users Active Reply SIC or FSTOP
IKT012D	TCAS termination in progress - specify U or DUMP

Processing Flow

When either of the following WTORs is received, a reply of SIC is generated:

IKT001D	nnn Users Active Reply U, SIC or FSTOP (received when TCAS is started after abnormal termination)
IKT010D	nnn Users Active Reply SIC or FSTOP (received when TCAS is stopped)

When the IKT012D TCAS Termination In Progress - Specify U or DUMP WTOR is received, a reply of U is generated.

EXECs

The EXECs for the TCAS Reply Solution are:

IKT001D	Handle IKT001D message
IKT010D	Handle IKT010D message
IKT012D	Handle IKT012D message

Rules

The Rules for the TCAS Reply Solution are:

IKT001D	Text-ID=IKT001D
IKT010D	Text-ID=IKT010D
IKT012D	Text-ID=IKT012D

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.

Chapter 4. Storage Subsystems Solutions

The DASD and tape subsystems can be just as important to your data center's performance as the processor. The commands and replies used to communicate with the storage subsystems are not always easy to use. Shared DASD environments present an even greater challenge.

BMC Software provides the following Storage Subsystems Solutions:

Shared DASD Control	Allows devices to be varied online/offline to all shared systems with one command
TLMS	Provides a console interface to TLMS
Storage Reply	Provides automatic reply to storage WTORs

Shared DASD Control Solution

You must issue commands on each system in the shared DASD configuration to change the status of a volume. This can also require switching consoles.

The Shared DASD Control Solution allows the status of a shared DASD volume to be changed using one command. The status change is automatically propagated to the other systems in the configuration.

Variables

For this solution to function properly in your environment, you must establish values for the following variables.

SYSN	Number of MVS systems in the shared DASD configuration
SYS1-SYSn	Names of the primary MAINVIEW AutoOPERATOR subsystems on each MVS system in the shared DASD configuration

See [Appendix C, "MVS Solution Variables" on page 159](#) for information about initializing the variables and default settings.

Invocation

This solution is operator-invoked. Enter:

```
%XSYSVARY P1 P2 P3
```

on the COMMAND line to change the status of a shared DASD volume, where:

P1	Is the UCB address
P2	Is the desired status (for example, online or offline)
P3	Should be SHR for 3480s

Processing Flow

When the operator requests that the status of a device be changed, an EXEC is scheduled to execute on each system in the shared DASD complex.

The scheduled EXEC issues either the VARY DEV, ONLINE or VARY DEV, OFFLINE command depending on the operator specification.

EXECs

The EXECs for the Shared DASD Control Solution are:

XSYSVARY	Schedule EXECs on all systems
MST001C	Issue VARY commands

Rules

There are no Rules for the Shared DASD Control Solution.

TLMS Solution

Some data centers dedicate a console to the TLMS INQR task so the operator need not search constantly for the outstanding reply number using the D R, L command.

This solution lets TLMS commands be entered from the MCS console without knowing the outstanding reply number.

Variables

This solution does not require any values to be set for variables.

Invocation

This solution is operator-invoked. Enter:

```
%TLMS P1 P2
```

on the COMMAND line to issue a TLMS command, where:

P1 Is the TLMS command; valid values are DV, DVA, DVH, DVL, DVM, and DVR
P2 Is the volume serial number

Processing Flow

The RESOLVE REPLIES command is used to determine the outstanding reply number for the CAT2291D Message ID.

When the outstanding reply number is found, the TLMS command is issued using the command and volume specified by the operator.

EXECs

The EXEC for the TLMS Solution is:

TLMS Finds outstanding reply, issues command

Rules

There are no Rules for the TLMS Solution.

Storage Reply Solution

The DASD and tape storage subsystems issue WTORs that require operator intervention. If these are not replied to in a timely manner, allocation queues can back up and cause degradation of system throughput. This solution ensures that the storage WTORs are replied to correctly and immediately.

Variables

This solution does not require any values to be set for variables.

Invocation

This solution is invoked by Rules for the following messages:

IEC701D	M ddd, Volume To Be Labeled ser
IEF238D	Reply Device Name, Wait or Cancel
IEF433D	Wait Requested - Reply Hold or Nohold

Processing Flow

The following table shows the replies generated automatically for each WTOR.

Table 3. WTOR Generated Replies

WTOR	Description	Reply
IEC701D	M ddd, Volume To Be Labeled ser	M
IEF238D	Reply Device Name, Wait or Cancel	Wait
IEF433D	Wait Requested - Reply Hold or Nohold	Nohold

EXECs

The EXECs for the Storage Reply Solution are:

IEC701D	Handles IEC701D Message
IEF238D	Handles IEF238D Message
IEF433D	Handles IEF433D Message

Rules

The Rules for the Storage Reply Solution are:

IEC701D	Text-ID=IEC701D
IEF238D	Text-ID=IEF238D
IEF433D	Text-ID=IEF433D

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

Chapter 5. Monitoring Solutions

The following Monitoring Solutions, provided by BMC Software, are designed to assist data center personnel in monitoring events occurring in the system:

Dump Data Sets	Manages dump data sets to keep a minimum number available
WTO Buffers	Resolves WTO buffer shortages
SMF Data Sets	Manages dumping and switching of SMF data sets
LOGREC	Manages clearing of SYS1.LOGREC
RMF Monitor	Monitors status of RMF

Dump Data Sets Monitoring Solution

This solution lets the system automatically manage its system dump data sets. It attempts to keep a minimum number of dump data sets free at all times.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

DUMPCLER	Name of task to clear dump data sets
DUMPMF	Number of dump data sets to keep clear
DUMPNDS	Total number of dump data sets

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

Invocation

This solution is invoked by Rules for the following messages:

IEA911E	Complete/Partial Dump on SYS1.DUMPnn
IEA994A	All Dump Data Sets Are Full And No SVC Dumps Can Be Taken
IEA994E	All Allocated SYS1.DUMP Data Sets Are Full

This solution can also be operator-invoked. Type:

```
%@DUMPCLRO
```

to clear the oldest dump data set.

Processing Flow

All of the following processing checks for a value specification for the variable DUMPCLER. If a value was not specified, processing ends.

When the IEA911E Complete/Partial Dump on SYS1. DUMPnn message is received, the current number of free dump data sets is compared to the minimum number that should be kept free. If the number of free data sets is less than the minimum, a task to clear the oldest dump data set is started.

When either of the following messages is received, a task to clear the oldest dump data set is started:

IEA994A All Dump Data Sets Are Full And No SVC Dumps Can Be Taken

IEA994E All Allocated SYS1.DUMP Data Sets Are Full

EXECs

The EXECs for the Dump Data Sets Monitoring Solution are:

@DMPCLRO Starts task to clear oldest dump data set
IEA911E Compares minimum free with current free data sets
IEA994A Invokes @DMPCLRO
IEA994E Invokes @DMPCLRO

Rules

The Rules for the Dump Data Sets Monitoring Solution are:

IEA911E Text-ID=IEA911E
IEA994A Text-ID=IEA994A
IEA994E Text-ID=IEA994E

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

WTO Buffers Monitoring Solution

This solution lets the system automatically resolve WTO buffer shortages.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

SYSPROG TSO userid of primary system programmer to receive warning messages

SYSMVS Version, release, and modification level of the MVS system in use

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

In addition to setting variables, the parameter CMDCON must be specified in member BBISSP00 of the BBPARM data set. This is required because the K Q command used to clear buffers cannot be issued from a subsystem console. Specifying CMDCON causes all commands issued without response to be issued with the CMDCON console ID.

Invocation

This solution is invoked by Rules for the following messages:

IEA404A Severe WTO Buffer Shortage - 100% Full

IEA405E Severe WTO Buffer Shortage - 80% Full

Processing Flow

When either the IEA404A or IEA405E buffer shortage message is received, an ALERT is generated. If a value was specified for SYSPROG, a warning message is sent to the primary system programmer.

The D C command is issued to determine the number of buffers in use by each console. The buffers are deleted using one of the following commands:

K Q, L=console id (clearing console buffers)
WRI TELOG (clearing syslog buffers)

After the shortage is relieved, the ALERT is deleted. If a value was specified for SYSPROG, a message is sent to notify the primary system programmer that the shortage has been relieved.

EXECs

The EXECs for the WTO Buffers Monitoring Solution are:

IEA404A Generate ALERTs, invoke IEE249I/IEE889I
IEA405E Generate ALERTs, invoke IEE249I/IEE889I
IEE249I Clear buffers (pre-MVS SP4)
IEE889I Clear buffers (SP4 and later)

Rules

The Rules for the WTO Buffers Monitoring Solution are:

IEA404A Text-ID=IEA404A

IEA405E Text-ID=IEA405E

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.

SMF Data Set Monitoring Solution

This solution manages dumping and switching of the system to automatically determine which other system in the shared DASD configuration is preventing access to a shared device.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

SMFALT	Suffix of alternate SMF parameters
SMFCLEAR	Task name to dump SMF data sets

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

Invocation

This solution is invoked by Rules for the following SMF messages:

IEE361I	SMF Data Lost - No Data Sets Available
IEE362A	SMF Enter Dump For SYS1.MANx On ser
IEE364I	SMF (Logical/Physical) Error On SYS1.MANx
IEE366I	No SMF Data Sets Available - Data Being Buffered

Processing Flow

The following describes the processing flow for the SMF Data Set Monitoring Solution.

When the IEE362A SMF Enter Dump For SYS1.MANx ON SER message is received, the task to dump the data set (SMFCLEAR) is started. If a value was not specified for the SMFCLEAR variable, no processing occurs.

When the IEE949I message (output from D SMF command) is received, the task to dump the data set (SMFCLEAR) is started if it is not already active. If a value was not specified for the SMFCLEAR variable, no processing occurs.

When the IEE361I, IEE364I, or IEE366I messages are received, the primary system programmer is notified if a value was specified for the SYSPROG variable.

If a value was specified for variable SMFALT, the operator receives an ALERT requesting that the alternate SMF parameters be switched. If the operator confirms the switch, the T SMF command is issued. The primary system programmer is notified of the switch if a value was specified for the SYSPROG variable.

EXECs

The EXECs for the SMF Data Set Monitoring Solution are:

IEE361I	Handles IEE361I message
IEE362A	Handles IEE362A message
IEE364I	Handles IEE364I message
IEE366I	Handles IEE366I message
IEE949I	Handles IEE949I message
MMN001C	Switches to alternate SMF parameters

Rules

The Rules for the SMF Data Set Monitoring Solution are:

IEE361I	Text-ID=IEE361I
IEE362A	Text-ID=IEE362A
IEE364I	Text-ID=IEE364I
IEE366I	Text-ID=IEE365I
IEE949I	Text-ID=IEE949I

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

LOGREC Data Set Monitoring Solution

This solution lets the system automatically manage the SYS1.LOGREC data set.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

LOGREC	Task name to clear SYS1.LOGREC data set
SYSPROG	TSO user ID of primary system programmer to receive warning messages
SYSBEEP	Information to be placed on pager
MVSCALL	Name of the support person for Elan to page; this name must be defined on the Elan Workstation

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

Invocation

This solution is invoked by a rule for the following messages:

IFB040I	SYS1.LOGREC Area Is Full
IFB060E	SYS1.LOGREC Near Full
IFB070I	Logrec Cannot Be Accessed. Record Is Lost

Processing Flow

When either the IFB040I or IFB060E LOGREC FULL message is received and a value was specified for variable LOGREC, the following actions are taken:

- The task to clear SYS1.LOGREC is started.
- An ALERT is issued.
- If a value was specified for variable SYSPROG, a message is sent to the system programmer.

When the IFB070I Logrec Cannot Be Accessed message is received, the following actions are taken:

- An ALERT is issued.
- If a value was specified for variable SYSPROG, a notification is sent to the system programmer.
- If a value was specified for variable MVSCALL and SYSBEEP, an outboard pager call is made.

EXECs

The EXECs for the LOGREC Data Set Monitoring Solution are:

IFB040I	Handles IFB040I message
IFB060E	Handles IFB060E message
IFB070I	Handles IFB070I message

Rules

The Rules for the LOGREC Data Set Monitoring Solution are:

IFB040I	Text-ID=IFB040I
IFB060E	Text-ID=IFB060E
IFB070I	Text-ID=IFB070I

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

RMF Monitoring Solution

This solution issues an ALERT if RMF ends.

Variables

This solution does not require any values to be set for variables.

Invocation

This solution is invoked by a rule for the following message:

ERB102I sid: Terminated

Processing Flow

When the ERB102I message is received, an ALERT is issued.

EXECs

The EXEC for the RMF Monitoring Solution is:

ERB102I Handles ERB102I message

Rules

The rule for the RMF Monitoring Solution is:

ERB102I Text-ID=ERB102I

The rule is distributed DISABLED in RULESET AAORULM1. You must ENABLE this rule to implement this solution.

Chapter 6. Access Methods Solution

The Access Methods Solution is designed to alert the data center personnel when failures occur during I/O processing.

BMC Software provides the VSAM Failures Solution to alert the operator when a failure occurs in specified address spaces.

VSAM Failures Solution

This solution alerts the operator when VSAM I/O fails in a critical address space or job stream.

Variables

For this solution to function properly in your environment, edit the IDC3009I and IDC3351I EXECs to set values for the jobs where VSAM failures are critical.

SYSPROG TSO user ID of primary system programmer to receive warning messages

SYSBEEP Information to be placed on pager

MVSCALL Name of the support person for Elan to page; this name must be defined on the Elan Workstation

See [Appendix C, “MVS Solution Variables” on page 159](#) for information on these variables.

Invocation

This solution is invoked by Rules for the following text-IDs:

IDC3009I VSAM CATALOG RETURN CODE IS rc

IDC3351I VSAM OPEN|CLOSE|I/O RETURN CODE IS rc

Processing Flow

When the message for either IDC3009I or IDC3351I is received, a comparison is made against the originating job name and the user-specified job names; if a match is found, an ALERT is issued to inform the operator of the failure.

If values for the SYSPROG, SYSBEEP, and MVSCALL variables were established during MAINVIEW AutoOPERATOR initialization, the processing continues by sending a TSO message to SYSPROG and by paging the SYSBEEP number through the outboard processor.

EXECs

The EXECs for the VSAM Failures Solution are:

IDC3009I Issues an ALERT

IDC3351I Issues an ALERT

Rules

The Rules for the VSAM Failures Solution are:

IDC3009I Text-ID=IDC3009I

IDC3351I Text-ID=IDC3351I

The Rules are distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** these Rules to implement this solution.

Chapter 7. Job Scheduling Solution

The following Job Scheduling Solution is designed to assist data center personnel schedule and manage time dependent tasks:

CONTROLM Provides a command interface to CONTROLM

CONTROLM Solution

The operator uses an MVS **MODIFY** command to communicate with the CONTROLM job scheduling software. This solution provides a command interface to CONTROLM that alleviates the need for the **MODIFY** command.

Variables

This solution does not require you to set values for any variables.

Invocation

This solution is operator-invoked. Enter:

```
%SCHEDCOM P1
```

on the **COMMAND** line to issue a **CONTROLM** command, where:

P1 CONTROLM command; valid values are CTMX004 and NEWDEST

Processing Flow

The following command is issued using the input parameter as the **CONTROLM** command:

```
F CONTROLM, cmd
```

EXECs

The EXEC for the CONTROLM Solution is:

SCHEDCOM Issues **MODIFY** command

Rules

There are no rules for the CONTROLM Solution.

Chapter 8. MVS Performance Management Solutions

The Performance Management Solutions are designed to assist data center personnel monitor and adjust system parameters affecting system performance.

BMC Software provides the following three solutions:

Load Balancing	Lets the system dynamically adjust job initiator structure based on performance considerations
DASD Reserve Analysis	Provides immediate insight into contention problems in a shared DASD environment
Exception Monitoring	Extends the capabilities of the RESOLVE Advanced Early Warning (AEW) system

Load Balancing Solution

As workload type and activity on your system changes, you might want to adjust your job initiator configuration. For example, you might want to take advantage of times when the system resources support additional initiators without adversely affecting system performance. Or you might want to limit access to the system at times when resources are constrained.

This solution lets the system automatically increase or decrease its workload (initiators) based on current system performance and job demand.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

LBGPAGE	Paging rate at which an attempt is made to decrease workload
LBGCPUH	CPU utilization percentage at which an attempt is made to decrease workload
LBGCPUL	CPU utilization percentage at which an attempt is made to increase workload
LBGIBEG	Beginning initiator number to manage
LBGIEND	Ending initiator number to manage
LBGIINC	Number of initiators to start in an increase situation
LBGIDEC	Number of initiators to stop in an decrease situation
LBGCLAS	Class priority list from high to low

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

Invocation

This solution is operator-invoked. Type:

```
%LBSTART
```

on the COMMAND line to begin load balancing on your system.

Processing Flow

Every five minutes, this solution compares the current CPU utilization and paging rate to the threshold variables. If any threshold is crossed 3 times within 15 minutes, an action is taken.

If CPU utilization is below the low CPU threshold, an attempt is made to increase the workload by starting additional initiators.

Initiators (up to the maximum specified by variable LBGIIINC) that have a status of either DRAINED or HALTED are started. Classes are assigned to the initiators based on the class priority list (variable LBGCLAS) and work waiting to execute.

If the CPU utilization is above the high CPU threshold or the page rate is above the page rate threshold, an attempt is made to decrease the workload by stopping initiators. Initiators (up to the maximum specified by variable LBGIDEC) that have a status of either active or INACTIVE are purged or drained.

EXECs

Following are the EXECs for the Load Balancing Solution:

LBSTART	Initializes CPU and page threshold counters, calls MPE003C to build initiator table, schedules MPE004C to execute in five minutes
MPE003C	Retrieves initiator information from JES
MPE004C	Manages CPU and page threshold counters, schedules MPE005C to execute
MPE005C	Compares current CPU and page values against thresholds (if adjustment is necessary), and schedules MPE006C to execute or schedules MPE004C to execute in five minutes
MPE006C	Starts/stops initiators and schedules MPE004C to execute again in five minutes

Rules

There are no Rules for the Load Balancing Solution.

DASD Reserve Analysis Solution

In shared DASD environments, one or more systems can be prevented from accessing an entire DASD volume due to hardware reserves by another system. If a reserve remains in effect for an extended period of time, end-user response time can be severely affected.

This solution lets the system automatically determine which other system in the shared DASD configuration is preventing access to a shared device.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

SYSN	Number of MVS systems in the shared DASD configuration
SYS1-SYSn	Names of the primary MAINVIEW AutoOPERATOR subsystem on each MVS system in the shared DASD configuration

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

Invocation

This solution is invoked by a rule for the following IOS message:

```
IOS071I UCB, CHPID, JOBNAME, START PENDING
```

Processing Flow

When the IOS071I message is received, a remote EXEC is scheduled on every other system in the shared DASD complex.

Each of the other systems use the RESOLVE RESERVE command to examine the DASD reserves it is currently holding. Any system that has a device reserved issues an operator ALERT back to the system that is being prevented access to the device.

EXECs

The EXECs for the DASD Reserve Analysis Solution are:

IOS071I	Schedules remote EXECs
MPE001C	Examines currently held reserves
MPE002C	Issues ALERT

Rules

The rule for the DASD Reserve Analysis Solution is:

IOS071I Text-ID=IOS071I

The rule is distributed **DISABLED** in RULESET AAORULM1. You must **ENABLE** this rule to implement this solution.

Exception Monitoring Solution

If you use BMC Software's SYSPROG Services product, you might want to extend its Advanced Early Warning (AEW) capabilities.

This solution lets RESOLVE AEW messages be posted to the MAINVIEW AutoOPERATOR ALERT application and, in some instances, provides information about the causes of an exception condition.

Variables

For this solution to function properly in your environment, you must establish values for the following variables:

MAXCCPU	Maximum complex CPU utilization percentage
MAXTCPU	Maximum CPU utilization by TSO address spaces
MAXBCSA	Maximum CSA utilization percentage below the 16M line
MAXACSA	Maximum extended CSA utilization percentage
MAXDEVU	Maximum device utilization percentage

See [Appendix C, “MVS Solution Variables” on page 159](#) for information about initializing the variables and default settings.

Invocation

To invoke the Exception Monitoring Solution:

1. The entire RESOLVE PLUS software is required to use the MVS Performance Monitoring Solution.

ENABLE the following Rules in the AAORULM1 RULESET:

```
PWSCPU00 *WARNING* CPU 1 USAGE IS (x)%
```

OR

```
PWSCPU01 *WARNING* CPU COMPLEX USAGE IS (x)%
```

```
PWSCPU02 *WARNING* CPU USAGE IS (x)% FOR TSO
```

```
PWSCSA01 *WARNING* CSA/ECSA USAGE IS (x)%; (x)K ARE FREE
```

```
PWSDEV01 *WARNING* DVN VOLUME USAGE IS (x)%
```

2. To install these solutions:

- a. Activate the AEW sampler.

```
COPY <PREFIX>. BBPARM(PWSCPMZZ) TO <PREFIX>. UBBPARM, DISP=SHR
```

- b. Add the following statement to your BBI-SS JCL to activate the PWSCPM00 rule.

```
//LIB DD DSN=<ORPREFIX>. UBBPARM, DISP=SHR
```

Restart your BBI subsystem after making those changes.

3. Set the variables for these solutions in <PREFIX>. UBBPROC member MSU002C.

Additional information is provided in MSU002C and PWSCPM00.

4. ENABLE the Rules in RULESET AAORULM1.

If you have only one CPU in the system in which you are running this BBI subsystem, ENABLE rule PWSCPU00.

If there are multiple CPUs running, ENABLE rule PWSCPU01.

Issue a RESOLVE CPU command to check the number of CPUs running in your system.

Caution

Do not ENABLE both PWSCPU00 and PWSCPU01. These two Rules schedule EXECs that use the same variables in EXEC MSU002C and are mutually exclusive.

Processing Flow

If either of the PWSCPU messages is received 3 or more times within 10 minutes, the CPU value in the message is compared against the appropriate CPU threshold variable. If the threshold is exceeded, an ALERT is issued and an additional monitor (\$TOTCPU or \$TSOCP, depending on which threshold is exceeded) is invoked.

The \$TOTCPU and \$TSOCP EXECs issue the RESOLVE CPU command to determine who is the current major user of the CPU (\$TSOCP limits the search to TSO users only). The EXECs then issue ALERTs to give the operator insight into which users are responsible for the exception condition.

When either the PWSCSA01 or PWSDEV01 message is received, the value in the message is compared against the appropriate threshold variable. If the threshold is exceeded, an ALERT is issued.

EXECs

The EXECs for the Exception Monitoring Solution are:

PWSCPU00	Monitors MAXCCPU threshold
PWSCPU01	Monitors MAXCCPU threshold
PWSCPU02	Monitors MAXTCPU threshold
PWSCSA01	Monitors MAXBCSA and MAXACSA thresholds
PWSDEV01	Monitors MAXDEVU threshold

Rules

The Rules for the Exception Monitoring Solution are:

PWSCPU00	Text-ID=PWSCPU00
PWSCPU01	Text-ID=PWSCPU01
PWSCPU02	Text-ID=PWSCPU02
PWSCSA01	Text-ID=PWSCSA01
PWSDEV01	Text-ID=PWSDEV01

The Rules are distributed DISABLED in RULESET AAORULM1. You must ENABLE these Rules to implement this solution.

Part 3. CICS Solutions

The distributed CICS Solutions fall into these functional categories:

- Error Recovery
- CICS Performance Management

The following chapters discuss each function, their automation subjects, and the distributed solutions.

Chapter 9. Error Recovery Solutions	41
Storage Violation Solution	42
Variables	42
Invocation	42
Processing Flow	42
EXECs	43
Rules	43
CICS Abnormal Termination Solution	44
Variables	44
Invocation	44
Processing Flow	44
EXECs	44
Rules	44
VSAM Subtask Abnormal Termination Solution	45
Variables	45
Invocation	45
Processing Flow	45
EXECs	46
Rules	46
CICS Temporary Storage Suspensions Solution (CICSTSS)	47
Variables	47
Invocation	47
Processing Flow	47
EXECs	48
Rules	48
Temporary Storage (TS) Data Set Extension Failure Solution	49
Variables	49
Invocation	49
Processing Flow	49
EXECs	49
Rules	50
Terminal Errors Solution	51
Variables	51
Invocation	51
Processing Flow	51
EXECs	51
Rules	52
Chapter 10. CICS Performance Management Solutions	53
File Degradation Analysis Solution	54
Variables	54
Invocation	54
Processing Flow	55

EXECs	55
Rules	56
File Allocation - Deallocation Solution	57
Variables and JCL	57
Invocation	57
Processing Flow	58
EXECs	58
Rules	58
VSAM Control Area Split Monitor Solution	59
Variables and JCL	59
Invocation	59
Processing Flow	59
EXECs	59
Rules	60
Transaction Response Time Monitor Solution	61
Variables and JCL	61
Invocation	61
Processing Flow	61
EXECs	61
Rules	62

Chapter 9. Error Recovery Solutions

Critical messages from CICS can be lost among the message traffic at the operator console. Also, replies to those messages and the follow-up actions taken might not be consistent.

The Error Recovery Solutions assist data center personnel when critical errors occur in the CICS environment.

BMC Software provides the following Error Recovery Solutions:

Storage Violation

Notifies a central operator of storage violations

CICS Abnormal Termination

Notifies a central operator of any CICS region abend

VSAM Subtask Abnormal Termination

Notifies a central operator of a VSAM subtask abend and automatically replies to the abnormal WTOR

Temporary Storage Data Set Extension Failure

Notifies a central operator and provides a follow-up display for investigating the problem

Terminal Errors

Resets terminal status after a terminal error occurs

Storage Violation Solution

The Storage Violation Solution notifies an operator when a storage violation has occurred.

Variables

For the Storage Violation Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables”](#) on page 161 for information on default settings and initializing the variables.

Invocation

This solution is invoked by the rule for the following text-ID:

DFH0508 A STORAGE VIOLATION HAS OCCURRED

Processing Flow

When the DFH0508 message is received, the ALERT:

```
*STORAGE VIOLATION IN CICS ci csi d, REPLY YES TO CANCEL
```

is sent to ask the operator to confirm cancellation of the CICS region that suffered the storage violation. If the operator replies yes, the region is cancelled and the message:

```
*STORAGE VIOLATION IN CICS ci csi d, BEING CANCELED NOW
```

is sent.

To modify the solution to automatically cancel the region without asking for operator confirmation, edit BBPROC member DFH0508. When automatic cancel is in effect, the ALERT BEING CANCELED NOW is sent.

EXECs

The EXECs for the Storage Violation Solution are:

DFH0508	Handles DFH0508 message
CER001C	Cancels the CICS system after a storage violation

Rules

The Rules for the Storage Violation Solution are:

DFH0508	Text-ID=DFH0508
DFHSM010	Test-ID=DFHSM010

The Rules are distributed DISABLED in RULESET AAORULC1. You must ENABLE these Rules to implement this solution. You must ENABLE DFHSM010 for CICS/ESA.

CICS Abnormal Termination Solution

The CICS Abnormal Termination Solution notifies an operator when an abend in a CICS region has been detected.

Variables

For the CICS Abnormal Termination Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables” on page 161](#) for information on default settings and initializing the variables.

Invocation

This solution is invoked by the rule for the following text-ID:

DFH0606 ABEND xxxx - xxxx HAS BEEN DETECTED

Processing Flow

In response to the DFH0606 message, an ALERT is sent to the FOCAL POINT target to indicate a catastrophic abend ended a CICS system.

EXECs

The EXEC for the CICS Abnormal Termination Solution is:

DFH0606 Handles DFH0606 message

Rules

The rule for the CICS Abnormal Termination is:

DFH0606 Text-ID=DFH0606

The rule is distributed DISABLED in RULESET AAORULC1. You must ENABLE this rule to implement this solution.

VSAM Subtask Abnormal Termination Solution

The VSAM Subtask Abnormal Termination Solution automatically notifies an operator when a VSAM subtask abends.

Variables

For the VSAM Subtask Abnormal Termination Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLYTPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables”](#) on page 161 for information on default settings and initializing the variables.

Invocation

This solution is invoked by the rule for the following text-ID:

DFH0901 VSAM SUBTASK ABEND - DO YOU WANT TO CONTINUE IN A DEGRADED MODE OR ABEND? REPLY GO OR CANCEL.

Note: This solution cannot be used for CICS/ESA.

Processing Flow

When the DFH0901 WTOR is received, this solution replies **G0** to let CICS continue initialization. The ALERT message:

```
*VSAM SUBTASK ABEND, CICS CONTINUING IN DEGRADED MODE
```

also is sent.

To modify the solution to reply **CANCEL** instead of **G0**, edit BBPROC member DFH0901. When automatic cancel is in effect, the solution replies **CANCEL** to the DFH0901 WTOR and issues the ALERT:

```
*VSAM SUBTASK ABEND, CICS BEING CANCELLED AUTOMATICALLY
```

EXECs

The EXEC for the VSAM Subtask Abnormal Termination Solution is:

DFH0901 Handles DFH0901 message

Rules

The rule for the VSAM Subtask Abnormal Termination Solution is:

DFH0901 Text-ID=DFH0901

The rule is distributed DISABLED in RULESET AAORULC1. You must ENABLE this rule to implement this solution.

CICS Temporary Storage Suspensions Solution (CICSTSS)

This solution detects and automatically corrects suspensions of CICS Temporary Storage processing caused when Auxiliary Temporary Storage data set is full.

When that is the case, CICS suspends processing for many transactions—not only transactions that are writing to temporary storage directly but many transactions (including system functions) that use temporary storage indirectly are also affected. For example, the EXEC CICS START command with the DATA parameter causes CICS to use temporary storage to hold the data being passed to the started task.

Variables

This solution does not use any specific variable requiring special initialization. However, variables describing the CICS systems, as explained in [Appendix D, “CICS Solution Variables” on page 161](#), should be set.

Invocation

The EXECs in this solution are scheduled when the following message is received:

```
FT068*          NNNNNN TEMPORARY STORAGE SUSPENSIONS HAVE  
                OCCURRED
```

Processing Flow

All EXECs in this solution are message-driven. The EXECs create ALERT messages that may require an operator reply to initiate a follow-up EXEC, or may include extended help or follow-up commands. The IND column of the Alert Detail Display shows which options are available for each ALERT.

If an ALERT message has an E in the IND column, a follow-up EXEC is assigned. The message text (or extended help panel) explains what reply is requested. You should enter the message text in the LC column of the ALERT display to invoke the follow-on EXEC.

If an ALERT message has a H in the IND column, an Extended Help Panel is assigned. You should type the EXPAND command and press ENTER (or use the EXPAND PFK, if assigned) after placing the cursor anywhere in the ALERT message text to display the help panel.

If an ALERT message has a C in the IND column, a follow-up Primary Command is assigned. You should type the TRANSFER command and press ENTER (or use the TRANSFER PFK, if assigned) after placing the cursor anywhere in the ALERT message text to invoke the command. The follow-up command transfers you to a display with more information to help solve the problem.

EXECs

The EXECs for the CICSTSS Solution are:

- FT068** Starts follow-up of temporary storage suspensions by scheduling CER004C and setting a timer for CER003C.
- CER003C** Executes the CICS MANAGER Problem Display and looks for the FT068 message. If found, it compares the count in the message to the count saved in a variable. If the count has increased, the investigation EXECs are kicked off.
- CER004C** Executes the CICS MANAGER Task Display and purges any tasks waiting on temporary storage processing (ATSP).
- CER005C** Executes the CICS MANAGER Temporary Storage Unit Table (TSUT) display and purges some Auxiliary TSUT entries. It first purges any entries that start with CEBR, which typically contain transaction dump output. If the TSP suspension persists, on the next iteration, CER005C purges the largest TSUT entry. The largest entry is determined by examining the Data Length and PUTQ count fields.
- CER006C** Handles the reply from the ALERT created by CER004C.

Rules

There is one rule for each of the messages listed in [“Invocation” on page 47](#). The rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULC1. You must ENABLE the following Rules to implement this solution: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

Temporary Storage (TS) Data Set Extension Failure Solution

The Temporary Storage (TS) Data Set Extension Failure Solution notifies a central operator and provides a follow-up display for investigating the problem.

Variables

For the Temporary Storage Data Set Extension Failure Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables” on page 161](#) for information on default settings and initializing the variables.

Invocation

This solution is invoked by the rule for the following text-ID:

DFH1311 TEMPORARY STORAGE DATA SET IS FULL AND CANNOT BE EXTENDED

Processing Flow

In response to DFH1311, an ALERT is issued. In addition, the CICS MANAGER TEMPSTOR command is associated with the ALERT. The operator can use the command to map the temporary storage usage.

EXECs

The EXEC for the Temporary Storage Data Set Extension Failure Solution is:

DFH1311 Handles DFH1311 message

Rules

The rule for the Temporary Storage Data Set Extension Failure Solution is:

DFH1311 Text-ID=DFH1311

The rule is distributed **DISABLED** in RULESET AAORULC1. You must **ENABLE** this rule to implement this solution.

Chapter 10. CICS Performance Management Solutions

The CICS Performance Management Solutions help data center personnel achieve maximum availability and increased response time.

BMC Software provides the following Performance Management Solutions:

File Degradation Analysis

Automatically analyzes factors causing poor response time for a CICS file

File Allocation - Deallocation

Lets you allocate, enable, and open, or lets you close, disable, and deallocate files from a target CICS system

VSAM Control Area Split Monitor

Notifies the operator of VSAM Control Area split problems

Transaction Response Time Monitor

Notifies the operator when transactions are exceeding response time objectives

File Degradation Analysis Solution

Most CICS response time problems are caused by I/O performance problems. High I/O service times, as reported by the CICS MANAGER Background Problem Service, are presented on various displays and logged to the BBI-SS Journal. Constant investigation of I/O problems can be very time consuming. Further, because most I/O contention problems are transient, factors causing a problem seem to disappear before an investigation is started.

This solution automatically analyzes the factors causing disk I/O performance problems that affect CICS response time.

Note: This solution requires RESOLVE services.

Variables

For the File Degradation Analysis Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, "CICS Solution Variables"](#) on page 161 for information on default settings and initializing the variables.

Invocation

This solution is invoked by a Rule for the following text-ID:

FT095W	FILE filename EXCEEDED SERVICE TIME, n.nnn IS AVERAGE SERVICE TIME
---------------	--

Processing Flow

When the FT095W message is received, contention for logical and physical resources in the system where the target CICS is executing is analyzed using CICS MANAGER and RESOLVE services. EXECs also are scheduled to look for any contention from other systems in the shared DASD environments. Results of the analysis are written in the MAINVIEW AutoOPERATOR Subsystem Log.

All results messages written to the MAINVIEW AutoOPERATOR Subsystem log are prefixed with a message-ID. The message-ID consists of a fixed literal (CAF#), followed by the EXEC sequence number (IMFEID); for example, CAF#1234. Use the message-ID to search for all messages from a given file analysis in the log, even when they arrive from different systems at different times and are intermixed with other messages in the SS Log.

The following messages are written to the MAINVIEW AutoOPERATOR Subsystem Log:

```
STARTING FILE DEGRADATION ANALYSIS FOLLOWING FT095W
FOR FILE filename IN CICS SYSTEM cicsid (smfid)
ON VOLUME volser ON UNIT ucb WITH nnnnn EXCPS.
```

```
DATASET dsname IS ALSO ALLOCATED TO JOB(S):
jobname1, jobname2, . . . . ON SYSTEM smfid.
```

```
NO OTHER JOBS ON SYSTEM smfid ARE USING DATASET dsname
IN CICS SYSTEM cicsid.
```

```
DEVICE ucb IS LESS THAN n% BUSY ON SYSTEM smfid.
```

```
DEVICE ucb WAS OBSERVED mnn% BUSY ON SYSTEM smfid
WITH AN I/O RATE OF nnn PER SECOND
AND AN AVERAGE WAIT TIME OF n.nnn MS.
```

```
nnn% OF THE ACTIVITY ON ucb IS FROM JOB jobname.
```

```
DEGRADATION ANALYSIS COMPLETE FOR FILE filename
ON SYSTEM smfid, RESULTS FROM OTHER SYSTEM(S)
WILL FOLLOW IN THE LOG WITH THE SAME MESSAGE ID.
```

EXECs

The EXECs for the File Degradation Analysis Solution are:

FT095W	As primary EXEC, schedules all secondary CPEnnnx EXECs that follow
CPE001C	Searches for the DD name corresponding to the filename when the file is an IMS data base
CPE002C	Reports on other jobs allocated to the CICS file data set name
CPE003C	Reports on other jobs using the device where the CICS file is located
CPE004C	Gets the data set name, volser number, and UCB for the CICS filename using the RESOLVE TIOT command

Rules

The Rule for the File Degradation Analysis Solution is:

FT095W Text-ID=FT095W

The rule is distributed **DISABLED** in RULESET AAORULC1. In addition to this Rule, the following Rules must be **ENABLED** to implement this solution: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

File Allocation - Deallocation Solution

The batch window (the interval between the time CICS is stopped and started again) might be too short to accommodate the necessary batch jobs to extract, update, and back up files used by CICS. If the window is too small, the operator must enter commands to close, disable, and free, and then later allocate, enable, and open files. This takes time, is tedious, and is prone to error. Syntax errors or a missed step in a complicated procedure can cause resources to be unavailable for use by CICS transactions.

This solution lets batch jobs allocate and deallocate files from a running CICS system. This lets files be freed from CICS, processed by other jobs, and returned to CICS, while CICS remains available for other users.

Variables and JCL

For the File Allocation - Deallocation Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables” on page 161](#) for information on default settings and initializing the variables.

Invocation

Most aspects of this solution are invoked from batch JCL. Edit two sample JCL members distributed in the BBUSER data set to specify file names and CICS region names.

CAOALLOC	Allocate, enable, and open a file in a CICS system from a batch job
CAODEALO	Close, disable, and free a file in a CICS system from a batch job

The remaining aspects of this solution are invoked by Rules for the following text-IDs:

FT401E	ALLOC FAILED RC=XX EC=XX FILE filename DSN dsname
FT402I	ALLOC COMPLETE FOR FILE filename DSN dsname
FT403E	DEALLOC FAILED NOT ALLOCATED FILE filename
FT404I	DEALLOC COMPLETE FOR FILE filename

Processing Flow

When the batch job CAOALLOC is run, the file you specified is allocated, enabled, and opened on the CICS system you specified.

When the batch job CAODEALO is run, the file you specified is closed, disabled, and freed on the CICS system you specified.

EXECs

The EXECs for the File Allocation-Deallocation Solution are:

CPEALLOC	Allocates, enables, and opens a file in a CICS system
CPEDEALO	Closes, disables, and deallocates a file in a CICS system
FT401E	Stores the result from IMFEXEC ALLOC in a shared variable
FT402I	Stores the result from IMFEXEC ALLOC in a shared variable
FT403E	Stores the result from IMFEXEC FREE in a shared variable
FT404I	Stores the result from IMFEXEC FREE in a shared variable

Rules

The Rules for the File Allocation-Deallocation Solution are:

FT401E	Text-ID=FT401E
FT402I	Text-ID=FT402I
FT403E	Text-ID=FT403E
FT404I	Text-ID=FT404I

The Rules are distributed DISABLED in RULESET AAORULC1. In addition to these Rules, you must ENABLE the following Rules: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

VSAM Control Area Split Monitor Solution

VSAM Control Area splits can cause I/O response times to be extended. This solution assists the operator in monitoring which files are suffering Control Area splits.

Variables and JCL

For the VSAM Control Area Split Monitor Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables” on page 161](#) for information on default settings and initializing the variables.

Invocation

This solution is invoked by a Rule for the following text-ID:

FT092S FILE filename HAS HAD nnn CONTROL AREA SPLITS

Processing Flow

In response to the FT092S message, an ALERT is sent, telling the operator what to do about a VSAM file with several Control Area splits:

*FT092S FILE file SHOULD BE RE-DEFINED (nnn C. A. SPLITS)

In addition, the CICS MANAGER FILEXPND command is associated with the ALERT. The operator can use the command to display the status of the VSAM cluster.

EXECs

The EXEC for the VSAM Control Area Split Monitor Solution is:

FT092S Handles FT092S message

Rules

The rule for the VSAM Control Area Split Monitor Solution is:

FT092S Text-ID=FT092S

The rule is distributed **DISABLED** in RULESET AAORULC1. In addition to this Rule, you must implement the following Rules: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

Transaction Response Time Monitor Solution

This solution ensures that operators are aware of transactions exceeding service level objectives.

Variables and JCL

For the Transaction Response Time Monitor Solution to function properly in your environment, you must establish values for the following variables:

FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs
ONLSYSN	Job name of the target system
ONLCNTN	Subsystem ID of the controlling BBI-SS
ONLALTN	Color for ALERT messages (can be BLUE, WHITE, RED, or GREEN)
ONLTYPN	Type of target or BBI-SS (can be CICS, DB2, IMS, CICS AO, or IMS AO)
ONLALMN	Indicates whether an alarm sounds (can be Y or N)

See [Appendix D, “CICS Solution Variables” on page 161](#) for information on default settings and initializing the variables.

Invocation

This solution is invoked by a Rule for the following text-ID:

FT094W TRAN tranid HAS EXCEEDED SERVICE LEVEL, nn.nnn IS AVG
REPSNSE

Processing Flow

In response to FT094W, the ALERT message:

*SERVICE LEVELS EXCEEDED, ENTER ON THIS MSG TO SEE ACTIVITY

is sent. In addition, the CICS MANAGER MONITOR command is associated with the ALERT. The operator can use the command to display service level information in the CICS region.

EXECs

The EXEC for the Transaction Response Time Monitor Solution is:

FT094W Handles FT094W message

Rules

The rule for the Transaction Response Time Monitor Solution is:

FT094W Text-ID=FT094W

The rule is distributed **DISABLED** in RULESET AAORULM1. In addition to this Rule, you must implement the following Rules: CICSTART, CICSTERM, FT426W, FT425W, and FT435I.

Part 4. DB2 Solutions

The distributed DB2 Solutions fall into three functional categories:

- Global operations
- Resource contentions
- DB2 performance management

The first chapter in this part discusses the customization steps needed to take advantage of the automation features of the DB2 Solutions.

The remaining chapters discuss the function, automation subjects, and the distributed solutions within each category.

Note: Several DB2 Solutions have access to the ALERT help facility of AutoOPERATOR. If you see the ALERT help indicator of **h** or **H** when using a DB2 Solution, then help panels exist for that ALERT.

Chapter 11. Initial Customization Steps	65
Steps	65
Chapter 12. DB2 Global Operations Solutions	69
DB2 Major Messages (DB2ALRT Solution)	69
Variables	69
Invocation	69
Processing Flow	71
EXECs	72
Rules	72
DB2 Environment Set-Up (DB2INIT Solution)	73
Variables	73
Invocation	74
Processing Flow	74
EXECs	74
Rules	74
DMR Major Messages (DMRALRT Solution)	75
Variables	75
Invocation	75
Processing Flow	76
EXECs	76
Rules	76
DMR Utilities (DMRUTIL Solution)	77
Variables	77
Invocation	77
EXECs	77
Rules	78
Chapter 13. DB2 Resource Contention Analysis Solutions	79
Deadlock/Timeout Analysis (DB2DLOK Solution)	79
Variables	79
Invocation	80
Processing Flow	80
EXECs	83
Rules	83

DB2 Table Space Filling Up (DB2TFUL Solution)	84
Variables	84
Invocation	85
Processing Flow	85
EXECs	85
Rules	85
DB2 Thread Control (DB2THRD Solution)	86
Variables	86
Invocation	86
Processing Flow	86
EXECs	86
Rules	87
Chapter 14. DB2 Performance Management Solutions	89
DB2 Runaway Query Control (DB2RNWY Solution)	89
Variables	89
Invocation	90
Processing Flow	90
EXECs	90
Rules	91
DB2 Response Time (DB2RESP Solution)	92
Variables	92
Invocation	92
Processing Flow	93
EXECs	93
Rules	93

Chapter 11. Initial Customization Steps

Before you can take advantage of the automation features of the DB2 Solutions, you must customize your automation environment to accommodate them. You can do this by implementing the solutions described in [Chapter 12, “DB2 Global Operations Solutions” on page 69](#). You can then successfully use the other DB2 Solutions. Or you can follow these steps to customize your automation environment to use the DB2 Solutions:

Steps

Follow these steps to customize your automation environment:

Step 1: Customize EXECs

Make these changes to the following EXECs:

EXEC	Change
DSU101C	Change the value specified for variable DB2 to the name of one of your DB2 subsystems.
DSU101C	Verify that the variables MSTR, DBM1, IRLM and DIST correctly identify the names of the started tasks for the DB2 you named. Note: For the initial install, you can accept the default values for the rest of the variables in this EXEC.

Step 2: Determine number of DB2 subsystems

You can monitor up to nine different DB2 subsystems with the DB2 Solutions. However, changing the number of DB2 subsystems that your BBI subsystem (BBI-SS) monitors requires that you also change the settings of some DB2 variables (See [Appendix E, “DB2 Solution Variables and Parameters” on page 163](#)).

If your BBI-SS is monitoring more than one subsystem, follow these steps:

1. Set the value of n in D_DB2NUMn to the number of DB2 subsystems in DSU001C.
2. Create as many copies of EXEC DSU101C as the number of subsystems you plan to monitor and name the copies as follows:
DSU101C, DSU102C. . . DSU10nC
where n equals the value of D_DB2NUM in step one.
3. Set each copy's variable to identify a unique DB2 subsystem.
4. Verify that the following variables' values correctly identify the names of the started tasks for this DB2 subsystem:
 - MSTR
 - IRLM
 - DBM1
 - DIST

Step 3: Update and activate rules

Rules are initially DISABLED with MAINVIEW AutoOPERATOR Solutions. You must enable the rules that affect the solutions you plan to use. In addition, you must update certain parameters in BBPARM to reflect the solutions rules you enabled.

To activate and update DB2 Solutions rules:

1. Activate the rules in each set of rules for the DB2 Solutions by using a text editor to change STATUS(DISABLED) to STATUS(ENABLED).
2. Enable the rule for message PM0010I found in rule set AA0RUL00 of the MAINVIEW AutoOPERATOR product.
3. Activate the following DB2 Solutions sets of rules from the AAO RULE PROCESSOR status control panel:
 - AAORULD1
 - AAORULD2
 - AAORULD3
 - AAORULD5
 - AAORULD6
 - AAORULD8
4. Update the RULESCAN and RULESET parameters in BBPARM member BBISSP00 to reflect the DB2 Solutions rules and make these sets of rules permanently active.

Note: A sample is provided in BBPARM member BBISSPD1.

Step 4: Add monitors and EXECs

- Add the monitors and time-initiated EXECs for the DB2 Solutions to your startup list, which is specified in BBPARM member BBISSP00 (as shown in the sample provided in BBPARM member BBISSPD1).
- Start the monitors and time-initiated EXECs from the AAO Time-Initiated EXEC panel by specifying:
BLK=DPE002B

and

BLK=DPE030B

Step 5: Determine threshold levels

When running DB2 applications, DB2 informational messages appear frequently on your console screen. For the DB2RNWY Solution (see [“DB2 Runaway Query Control \(DB2RNWY Solution\)” on page 89](#)), the frequency of a message's appearance is directly related to any threshold levels you set before using this solution:

- Browse hilevel.BBPARM(DMRBEX00) to review DB2RNWY threshold level parameters.
- Browse hilevel.BBPARM(DMRBEXD1) to review the BMC Software recommended settings for these parameters.
- Copy member DMRBEXD1 into member DMRBEX00 and set each threshold parameter in DMRBEXD1 to the value of your choice.

Note: If you have more than one DB2 subsystem, you must copy the thresholds for each DB2 TARGET statement.

- Put these changes into effect by issuing the command:

```
. E P DMRBEX00
```

from any command line.

Step 6: Modify your ALERT profile

To display the DB2 ALERT Queue on the ALERT Overview application, you must modify your ALERT Profile.

Use the primary command PROFILE and set the name of the ALERT Queue to match what you used for the shared variable D_ALERT_Q.

Chapter 12. DB2 Global Operations Solutions

The following DB2 Global Operations Solutions provide initialization routines as well as utilities and minimum alert management for DB2.

DB2ALRT Issues major DB2 messages as MAINVIEW AutoOPERATOR ALERTs

DB2INIT Provides information about the DB2 complex to other solutions through variables

DMRALRT Issues major DMR¹ messages as MAINVIEW AutoOPERATOR ALERTs

DMRUTIL Provides service routines that convert DMR services to local variables

DMR is a prerequisite for several DB2 Solutions as indicated in the individual solution description.

DB2 Major Messages (DB2ALRT Solution)

During its operations, DB2 issues a number of messages with various severity levels; some of these messages require immediate attention.

The DB2ALRT Solution transforms these messages, listed under “[Invocation](#)” into MAINVIEW AutoOPERATOR ALERTs, assigning a color to each message based on the severity level of the DB2 problem.

Variables

This solution does not require setting any variables.

Invocation

This solution is invoked by Rules for these major DB2 messages:

```
DSN3201I ABNORMAL EOT IN PROGRESS FOR USER= CON-ID= COR-ID=
DSNB200I UPDATE VVDS FAILED
DSNB204I OPEN OF DATA SET FAILED. DSNAME = dsn
DSNB207I DYNAMIC ALLOCATION OF DATA SET FAILED. REASON = rrr
        DSNAME = dsn
DSNB217I csect-name - ONLINE RECOVERY FOR AN INCONSISTENT PAGE
        WAS UNSUCCESSFUL FOR DBNAME = dbn, SPACENAME = spn,
        PAGE NUMBER = X' pno'
DSNB224I csect-name - BUFFER MANAGER I/O ERROR DURING function
        DBNAME=dbn, ...
DSNB225I BUFFER MANAGER I/O ERROR
DSNB226I BUFFER MANAGER DETECTED INVALID PAGE
DSNB227I DFHSM RECALL FAILED
DSNB551I BSDS READ ERROR
DSNB552I BSDS WRITE ERROR
```

¹ DMR refers to MAINVIEW for DB2

DSNB553I BSDS INSERT ERROR
 DSNB601I BUFFER POOL nn FULL
 DSNB602I UNABLE TO CREATE BUFFERPOOL
 DSNB603I INSUFFICIENT STORAGE FOR BUFFERPOOL EXPAND/CREATE
 DSNB605I INSUFFICIENT VIRTUAL STORAGE FOR BUFFERPOOL
 DSNB606I INSUFFICIENT STORAGE FOR HIPERPOOL
 DSNB607I UNABLE TO CREATE HIPERPOOL - NO EXPANDED
 DSNB608I UNABLE TO CREATE HIPERPOOL
 DSNB609I VIRTUAL BUFFERPOOL IS ZERO - DEFAULT USED
 DSNB610I UNABLE TO CREATE HIPERPOOL - NO ADMF
 DSNB611I HIPERPOOL DELETED - ADMF INACTIVE
 DSNCO01I CICS UR INDOUBT RESOLUTION IS INCOMPLETE FOR name
 DSNCO30E ERROR WRITING TO TRANSIENT DS
 DSNCO34I INDOUBT RESOLUTION FOR ur-id IS INCORRECT
 DSNCO35I INDOUBT RESOLUTION INCOMPLETE
 DSNCO36I INDOUBT RESOLUTION FOR ur-id IS INCONSISTENT
 DSNCO901I UNRECOVERABLE I/O ERROR IN DSNCCOMI
 DSNIO01I RESTART HAS BEEN DEFERRED
 DSNIO07I UNABLE TO DIRECT READ A LOG RECORD
 DSNIO10I BROKEN PAGE ACCESSED TYPE type NAME name MODNAME
 csect-name CONN-ID id CORR-ID id
 DSNIO12I PAGE LOGICALLY BROKEN TYPE type NAME name MODNAME
 modname ERQUAL erqual
 DSNJO04I ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE,
 ENDRBA=...
 DSNJO08E nn OF mm ACTIVE LOGS ARE FULL sname NEEDS ARCHIVE
 SCRATCH. REPLY YY WHEN DEVICE READY OR N TO CANCEL
 DSNJO13I TERMINAL ERROR ccc IN BUFFER rrr BEFORE ACTIVE LOG WRITE
 DSNJO14I TERMINAL ERROR ccc IN BUFFER rrr AFTER ACTIVE LOG WRITE
 DSNJO73I LOG ARCHIVE UNIT ALLOCATION FAILURE DETECTED,
 RETURN CODE = nnnn. ALLOCATION OR OFF-LOAD OF ARCHIVE
 LOG DATA SET MAY FAIL
 DSNJ100I csect-name ERROR OPENING BSDSn DSNAME=..., ERROR
 STATUS=...
 DSNJ102I LOG RBA CONTENT OF LOG DATA SET DSNAME=... STARTRBA=..
 ENDRBA=.. DOES NOT AGREE WITH BSDS INFORMATION
 DSNJ103I LOG ALLOCATION ERROR DSNAME=dsname, ERROR STATUS=eeeeiii
 DSNJ104I csect-name RECEIVED ERROR STATUS nnn FROM macro-name
 FOR DSNAME dsname
 DSNJ105I csect-name LOG WRITE ERROR DSNAME=..., LOGRBA=...,
 ERROR STATUS=ccccffss
 DSNJ106I LOG READ ERROR DSNAME=..., LOGRBA=...,
 ERROR STATUS=ccccffss
 DSNJ107I READ ERROR ON BSDS DSNAME=... ERROR STATUS=...
 DSNJ108I WRITE ERROR ON BSDS DSNAME=... ERROR STATUS=...
 DSNJ109I OUT OF SPACE IN BSDS DSNAME=...
 DSNJ110E LAST COPYn ACTIVE LOG DATA SET IS nnn PERCENT FULL
 DSNJ111E OUT OF SPACE IN ACTIVE LOG DATA SETS
 DSNJ115I OFFLOAD FAILED FOR ARCHIVE
 DSNJ117I INITIALIZATION ERROR READING BSDS DSNAME=...,
 ERROR STATUS=...
 DSNJ120I DUAL BSDS DATA SETS HAVE UNEQUAL TIME STAMPS, BSDS1
 SYSTEM=..., UTILITY=..., BSDS2 SYSTEM=..., UTILITY=...
 DSNJ124I OFFLOAD OF ACTIVE LOG SUSPENDED FROM RBA xxxx TO RBA
 yyyy DUE TO I/O ERROR
 DSNJ126I BSDS ERROR FORCED SINGLE MODE
 DSNJ150E LOG CAPTURE EXIT ABEND
 DSNL007I DDF IS ABNORMALLY TERMINATING
 DSNL033I DDF TERMINATION BECAUSE OF ABEND

DSNL400E INDOUBT THREAD HEURISTIC DAMAGE
 DSNL401E INDOUBT THREAD REMOTE ABORT - HEURISTIC DAMAGE
 DSNL402I INDOUBT THREAD REMOTE COMMIT - HEURISTIC DAMAGE
 DSNL403I INDOUBT THREAD REMOTE ABORT - HEURISTIC DAMAGE
 DSNL404E PROTOCOL ERROR
 DSNL405I THREAD PLACED INDOUBT
 DSNL406I THREAD MAY BE INDOUBT - COMM FAILURE
 DSNL408I INDOUBT THREAD HEURISTIC DAMAGE - COORDINATOR
 DSNL409I INDOUBT THREAD HEURISTIC DAMAGE - COORDINATOR
 DSNL411E COLD START BY COORDINATOR - MANUAL RESOLUTION
 DSNL412I PROTOCOL ERROR DURING SYNCHPOINT
 DSNL413I PROTOCOL ERROR DURING SYNCHPOINT
 DSNL414E PROTOCOL ERROR DURING INDOUBT
 DSNL420I COLD START BY PARTICIPANT - POSSIBLE DAMAGE
 DSNL421I SNA XLN PROTOCOL VIOLATION
 DSNL500I CONVERSATION FAILED TO LOCATION locname ...
 DSNL501I CNOS PROCESSING FAILED
 DSNL502I SYSTEM CONVERSATION FAILED TO LOCATION=locname ...
 DSNM002I IMS/V S xxxx DISCONNECTED FROM SUBSYSTEM yyyy RC=rc
 DSNM004I RESOLVE INDOUBT ENTRY(S) ARE OUTSTANDING FOR SUBSYS xxxx
 DSNM005I RESOLVE INDOUBT SYNCHRONIZATION PROBLEM WITH SUBSYS xxxx
 DSNP001I DSNPmmmm - dsn IS WITHIN n KBYTES OF AVAILABLE SPACE
 RC=r CONNECTION-ID=id CORRELATION-ID=id
 DSNP007I DSNPmmmm - EXTEND FAILED FOR dsn
 RC=r CONNECTION-ID=id CORRELATION-ID=id
 DSNP011I DSNPmmmm - MEDIA MANAGER SERVICES ERROR FOR dsn. MMRC=C
 DSMRC=r CONNECTION-ID=id CORRELATION-ID=id
 DSNP012I DSNPmmmm - ERROR IN VSAM CATALOG LOCATE FUNCTION FOR dsn
 CTLGRC=r CTLGRSN=r CONNECTION-ID=id CORRELATION-ID=id
 DSNP015I IRLM MANUAL UNLOCK FAILED
 DSNP028I HSM RECALL FAILED
 DSNT377I PLAN plan-id1 WITH CORRELATION ID id1 CONNECTION ID id2
 IS IN CONFLICT WITH AN INDOUBT THREAD
 DSNT500I csect-name RESOURCE UNAVAILABLE REASON r TYPE t NAME n
 DSNT501I csect-name RESOURCE UNAVAILABLE CORRELATION-ID id1
 CONNECTION-ID id2 REASON r TYPE t NAME n
 DSNV086E DB2 ABNORMAL TERMINATION REASON=xxxxxxxxx
 DXR013E ABEND UNDER IRLM TCB/SRB
 DXR016E BUCK PROCESS TIME EXCEEDS 30 SECONDS
 DXR019E STORAGE POOL DESTROYED
 DXR021E ABEND VTAM ERROR
 DXR022E SUBTASK FAILURE
 DXR023E INTERNAL OR MVS ERROR
 DXR024E VTAM REQUEST UNSUCCESSFUL
 DXR027A SESSION LOST
 DXR031E ABEND UNDER PTB SRB
 DXR050I OUT OF STORAGE DETECTED DURING DEADLOCK

Processing Flow

When any message listed under “[Invocation](#)” on [page 75](#) is received, an ALERT is issued. Color has been assigned to each ALERT based on the known severity of the attached message.

EXECs

There is one EXEC for this solution:

DPE992C Creates an ALERT for any message that invokes DPE992C

Rules

There is one rule for each of the messages listed under [“Invocation” on page 75](#).

The Rules are distributed DISABLED in RULESET AAORULD8. You must ENABLE these Rules to implement this solution.

DB2 Environment Set-Up (DB2INIT Solution)

Before you use the DB2 Solutions, you must provide some descriptive information about the DB2 subsystems you plan to use. This section describes how you provide this information to your automation environment.

Note: For additional information about customization steps affecting all solutions, see [Chapter 11, “Initial Customization Steps”](#) on page 65.

Variables

To use DB2INIT, you must establish values for both the general DB2 variables and the DB2 subsystem variables as follows:

General DB2 variables:

D_ALERT_TGT	Target to receive ALERTS
D_ALERT_Q	Name of the ALERT queue for all ALERTS generated by the solutions
D_ALERT_IDS	TSO IDs to receive notification of ALERTS through TSO SEND commands
D_ALERT_SEND	Indicates whether to issue the send command (Y/N)
D_ALERT_SENDOPT	MVS SEND option; either LOGON or NOW
D_DB2NUM	Number of DB2s defined to this BBI-SS

Set these variables by modifying EXEC DSU001C. For a complete list of all the DB2 Solutions variables, see [Appendix E, “DB2 Solution Variables and Parameters”](#) on page 163.

For the DB2 subsystems variables, one set of the following variables should be defined for each DB2 system you intend to control through solutions:

D_DB2n	nth DB2 target name
D_MSTRn	nth DB2 MSTR address space name
D_DBM1n	nth DB2 DBM1 address space name
D_IRLMn	nth DB2 IRLM address space name
D_DISTn	nth DB2 DIST address space name

Set these variables by modifying EXEC DSU10nC, where *n* is the number of the associated DB2 subsystem. If you start more than one DB2 subsystem, you must duplicate the sample EXEC DSU101C provided. Other EXECs, up to DSU105C, are included with only the comments box.

Note: If you try to display the DB2 variables names and content using the MAINVIEW AutoOPERATOR facilities, be aware that there might be several DB2 subsystems.

Invocation

As seen in “[Variables](#)” on page 73, DB2INIT consists of EXEC DSU001C and one or several EXECs DSU10 n C. You should ensure that these EXECs run as soon as possible after the host BBI-SS has been started.

This type of request usually is honored by having a Rule for PM0010I, which is the primary MAINVIEW AutoOPERATOR message. This rule should initiate some kind of start-up EXEC. You need to add the necessary calls to EXECs found in “[Variables](#)” on page 73 to that start-up EXEC, such as:

```
IMFEXEC SELECT EXEC(DSU001C) WAIT(Y)
```

Processing Flow

Solution DB2INIT is executed only once—at BBI-SS start-up. If you need to run DB2INIT dynamically (such as the first time, or after a variable change), you can call the processing EXECs from any MAINVIEW AutoOPERATOR workstation.

EXECs

The EXECs for the DB2INIT Solution are:

DSU001C	Initializes shared variables for solutions
DSU10nC nC	Initializes shared variables for a particular DB2 subsystem where n is the number of DB2 subsystems monitored by this BBI-SS

Rules

There is no rule for the DB2INIT Solution.

DMR Major Messages (DMRALRT Solution)

MAINVIEW for DB2 (DMR) is one of the major components of MAINVIEW. It provides DB2-related performance information for the system programmer and the Data Base Administrator.

Several DMR components issue messages with various severity levels. DMR experts have sorted out these messages and selected a set of those requiring immediate attention. In fact, only the messages reported as *severe exceptions* have been retained, along with their associated *clearing* message.

The solutions found within [Part 4, “DB2 Solutions” on page 63](#) were developed from a subset of those DMR messages—those considered to be candidates for automation.

The remaining messages are listed following “[Invocation](#)”. DMRALRT transforms the messages in this set into MAINVIEW AutoOPERATOR alerts, assigning a color to each message based on the severity level of the DMR problem.

Note: DMRALRT applies only when MAINVIEW for DB2 is installed.

Variables

For this solution to function properly in your environment, you must establish values for the general DB2 variables and the DB2 subsystems variables defined in the DB2INIT solution.

Invocation

This solution is invoked by Rules for these major DMR messages:

DZ1010S - INDOUBT THREAD
DZ1011I - EXCEPTION CLEARED: INDOUBT THREAD
DZ1040S - EDM POOL FULL FAILURES
DZ1041I - EXCEPTION CLEARED: EDM POOL FULL FAILURES
DZ1050S - BP(n) DM CRITICAL THRESHOLD REACHED
DZ1051I - EXCEPTION CLEARED: BP(n) DM CRITICAL THRESHOLD REACHED
DZ1060S - BP(n) IMMEDIATE WRITE THRESHOLD REACHED
DZ1061I - EXCEPTION CLEARED: BP(n) IMMEDIATE WRITE THRESHOLD REACHED
DZ1070S - BP(n) EXPANSION FAILURE, MAXPAGES REACHED
DZ1071I - EXCEPTION CLEARED: BP(n) EXPANSION FAILURE, MAXPAGES REACHED
DZ1080S - BP(n) EXPANSION FAILURE, VIRTUAL STORAGE SHORTAGE
DZ1081I - EXCEPTION CLEARED: BP(n) EXPANSION FAILURE, VIRT. ST.
SHORTAGE
DZ1090S - FINAL ACTIVE LOG DATASET 75% FULL
DZ1091I - EXCEPTION CLEARED: FINAL ACTIVE LOG DATASET 75% FULL
DZ1100S - ACTIVE LOG REDUCED TO SINGLE MODE
DZ1101I - EXCEPTION CLEARED: ACTIVE LOG REDUCED TO SINGLE MODE
DZ1110S - BSDS REDUCED TO SINGLE MODE
DZ1111I - EXCEPTION CLEARED: BSDS REDUCED TO SINGLE MODE
DZ1120S - SOS CRITICAL
DZ1121I - EXCEPTION CLEARED: SOS CRITICAL

Processing Flow

When any problem message is received, an ALERT is issued. Color has been assigned to each ALERT based on the known severity of the attached message. When any clearing message is received, the ALERT is cleared.

EXECs

There is one EXEC for this solution:

DPE994C Issues or deletes the ALERT associated with any of the DMR messages found in [“Invocation”](#)

Rules

There is one rule for each of the messages given in [“Invocation”](#). The rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULD3. You must ENABLE these Rules to implement this solution.

DMR Utilities (DMRUTIL Solution)

MAINVIEW for DB2 (DMR) is one of the major components of MAINVIEW. It provides DB2-related performance information for the system programmer and the Data Base Administrator (DBA).

The MAINVIEW architecture lets AutoOPERATOR and DMR communicate internally. This provides great power by letting the solutions investigate what is happening within DB2. This communication is based on an internal exchange of the regular DMR screen image in reply to a received command.

To shorten development time, service routines are provided for the major DMR services, which let you get direct DB2 data from local variables. These routines are included in the DMRUTIL Solution.

Note: DMRUTIL applies only when MAINVIEW for DB2 is installed.

Variables

This solution does not require any value to be set for variables.

Invocation

The EXECs from this solution are called by the various DB2 Solutions. Therefore, any DMR-based solution requires DMRUTIL.

EXECs

Following are the EXECs for the DB2UTIL Solution. Each utility EXEC is completed by an example on how to call it.

DPE002C	Support for BFRPL format in DB2 3.1
DPE014C	Start trace TYPE=D
DPE900C	DB2ST formatter — breaks DB2ST display into variables
DPE900D	Driver example for DPE900C and DPE901C
DPE901C	USERS formatter — breaks USERS display into variables
DPE901D	Driver example for DPE901C
DPE902C	Breaks DB2EX display into variables
DPE902D	Driver example for DPE902C
DPE903C	Breaks LOCKD display into variables
DPE903D	Driver example for DPE903C
DPE905C	BFRPL formatter — breaks BFRPL display into variables
DPE905D	Driver example for DPE905C
DPE908C	DBTS formatter — breaks DBTS display into variables
DPE908D	Driver example for DPE908C
DPE908DR	Driver for DPE908C (written in REXX)
DPE909C	IMFC call (IMAGE=NO)
DPE917C	Breaks BFRPL display into variables (DB2 release 2.3) and makes BFRPL variables compatible for DPE002C
DPE919C	Fixed column positions for DBTS display
DBE920C	DB2ST formatter
DPE930C	DB2ST formatter — breaks DB2ST display into variables (DB2 release 2.3)

DPE931C	USERS formatter — breaks USERS display into variables (DB2 release 3.1)
DPE932C	DLOGS formatter — breaks DLOGS display into variables (DB2 release 3.1)
DPE933C	DBTS formatter — breaks DBTS display into variables (DB2 release 3.1)
DPE935C	BFRPL formatter — breaks BFRPL display into variables (DB2 release 3.1)

Rules

There are no Rules attached to this solution.

Chapter 13. DB2 Resource Contention Analysis Solutions

The following DB2 Resource Contention Analysis Solutions concentrate on a few major resources that are critical to DB2 operations:

DB2DLOK	Analyzes deadlock or timeout situations
DB2TFUL	Provides an early warning of table spaces filling up
DB2THRD	Resolves IMS and CICS queued for thread situations

Deadlock/Timeout Analysis (DB2DLOK Solution)

A *timeout* occurs when the length of time a user has been waiting for a lock exceeds the IRLM timeout limit.

A *deadlock* occurs when two or more users have requested locks for two or more resources and each user has a lock that the other needs.

If these events frequently occur, consider redesigning the data structure or application.

For DB2DLOK, the PLAN that was waiting for a lock on a resource is called the *lock requestor*. The PLAN that held the lock is called the *lock owner*. The lock owner continues to run normally; the lock requestor receives a negative SQL return code.

Note: DB2DLOK requires MAINVIEW for DB2.

Variables

To use DB2DLOK, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

You should also define values for the following thresholds:

MAX_AGE	The age in minutes of the oldest event to be used for threshold comparison. For example, if MAX_AGE = 30, then an event that occurred 31 minutes ago is not used for threshold checking. This limits the problem analysis to recent history. To include all events that are saved in the history table, set MAX_AGE = 0. The default is MAX_AGE = 30.
MAX_OWNER	Triggers an ALERT if a PLAN is a lock owner more than MAX_OWNER times in MAX_AGE minutes. The default is 2, so a PLAN has to be a lock owner 3 or more times to trigger the ALERT.
MAX_REQ	Triggers an ALERT if a PLAN is a lock requestor more than MAX_REQ times in MAX_AGE minutes. The default is 2, so a PLAN has to be a lock requestor 3 or more times to trigger the ALERT.

MAX_EITHER Triggers an ALERT if a PLAN is involved in locking conflicts either as a lock owner or lock requestor more than MAX_EITHER times in MAX_AGE minutes. This threshold is checked only if the PLAN did not exceed the MAX_OWNER or MAX_REQ thresholds. The default is 3. For example, if a PLAN is a lockowner 2 times and a lock requestor 2 times, an ALERT is issued.

These variables should be set by reviewing EXECs DSU10nC, attached to initializing values pertaining to each specific DB2. These EXECs are included in the DB2INIT Solution.

Invocation

This solution is invoked by Rules for the following messages:

DSNT375I PLAN **plan-id1** WITH CORRELATION-ID **id1** CONNECTION-ID **id2** LUW-ID **id3** IS DEADLOCKED WITH PLAN **plan-id2** WITH CORRELATION-ID **id4** CONNECTION-ID **id5** LUW-ID **id6**

DSNT376I PLAN **plan-id1** WITH CORRELATION-ID **id1** CONNECTION-ID **id2** LUW-ID **id3** IS TIMED OUT DUE TO A LOCK HELD BY PLAN **plan-id2** WITH CORRELATION-ID **id4** CONNECTION-ID **id5** LUW-ID **id6**

DSNT501I Csect-name RESOURCE UNAVAILABLE CORRELATION-ID **id1** CONNECTION-ID **id2** LUW-ID **id3** REASON **reason** TYPE **type** NAME **name**

Processing Flow

The first part of this solution maintains a history table of deadlock and timeout events for each DB2 user. Each time a new event is added to the table, an analysis is done of the history table to see whether any PLAN has exceeded any threshold for contention.

History Table

By running EXEC DPE019C, you can display the deadlock history table at any time. This wrap-around table shows all previously recorded deadlock and timeout events. Once the table is full, a new event takes the place of the oldest event in the table. DPE019C writes two reports to the journal.

Report 1: The first DPE019C report is a display of all the events saved in the deadlock table. It starts with the most current event and ends with the oldest event:

```
.. DPE019C DEADLOCK/TIMEOUT HISTORY FOR DB2 FOLLOWS
```

DATE	TIME	TYPE	LOCK	PLAN	CONN-ID	CORR-ID
95.031	11:11:11	(1)	OWNR	(2)	(3)	(4)
			REQR	(5)	(6)	(7)
			NAME	(8)		

Numbered data items are:

- (1) Event type - either TIMEOUT or DEADLOCK
- (2) Plan name of lock owner
- (3) Connection-ID of lock owner
- (4) Correlation-ID of lock owner
- (5) Plan name of lock requestor
- (6) Connection-ID of lock requestor
- (7) Correlation-ID of lock requestor
- (8) Resource name in contention (from DSNT501I)

Report 2: The second report from DPE019C summarizes the events by PLAN and counts events above and below the MAX_AGE threshold:

```
.. DPE019C DEADLOCK/TIMEOUT SUMMARY FOR xxxx
.. DPE019C MAX_AGE = nnn MINUTES
```

DB2	PLAN	LOCKOWNR < MAX_AGE	LOCKOWNR TOTAL	LOCKREQR < MAX_AGE	LOCKREQR TOTAL
1	2	3	4	5	6

- (1) The target DB2 (whose subsys ID is xxxx)
- (2) The DB2 PLAN
- (3) The number of times the PLAN was a lock owner within the last &MAXAGE minutes
- (4) The total number of times the PLAN was a lock owner
- (5) The number of times the PLAN was a lock requestor within the last &MAXAGE minutes
- (6) The total number of times the PLAN was a lock requestor

You can use this report to adjust your thresholds.

ALERTs

If a threshold is exceeded, DB2DLOK creates an ALERT that details all the conflicting events for the plan in question and writes a message in the journal. You also can choose to notify a select group of TSO users.

From the ALERT, you can request a report of all deadlock events, request that a detail trace be started, or start a LOCKD data extractor EXEC. This is documented in the ALERT help panel.

Entering the EXT command in the ALERT RSP field invokes EXEC DPE015C. DPE015C examines the LOCKD display periodically and looks for lock contentions where this PLAN is a lock owner. When complete, a report of all contentions is written to the journal, followed by a summary by data base and table space.

- The variable STOPTIME determines how long the EXEC runs (default of 10 minutes, set in DSU10nC)
- The variable INTVL determines how long the EXEC waits between LOCKD displays (default of 5 seconds, set in DSU10nC)

Data Extractor

The LOCKD data extractor (EXEC DPE015C) writes two reports to the journal.

Report 1: The first report is a history report of all the events found:

DPE015C &DB2 LOCK EXTRACTION HISTORY FOR PLAN &PLAN FOLLOWS.
DPE015C &DB2 &MAX EVENTS FOUND IN THE LAST &STOPTIME MINUTES.

TIME	DB	TS	USERID	CNT	WAITER	TYPE
-----	-----	-----	-----	-----	-----	-----
1	2	3	4	5	6	7

- (1) HH:MM:SS that the locking conflict was observed
- (2) Data base in conflict
- (3) Table space in conflict
- (4) User ID owning the lock
- (5) Count of lock waiters
- (6) User ID of the first waiter
- (7) Waiter type

For a further explanation of the above fields, consult the *MAINVIEW for DB2 User Guide* for the LOCKD service.

Report 2: The second report from DPE015C is a summary of the above events by DB/TS.

DPE015C &DB2: LOCK EXTRACTION REPORT BY DB/TS FOR PLAN &PLAN
DPE015C OVER THE LAST &STOPTIME MINUTES FOLLOWS:

DATA BASE	TBLSPACE	# EVENTS
-----	-----	-----
1	2	3

- (1) Data base held in conflict by PLAN &plan
- (2) Table space held in conflict by PLAN &plan
- (3) Number of times this DB/TS was held in conflict by PLAN &plan

EXECs

The EXECs for the DB2 Deadlock/Timeout Solution are:

DPE010C	Determines DB2 target and plans involved in deadlocks or timeouts
DPE011C	Performs deadlock history update
DPE012C	Updates resource name in history table, then checks against thresholds; issues ALERTS/MSGS and optional MVS SEND commands
DPE013C	Follow-up EXEC for ALERT
DPE014C	Starts trace
DPE015C	Analyzes LOCKD display
DPE019C	Reports deadlock history table to journal

Rules

The Rules for the DB2 Deadlock/Timeout Analysis Solution are distributed in member AAORULD1:

DSNT375I	Text-ID=DSNT375I, distributed DISABLED
DSNT376I	Text-ID=DSNT376I, distributed DISABLED
DSNT501X	Text-ID=DSNT501I, distributed DISABLED, applied to message DSNT501I only when the CSECT name is DSNILMCL.

The Rules are distributed DISABLED in RULESET AAORULD1. You must ENABLE these Rules to implement this solution.

DB2 Table Space Filling Up (DB2TFUL Solution)

Many users access DB2 data bases during the day, and some users might continue to add data until they run out of space.

When DB2 reports that all the space has been used up, it is too late to avoid an outage. DB2TFUL provides an early warning of a table space or a partition filling up.

DB2TFUL analyzes open table spaces and imposes user-specified thresholds on them. If a threshold is exceeded, an ALERT is issued.

Looking at many DB2 data bases takes time and resources, so the attached analysis is run only every half-hour, through a timer request you set in MAINVIEW AutoOPERATOR.

Note: DB2TFUL requires MAINVIEW for DB2.

Variables

To use DB2TFUL, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

Additionally, you can define values for the following thresholds:

MAX_EXTS	The upper bound for the number of extents for a tablespace unless overridden by an exception. If zero, this threshold is ignored.
MAX_UTIL	The upper bound for the percent utilization for a tablespace unless overridden by an exception. If zero, this threshold is ignored.
NUM	Number of exceptions for this DB2. Must be equal to the highest suffix for variable NAME.
NAMEn	The name of the n th exception. Must be specified as a tablespace. Can be generic.
MAX_EXTSn	The extent threshold for the n th exception name. If zero, this threshold is ignored.
MAX_UTILn	The utilization threshold for the n th exception name. If zero, this threshold is ignored.

The above variables should be set by reviewing EXEC DSU10nC, attached to initializing values pertaining to each specific DB2 subsystem. These EXECs are included in the DB2INIT Solution.

For example, suppose the following variables are set in EXEC DSU101C:

```
SET MAX_EXTS = 0
SET MAX_UTIL = 0

SET NUM      = 2

SET NAME1    = &STR(DSN*)
SET EXTS1    = 15
SET UTIL1    = 0

SET NAME2    = &STR(DSNDB06_SYS*)
SET EXTS2    = 5
SET UTIL2    = 90
```

Setting both MAX_EXTS and MAX_UTIL to zero tells the solution to check only the exception thresholds.

The first exception states that any tablespace in a data base starting with DSN that has more than 15 extents creates an ALERT.

The second exception states that an ALERT is generated for any tablespace starting with SYS in data base DSNDB06 that has more than 5 extents or is over 90% utilized on the fifth extent.

Invocation

DB2TFUL usually is initiated through a time-initiated request. A sample timer is provided in BBPARM member DPE030B. You also should create a call to this timer in your BBIISPxx member. A sample entry is provided in BBPARM member BBIISPD1.

Processing Flow

This solution is intended to run in the background, using the timer facility described previously in “[Invocation](#)”, and to sample data periodically. However, both DPE030C and DPE031C can be executed from any command line of any MAINVIEW screen display by using the command character % (percent sign).

EXECs

The EXECs for the DB2TFUL Solution are:

DPE030C	Threshold analysis of the DBTS display for a particular DB2
DPE031C	The EXEC that calls EXEC DPE030C for each defined DB2

Rules

There are no Rules for this solution.

DB2 Thread Control (DB2THRD Solution)

DB2 offers its users a number of access paths, referred to as *threads*. DSNZPARM specifies the maximum number of threads DB2 can have. When this limit is reached, DB2 queues each request until a thread is available. Once this queuing process is initiated, it causes delays to users.

For transaction-oriented systems such as IMS and CICS, waiting for a DB2 thread can degrade the throughput of the entire system and should be avoided.

In a data center using DB2, users often encounter a shortage of threads. In response to this situation, the DB2THRD Solution attempts to identify a TSO thread that can be made available for IMS or CICS to use.

Note: DB2THRD requires MAINVIEW for DB2.

Variables

To use DB2THRD, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

Additionally, you should define the following variable:

Q_VERIFY Indicates (Y/N) whether to issue an ALERT to verify the TSO cancel.

Invocation

This solution is invoked by Rules for these major DMR messages:

DZ1020S - IMS TASK(S) QUEUED FOR THREAD
DZ1021I - EXCEPTION CLEARED: IMS TASK(S) QUEUED FOR THREAD
DZ1030S - CICS TASK(S) QUEUED FOR THREAD
DZ1031I - EXCEPTION CLEARED: CICS TASK(S) QUEUED FOR THREAD

Processing Flow

If **Q_VERIFY = Y**, then an ALERT is created when the message DZ1020W or DZ1030W appears. This ALERT has an extended help panel (DPE040A) associated with it that can be modified by your site to give the operator-specific instructions for this situation. When the queued for thread condition clears, the solution intercepts messages DZ1021I and DZ1031I and deletes the ALERT.

If **Q_VERIFY = N**, then EXEC DPE041C is invoked to identify the TSO user with the shortest elapsed time. EXEC DPE040C then cancels that TSO user so that a thread can be freed up for CICS or TSO to use. A message is written to the journal identifying the canceled TSO user.

EXECs

The EXECs for the DB2THRD Solution are:

DPE040C Analyzes message it has been scheduled from
DPE041C Analyzes DMR USERS service output to locate the TSO user to be canceled

Rules

There is one rule for each of the messages listed under “[Invocation](#)”. The rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULD6. You must ENABLE these Rules to implement this solution.

Chapter 14. DB2 Performance Management Solutions

The following DB2 Performance Management Solutions help data center personnel achieve maximum availability and increased response time:

DB2RNWY Reacts to DMR¹ messages for CICS, IMS, TSO, Batch, CAF, and Utilities runaway queries

DB2RESP Uses DMR services to identify potential causes of poor DB2 response time

DMR is a prerequisite for several DB2 Solutions as indicated in the individual solution description.

DB2 Runaway Query Control (DB2RNWY Solution)

Runaway queries are one of the major problems affecting DB2 performance. MAINVIEW for DB2's dynamic and static SQL calls and warning capabilities made it possible to develop the DB2RNWY Solution, which provides selective cancel capabilities over such runaway queries.

This solution requires:

- MAINVIEW for DB2
- MAINVIEW AutoOPERATOR for IMS and IMS RESOURCE ANALYZER (to cancel a runaway IMS query)
- CICS MAINVIEW AutoOPERATOR (to cancel a runaway CICS query)

Note: DB2 provides limited protection from runaway queries with the Resource Limit Facility (RLF), available with DB2 Version 2.1. However, this applies only to dynamic SQL. Moreover, the result from RLF is a query that has stopped with an SQL error code (-905 expected).

Variables

To use DB2RNWY, you must establish values for the global DB2 variables defined in the DB2INIT Solution.

¹ DMR refers to MAINVIEW for DB2

Invocation

This solution is invoked by Rules for the following messages:

```
DZ0610W RUNAWAY IMS <threshold> PST= TRAN= USER= J= TYPE= CRGN=  
DZ0611I EXCEPTI ON CLEARED: RUNAWAY IMS ...  
DZ0620W RUNAWAY CICS TRANSACTI ON USER=u <threshold> J= TRAN= TASK=  
DZ0621I EXCEPTI ON CLEARED: RUNAWAY CICS TRANSACTI ON ...  
DZ0630W RUNAWAY TSO QUERY USER= <threshold exceeded>  
DZ0631I EXCEPTI ON CLEARED: RUNAWAY TSO QUERY ...  
DZ0640W RUNAWAY BATCH USER= <threshold exceeded> J=  
DZ0641I EXCEPTI ON CLEARED: RUNAWAY BATCH ...  
DZ0650W RUNAWAY CAF, USER= <threshold exceeded> J=  
DZ0651I EXCEPTI ON CLEARED: RUNAWAY CAF ...  
DZ0660W RUNAWAY UTI LITY FUNCTI ON, USER= <threshold exceeded> J=  
DZ0661I EXCEPTI ON CLEARED: RUNAWAY UTI LITY ...
```

Processing Flow

When MAINVIEW for DB2 detects a runaway query, as determined by thresholds set in BBPARM member DMRBEX00, a message is sent to the LOG and filtered to be echoed as an ALERT on the MAINVIEW centralized screen. A sample member named DMRBEXD1 is provided in the BBPARM data set.

Operators must confirm the cancel of a runaway query job by entering CAN in the response field for the ALERT; the ALERT is cleared when the operator cancels the runaway query job, or when the runaway condition ends.

If the runaway query is:

- A job or TSO user, the cancel is done through the appropriate MVS cancel command.
- An IMS region, the solution verifies that the region name, number, and trancode match those in the warning message and that the region status is ACTV-USR or ACTV-DB2. It then issues a /STOP REG ABDUMP command followed by a /STOP REGION CANCEL command.
- A CICS transaction, the solution issues a KILL TASK xxxx FORCE.

Note: This solution does not cancel a runaway query automatically. It always requests operator agreement by the ALERT.

EXECs

The EXECs for the DB2 Runaway Query Control Solution are:

DPE020C	Create ALERTs for runaway query
DPE021C	Delete ALERTs for runaway query
DPE022C	Schedules MVS cancel EXEC DPE024C
DPE023C	Schedules IMS cancel EXEC DPE025C
DPE024C	Performs MVS cancel
DPE025C	Performs IMS cancel
DPE026C	Schedules CICS cancel EXEC DPE027C
DPE027C	Performs CICS cancel

Rules

The Rules are distributed DISABLED in RULESET AAORULD2. You must ENABLE these Rules to implement this solution.

DB2 Response Time (DB2RESP Solution)

Because transaction response time for all CICS, IMS, and TSO users relies on DB2's ability to reply to requests in the shortest possible time, it is important to obtain data about DB2 response time as early as possible.

This solution examines several MAINVIEW for DB2 (DMR) services for potential performance problems by comparing the observed values to user-defined thresholds and by reporting any observations that are above threshold levels.

DB2RESP is initiated by certain warning messages for elapsed and CPU time monitors or invoked directly by the operator.

The usefulness of this solution depends in large part on the thresholds chosen, so these must be chosen carefully. BMC Software has supplied what seem to be reasonable defaults with this solution, but you may want to make some modifications to fit your installation.

In the process of determining your installation thresholds, you should become aware of the DMR data items that are good performance indicators for your system.

This solution requires the DB2INIT and DMRUTIL Solutions.

Variables

To use DB2RESP, you must establish values for the general DB2 variables and the DB2 subsystem variables defined in the DB2INIT Solution.

This solution makes extensive use of thresholds for many domains of DB2 performance. For each DB2 subsystem, up to 35 important figures are analyzed. You can change any of the default thresholds. They are available for each DB2 system, and are located in the DSU10nC EXECs.

For more information on setting these threshold parameters, see [Appendix E, "DB2 Solution Variables and Parameters"](#) on page 163.

Invocation

This solution is invoked by Rules for these DMR messages:

```
DW0120W (nn) hh:mm: AVG ELAPSED TIME(parm) = nnn (>ppp) *****  
DW0150W (nn) hh:mm: AVG ELAPSED IN DB2(parm) = nnn (>ppp) *****  
DW0180W (nn) hh:mm: AVERAGE CPU USED(parm) = nnn (>ppp) *****  
DW0190W (nn) hh:mm: AVERAGE CPU IN DB2(parm) = nnn (>ppp) *****
```

where:

```
parm  
  {ALL|IMS|CICS|TSO}
```

To get these messages out of MAINVIEW for DB2, you must activate some monitors as listed in member DPE002B of the BBPARM data set. This is done in BBPARM member BBIISPxx using the TARGET statement, as shown in BBIISPD1 of BBPARM. For example, to set the DB2 monitors, specify:

```
TARGET=DB2D, BLK=DPE002B, USRID=xxxxxxx
```

You should adjust the WVAL values to values appropriate to your installation. Information about WVAL values can be found in the “Set Timer Request” chapter of the *MAINVIEW for DB2 User Guide*.

DB2RESP processes warning messages with an identifier of IMS, CICS, TSO, or ALL. These identifiers are required and cannot be modified.

Processing Flow

Occurrence of the DMR messages listed under “[Invocation](#)” on page 92 invokes DB2RESP, or you can invoke DB2RESP from any MAINVIEW command line by entering:

```
%DPE002C db2name
```

Here, db2name is the name of the target DB2.

Online help is available by entering:

```
%DPE002C ?
```

The positional parameters for DPE002C are:

PARM1	DB2 TARGET (REQUIRED)
PARM2	OUTPUT DESTINATION (OPTIONAL)DEFAULT = DB2 TARGET
PARM3	CONNECTION TYPE (OPTIONAL)

Values can be: ALL(default), IMS, CICS, or TSO.

EXECs

The EXECs for the DB2RESP Solution are:

DPE002C	Performs checking of thresholds for DB2
DPE005C	Drives DPE002C from DB2 monitor messages
DPE007C	Governor for solution-driven from monitor warning message

Rules

There is one rule for each of the messages listed under “[Invocation](#)” on page 92. The rule-IDs equal the message-IDs.

The Rules are distributed DISABLED in RULESET AAORULD5. You must ENABLE these Rules to implement this solution.

Part 5. Communication Solutions

The Communication solutions fall into four functional categories.

- E-mail
- SNMP
- Simple Network Paging Protocol
- AutoOPERATOR Web

The following chapters discuss each function, their automation subjects, and the communication solutions.

Chapter 15. E-mail Solution	97
EXECs	97
Chapter 16. SNMP Solution	101
EXECs	101
Chapter 17. Using the Paging Sample	105
EXECs	105
QAOSNPP1	105
QAOSNPP2	105
Processing Flow	107
Variables Returned from QAOSNPP2	108
Chapter 18. Using the AutoOPERATOR Web Sample	111
Introduction	111
What AutoOPERATOR Web Is	111
How AutoOPERATOR Web is Used	112
What the Sample Web Pages Look Like	112
Prerequisites	112
What You Need to Know	112
Warning about Case Sensitivity	113
What Information to Gather	113
What Software Must Be Installed	114
Installing and Customizing the Samples	114
Running the Installation EXEC	114
Configuring the HTTP Server	115
Making BBSAMP Available to AutoOPERATOR	115
Meeting AOAnywhere Requirements	115
Reviewing Security	116
Accessing Multiple Systems	116
Accessing the Samples	116
Viewing the Entry Page	117
Viewing the About Page	118
Viewing the CSM Page	119
Viewing the ALERTs Page	122
Viewing the Automation Reporter Page	124
Viewing the Commands Page	126
Debugging the Distributed CGI Scripts	128
Understanding CGIs	128

Debugging the CGI Script in a TSO Environment	128
Debugging the CGI Script without HTML Decoding	128
Avoiding Frames in the Display	128
Displaying the HTML Source in a Frame	129
Refreshing HTML Source Code	129
Common Error Messages	129

Chapter 15. E-mail Solution

MAINVIEW AutoOPERATOR provides a pair of sample EXECs that can be used to send an e-mail from a REXX EXEC.

EXECs

The EXECs for the e-mail Solution are:

QAOSMTP1 Resides in BBSAMP.

QAOSMTP2 Resides in BBPROC.

Note: Browse the AutoOPERATOR product libraries to view the most current version of these sample EXECs. BMC Software might ship PTFs at any time to enhance them.

This SMTP sample uses REXX sockets, which are part of the IBM TCP/IP product.

QAOSMTP1

QAOSMTP1 is a sample EXEC that demonstrates how you can pass e-mail data to QAOSMTP2, which performs the actual e-mail transmission. Most customers will be able to use QAOSMTP2 without modification.

To send an e-mail, make a copy of QAOSMTP1 and store it in UBBPROC using a new meaningful name. Replace the data in the user-input fields with appropriate data for your installation.

You can also incorporate the logic in QAOSMTP1 into your own REXX application to generate e-mails.

The areas that need to be modified in QAOSMTP1 are listed in the following table.

Name	Description
SMTP_SERVER	Local mail server. This server distributes mail to the POP3 and Exchange mail servers.
SMTP_REPLYTO	The e-mail address that receives replies when responding to this e-mail. It must be a valid address (or example, YourName@your-company.com).
Sender of the e-mail	The first line queued (your name), which might be different from the REPLYTO e-mail ID. Valid formats are dependant upon the mail server being used. Most mail servers accepts '<user@your-company.com>'. Other typical formats that are accepted are '"your name"' and '"your name"<id@bid.company.com>'. Check your SMTP documentation for acceptable values.
Recipient(s)	A blank delimited list. It is queued second. Do not use the form "name"<user@big.company.com>.

Note: A sender's name, subject, and at least one line of message text are required.
 Keep line lengths reasonable so that the e-mail has a readable format. The utility EXEC (QAOSMTP2) inserts a CrLf after each text line in the generated e-mail.

A QAOSMTP1 sample EXEC follows:

```

/* rexx qaosmtp1 */                                00010002
/* DOC GROUP(AO)  FUNC(SAMPLE)  AUTHOR(BMC)        */00020002
/* DOC DESC(sample EXEC for sending email via a SMTP server) */00030002
                                                    00040002
                                                    00050001
/*-----*/                                       00060001
* Sample EXEC which demonstrates sending an e-mail via a SMTP mail * 00070005
*   server.                                                    * 00080005
*                                                            * 00090005
* This EXEC constructs the e-mail data then calls QAOSMTP2 which * 00100005
*   reformats the data and sends it using the SMTP protocol.   * 00110005
*                                                            * 00120005
* QAOSMTP2 talks to the SMTP server at port 25 using IBM REXX * 00130005
* Sockets. NOTE: QAOSMTP2 resides in BBPROC.                  * 00131005
*                                                            * 00140005
* Change Log                                                  * 00150005
* 9-mar-01 bpo4962 Use queue command instead of push          * 00159005
* 9-feb-01 bpo4874 shipped to make bpo4874 prereq bpo4851     * 00159105
* 30-jan-01 bpo4851 update comments to note QAOSMTP2 is in BBPROC * 00159205
* 27-aug-99 bpo4319 initial release                          * 00160005
*                                                            * 00170005
-----*/                                         00180001
                                                    00190001
trace n                                             00200001
"IMFEXEC MSG '.QAOSMTP1 EID' IMFEID "Started'"          00210001
                                                    00211005
address mvs newstack                               /* make a new stack */ 00212005
                                                    00220001
/*-----*/                                       00230001
*                                                            * 00231005
* ----- Beginning of lines requiring changes ----- * 00232005
*                                                            * 00233005
* The external data queue is used to communicate the message text, * 00240005
*   subject and sender's name to the utility EXEC (qaosmtp2).  * 00250005
*                                                            * 00260005
* smtp_server is your local SMTP server. This server will distribute * 00261005
*   mail to the POP3 and exchange mail servers. Many mail servers * 00262005
*   will have "mail" as the first part of their network name but * 00262105
*   is not required to follow this standard.                  * 00262205
*                                                            * 00262305
* smtp_replyto is the Email address that will receive replies when * 00262405
*   responding to this Email. You should have a valid address here. * 00263005
*                                                            * 00264005
* Sender of the Email is the first line queued. It will be your * 00270005
*   name which may be different from the ReplyTo e-mail id.    * 00280005
* The formats allowed are dependant upon the mail server in use. * 00290005
* '<user@your-company.com>' is accepted by most mail servers. Some * 00291005
* other typical formats that are accepted '"your name"' and * 00291105
* '"your name"<id@bid.company.com>'. Check your SMTP documentation * 00291205
*   for acceptable values.                                    * 00292005
*                                                            * 00293005
* Recipient(s) is a blank delimited list. It is queued second. * 00300005
*   Do not use the form "name"<user@big.company.com>.         * 00320005
*                                                            * 00321005
* Subject line is the third queued line.                    * 00330005
*                                                            * 00370005
* Email message text will be the remainder of lines queued.  * 00371005
*                                                            * 00372005
* There must be a sender's name, a subject and at least one line of * 00380005
*   message text.                                           * 00390005

```

```

* 00400005
* Keep line lengths reasonable so that your e-mails will look nice. * 00410005
* 00420005
* The utility EXEC will insert a CrLf after each text line in the * 00430005
* generated e-mail. * 00440005
* 00443005
*-----*/ 00450005
smtp_server = 'mail.your-company.com' /* your SMTP mail server */ 00460005
smtp_replyto = 'YourName@your-company.com' /* Reply-to Email id */ 00470005
00471004
queue '<'strip(imforgn) '_AutoOPERATOR@your-company.com>' /* sender */ 00472005
00475004
queue 'user1@xxx.com user2@yyy.com group1@zzz.com' /* recipients */ 00478005
00480001
queue 'Email subject' /* Emails subject line */ 00520005
00520104
queue 'Email text line 1' /* Email body */ 00520205
queue 'Email text line 2' 00520305
queue 'Email text line 3' 00521005
00660105
/*----- 00660205
* 00661005
* ----- End of required changes ----- * 00662005
* 00663005
*-----*/ 00664005
00670001
smtp_debug = 'n' /* optional debug parm */ 00680001
00690001
smtp_error = 0 /* preset our error flag */ 00700001
00710001
if words(smtp_server) /= 1 then 00720001
do 00730001
"IMFEXEC MSG '.QAOSMTP1 EID" IMFEID "Error, blanks not allowed in",
"smtp_server="smtp_server"" 00740001
smtp_error = 1 00750001
00760001
end 00770001
00780001
if words(smtp_replyto) /= 1 then 00790001
do 00800001
"IMFEXEC MSG '.QAOSMTP1 EID" IMFEID "Error, blanks not allowed in",
"smtp_replyto="smtp_replyto"" 00810001
00820001
smtp_error = 1 00830001
00840001
end 00850001
00860001
if smtp_error = 1 then 00870001
do 00880001
"IMFEXEC MSG '.QAOSMTP1 EID" IMFEID "Ended with error(s)'" 00880001
"IMFEXEC EXIT CODE(8)" 00890001
exit 00900001
00910001
end 00920001
00930001
/* call the utility EXEC to send the e-mail */ 00940001
call qaosmtp2 smtp_server smtp_replyto smtp_debug 00950001
00960001
exec_rc = result /* save qaosmtp2 rc */ 00970001
address mvs delstack /* delete our stack */ 00980001
00990001
if exec_rc /= 0 then /* clean up and exit */ 01000001
"IMFEXEC MSG '.QAOSMTP2 ended with rc=" exec_rc"" 01010001
01020001
"IMFEXEC MSG '.QAOSMTP1 EID" IMFEID "Ended" " 01030001

```


Chapter 16. SNMP Solution

MAINVIEW AutoOPERATOR provides a pair of sample EXECs that can be used to generate SNMP Traps.

EXECs

The EXECs for the SNMP Solution are:

QAOSNMP1 Resides in BBSAMP.

QAOSNMP2 Resides in BBPROC.

Note: Browse the AutoOPERATOR product libraries to view the most current version of these sample EXECs. BMC Software might ship PTFs at any time to enhance them.

This SNMP sample uses REXX sockets, which are part of the IBM TCP/IP product.

QAOSNMP1

QAOSNMP1 is a sample REXX EXEC that demonstrates how to generate a SNMP Trap. The actual Trap generation logic is encapsulated in QAOSNMP2, which should be usable without modification by most customers.

To generate an SNMP Trap, copy QAOSNMP1 into UBBPROC using a new meaningful name. Replace the data in the user-input fields with the appropriate data for your installation.

You can also incorporate the logic in QAOSNMP1 into your own REXX application to generate SNMP Traps.

SNMP Traps are directed to SNMP Managers. Examples of SNMP Managers are PATROL[®] Enterprise Manager, Tivoli Enterprise Console, etc. You must coordinate your efforts with the administrators of your SNMP Manager. Consult them for any SNMP values of which you are unsure.

The user-input fields are as follows:

Name	Description
Community	A text string, usually used as a simple security mechanism.
SnmpManager	The IP address or host name of your SNMP Manager.
SnmpPort	The listening port number used by your SNMP Manager.
Enterprise	An object identifier that identifies the device that generates the Trap. Note: Nodes must be less than 128.
GenericTrap	An integer, specified from the SNMP-defined values for a generic Trap.

Name	Description
SpecificTrap	An integer, specified for a specific Trap.
TimeTicks	An integer; it usually represents the device uptime in hundredths of a second.
Object Identifier	An unique identifier for this event Trap. Note: Nodes must be less than 128.
OIDtype	The type of data; QAOSNMP2 supports Octet String (4) only.
OIDtext	Trap text.

A QAOSNMP1 sample EXEC follows:

```

/* rexx */ 00010000
/* */ 00020000
/* Sample EXEC which calls QAOSNMP2 to issue a */ 00030000
/* SNMP v1 Trap */ 00040000
/* */ 00050000
/* Nearly all parameters are passed to QAOSNMP2 */ 00060000
/* as positional parameters. QAOSNMP2 resides */ 00070001
/* in BBPROC. */ 00071001
/* */ 00080000
/* Community and OIDtext are passed on the data */ 00090000
/* stack. */ 00100000
/* */ 00110000
/* Note: Object Identifiers must begin with 1.3.x */ 00120000
/* Subsequent nodes must be less than 128 */ 00130000
/* eg. valid = 1.3.6.2.4 */ 00140000
/* invalid = 2.9.1 */ 00150000
/* invalid = 1.3.200 */ 00160000
/* */ 00170001
/* change log: */ 00180001
/* 9feb2001 bpo4874 shipped to make bpo4874 */ 00188002
/* prereq bpo4851 */ 00188102
/* 30jan2001 bpo4851 update comments to note */ 00188202
/* QAOSNMP2 is in BBPROC */ 00189001
/* 14apr2000 bpo4467 sample created */ 00190001
/* */ 00200001
/* */ 00210000
/* */ 00220000
/* this is a string ... primitive SNMP v1 security */ 00230000
Community = 'public' 00240000
/* */ 00250000
/* this can be any valid IP address or host name */ 00260000
/* eg. 132.60.1.3 or manager.bigcompany.com */ 00270000
SnmpManager = 'snmpMgr.BigCompany.com' 00280000
/* */ 00290000
/* this can be any valid port number (integer) */ 00300000
/* normally it is 162 */ 00310000
SnmpPort = 162 00320000
/* */ 00330000
/* this Object Identifier identifies the 'device' */ 00340000
/* which is generating the Trap */ 00350000
/* note: nodes must be less than 128 */ 00360000
Enterprise = '1.3.6.1.4.2' 00370000
/* */ 00380000
/* SNMP defined values for Generic Trap are */ 00390000
/* 0 = cold start */ 00400000
/* 1 = warm start */ 00410000
/* 2 = link down */ 00420000
/* 3 = link up */ 00430000

```

```

/* 4 = authentication failure */ 00440000
/* 5 = egp neighbor loss */ 00450000
/* 6 = enterprise specific */ 00460000
GenericTrap = 6 00470000
00480000
/* specify an integer for Specific Trap */ 00490000
SpecificTrap = 0 00500000
00510000
/* Time Ticks is an integer, it usually represents */ 00520000
/* the device uptime in hundredths of a second */ 00530000
TimeTicks = 5100000 00540000
00550000
/* Object Identifier which uniquely identifies */ 00560000
/* this event (Trap) */ 00570000
/* note: nodes must be less than 128 */ 00580000
ObjectId = '1.3.6.1.4.1.1.2' 00590000
00600000
/* OIDtype represents the type of data, */ 00610000
/* QAOSNMP2 supports Octet String (4) only */ 00620000
OIDtype = 4 00630000
00640000
/* This is the text of the Trap */ 00650000
OIDtext = '123 this is the text of the Alert' 00660000
00670000
00680000
/* Issue NewStack to insulate us from any callers */ 00690000
address mvs 'newstack' 00700000
00710000
/* put on data stack */ 00720000
push Community 00730000
push OIDtext 00740000
00750000
call QAOSNMP2 SnmpManager SnmpPort Enterprise , 00760000
GenericTrap SpecificTrap TimeTicks ObjectId OIDtype 00770000
00780000
/* destroy our stack */ 00790000
address mvs 'delstack' 00800000

```

Chapter 17. Using the Paging Sample

MAINVIEW AutoOPERATOR provides sample code that enables you to communicate with wireless devices, including both text and numeric pagers, which could be one- or two-way devices. You can also obtain confirmations and responses if the destination device and its provider offer these capabilities.

This sample was developed by using the Simple Network Paging Protocol (SNPP rfc 1861) as implemented by SkyTel Communications, Inc.

EXECs

The EXECs for the Paging sample are:

QAOSNPP1 Contains the calling communication logic.

QAOSNPP2 Contains the communication logic.

The EXECs are not compiled, so you can modify them for your application.

QAOSNPP1

The QAOSNPP1 EXEC is the calling sample and does not have input parameters.

QAOSNPP2

The QAOSNPP2 EXEC can have from 0 to 9 parameters, depending on the mode. Each parameter (except the message text) has the following format:

PARAMETER_NAME(VALUE)

Parameters are separated by blanks and can be in any order. The parameters are described in the following table.

Parameter	Description
HELP	Optional. Default is HELP(NO). If HELP(YES) is specified, if the program is called without parameters, or if HELP is specified with no parameters, the calling format is printed to the BBI Journal.
MODE	Optional. Default is MODE(SEND). MODE can have one of the following values: <ul style="list-style-type: none">• SEND - Send the message.• STATUS - Check the message status.

Parameters for MODE(SEND)

If you specify MODE(SEND), additional parameters are available that you can set. These parameters are described in the following table.

Parameter	Description
SERVER	Required. Specify the Internet name or address of the message server provider. For example, for SkyTel, the Internet name is snpp.skytel.com.
PORT	Required. Specify the message server provider port to which the program should connect. In the sample, the port is set to the value 7777.
PIN	Required. Specify the service provider access PIN or a direct telephone number.
USER	Optional. Specify the name of the user to whom you are sending a message. This value is used in the messages and ALERTs issued by the program. Default is NONE.
TIMEOUT	Optional. Specify the time (in seconds) that the program can wait for the TCP/IP response before timing out. Default is 10 seconds.
CONF	Optional. You can specify CONF(YES) if you want to receive a confirmation of a successful message delivery. CONF(YES) can be used only with two-way pagers or telephones with a messaging service. Default is CONF(NO). CONF(NO) can be specified when calling any pager or telephone.

Parameter	Description
DEBUG	Optional. Specify DEBUG(YES) to print the contents of all TCP/IP exchanges (and some other debugging information) to the BBI Journal.
<p>Note: Message text should be written after key parameters.</p> <p>If the message text must contain an open parentheses symbol, (, separate it with <<.</p> <p>When QAOSNPP2 is called from an IMFEXEC SELECT statement, the total statement length is restricted to 256 characters (including information content). If QAOSNPP2 is called from another EXEC with a CALL statement, the length can be up to 2,500 characters.</p>	

Parameters for MODE(STATUS)

Use MODE(STATUS) only when an ALERT is created when a message is sent. MODE(STATUS) has one parameter, which is the variable name containing the list of values for status requests.

Processing Flow

The calling EXEC, QAOSNPP1, has four components:

- A tuning section where you can change parameters for your needs
- A checking section to verify the information
- The CALL statement
- A section for result messages processing

QAOSNPP2 contains the main logic and performs the following tasks:

- It extracts and checks parameters.
- It prints the calling format and exit if help is requested.
- It connects to the server/port and performs the TCP/IP message exchange with a server.
- In the CONF(YES) mode, it creates an AutoOPERATOR ALERT that escalates within a minute and calls a follow-up EXEC in STATUS mode. It creates a unique variable name formatted PAGING.xxxxxxxx (where xxxxxxxx are the last eight numbers of the current time) and puts all necessary information into a long profile variable with this name. The variable name becomes the second parameter of QAOSNPP2.

Note: In the CONF(NO) mode, no more actions required

- When QAOSNPP2 is called in STATUS mode, it sends a status request to the server and replaces the original ALERT with the ALERT escalating up (if the message is still not delivered). Otherwise, it sends a status request to the server and replaces the original ALERT with a clearing ALERT escalating down.

Variables Returned from QAOSNPP2

If the QAOSNPP2 EXEC is invoked from IMFEXEC SELECT, the following variables are returned:

- IMFCC, which contains the return code from the IMFEXEC SELECT statement.
- IMFRC, which contains the condition code returned on the IMFEXEC EXIT statement of QAOSNPP2 when IMFEXEC SELECT WAIT(YES) is specified.
- LOCAL variables APFCC and APFRC, which contain condition and return codes.
- LOCAL variables APFLN1 through APFLNn, which contain additional information.
- LOCAL variable APFNOL, which contains the number of APFLN1-Nn variables.

The following table contains the error messages and the IMFRC, APFRC and APFCC values that are found in LOCAL variables APFLN1 through APFLNn.

Error Message in APFLNn	IMFRC	APFCC	APFRC
PF0000I SUCCESS	0	0	0
PF2001E Message timed out.	8	4	1
PF2002E <Message from the service provider>	8	4	2
PF2002E Transmission ID Error	8	4	2
PF2003E PIN is not recognized by service provider.	8	4	3
PF3001E QAOPAGE1: SOCKET(SOCKET) PF3001E QAOPAGE1: SOCKET(CONNECT) PF3001E QAOPAGE1: SOCKET(WRITE) PF3001E QAOPAGE1: SOCKET(SELECT)	8	8	TCP/IP return code
PF4001E QAOPAGE1: User ID is unknown.	8	12	1
PF4002E QAOPAGE1: Page service provider is unknown.	8	12	2
PF4003E QAOPAGE1: Unable to initialize SOCKET	8	12	3
PF5051E QAOPAGE1: Error found in MODE parameter	8	16	51
PF5052E QAOPAGE1: Error, SERVER or PORT parameter is not providedk.	8	16	52
PF5053E QAOPAGE1: Error in CONF parameter (YES/NO).	8	16	53
PF5054E QAOPAGE1: Message text is not provided.	8	16	54
PF5055E QAOPAGE1: PIN number is not provided or not numeric.	8	16	55
PF5056E QAOPAGE1: Error in TIMEOUT parameter, it is not numeric.	8	16	56

Error Message in APFLNn	IMFRC	APFCC	APFRC
PF5057E QAOPAGE1: Error in DEBUG parameter (YES/NO)	8	16	57
PF5061E QAOPAGE1: Error in parameter structure	8	16	61
PF5062E QAOPAGE1: Unknown parameter	8	16	62
PF5063E QAOPAGE1: Error during VGETL	8	16	63

Chapter 18. Using the AutoOPERATOR Web Sample

This chapter describes how to implement the AutoOPERATOR Web sample.

AutoOPERATOR Web consists of a set of samples distributed with MAINVIEW AutoOPERATOR. The code for these samples was developed for the IBM HTTP Server for OS/390 (WebSphere).

The purpose of these samples is to demonstrate AutoOPERATOR automation possibilities in web browser environments. The samples will not be upgraded or modified in response to user requests. You can choose to personally implement new samples or modify existing samples, but these changes are at your discretion.

Introduction

The AutoOPERATOR Web solution provides sample code where REXX EXECs use AOAnywhere statements in a Common Gateway Interface (CGI) script. When the IBM HTTP Server for OS/390 (WebSphere) is in place, the sample code delivers examples of four AutoOPERATOR features viewed from a web browser:

- Continuous State Manager (CSM)
- AutoOPERATOR ALERTs
- Automation Reporter
- Commands

This chapter documents the four samples.

The samples described in this chapter are not applications. Use these samples to get ideas about how you can create your own automation from a web browser.

The sample code also delivers MAINVIEW for DB2 examples.

What AutoOPERATOR Web Is

AutoOPERATOR Web is based on the AOAnywhere Application Program Interface. The AOAnywhere statements provide the ability to process key AutoOPERATOR functions from outside of the MAINVIEW AutoOPERATOR environment. With AOAnywhere, you can perform AutoOPERATOR automation from a batch program, an IMS MPP, a TSO CLIST, NetView, or UNIX System Services.

The function set of AOAnywhere covers basic AutoOPERATOR functions such as the variable interface, EXEC and ALERT functionality, and other automation functions. When present, SYSPLEX (XCF) connectivity provides multi-system support. AOAnywhere allows AutoOPERATOR programs (EXECs) to be run anywhere.

For more information about the AOAnywhere API and syntax, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

How AutoOPERATOR Web is Used

The AutoOPERATOR Web samples demonstrate how you can access MAINVIEW AutoOPERATOR from a web browser. Accessing AutoOPERATOR from a browser provides you the ability to display automation data and drive automation anywhere where you can open a browser.

For example, you can implement graphical displays that illustrate the hierarchical arrangement of CSM objects or different locales interwoven in an automation web. You can also use bar and pie charts. Graphically, these displays present a picture that is much more accessible to the operator, systems programmer, or support staff.

In addition, many forms of data representations can be much more easily implemented by using HTML, XML, Java, and so on. Even ordinary displays (such as tabular data formats) can become more usable in a browser where table sizing, scrolling, and panning are available.

What the Sample Web Pages Look Like

When installed, the samples create pages that have a similar look and feel, where each sample shows a different aspect of AutoOPERATOR as accessed with AOAnywhere statements in a CGI script. The samples create only a few pages, which are split into a number of individually scrollable panes.

All pages contain a plain white background with an AutoOPERATOR watermark. All of the pages are designed for a browser maximized at 1024x768. For most of the text, a sans serif variable pitch font is used. Detail displays use a serif variable pitch font.

Prerequisites

The section describes what you should know and how to prepare to install the AutoOPERATOR Web samples.

What You Need to Know

You should be familiar with the basic function set of AOAnywhere statements. For more information about the AOAnywhere API and syntax, refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* and the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs*.

You should have a basic understanding of IBM UNIX System Services (USS) because you will have to make changes to UNIX files. For beginners, you can use the IBM TSO ISHELL EXEC to enable you to view directories and edit USS files. This facility requires that your TSO PROC include the SBPX#### libraries; otherwise, the ISHELL EXEC does not function properly. However, someone proficient with UNIX programming can use RLOGIN and use VI to modify these files. Additionally, you might use the OMVS and OEDIT commands to configure your system. You should also be familiar with the IBM HTTP Server for OS/390.

You also need to be aware of the following considerations:

- The samples are written in standard HTML, and the underlying CGI scripts are written in REXX. You should have some understanding of these areas. Only standard HTML is used (without particular extensions), and no style sheets are used. The graphics are in GIF format. In a number of instances, the sample might contain a small amount of Java Script code.
- These HTML files and CGI scripts must be distributed in separate directories, and these directories must have entries added to the WebSphere configuration file to make them accessible to WebSphere.
- You need to review your security. USS uses the UNIX security model. Implementation of any Web application can create security exposures if the application is not properly protected. Ensure that you are familiar with your requirements. If you are implementing security for the first time, you should consult with your UNIX system administrator for security issues. Consistent user ID numbers, group ID numbers, and security policies between platforms will make USS a secure environment.
- You must ensure that OMVS segments exist for user IDs that will access the AutoOPERATOR Web samples.

Warning about Case Sensitivity

Everything in UNIX is case sensitive. Using file names, using directories, or simply modifying a file can have a serious impact if uppercase and lowercase characters are not used properly. If any problems do arise or if files are empty, ensure that you have used the correct case of every letter (for example, `/usr/local/data.txt` is different from `/usr/local/dAta.txt`).

What Information to Gather

The installation process requires that you supply the following information:

Installation directory name: The AutoOPERATOR Web samples must reside in a USS directory. The default name is:

```
/usr/local/bmc/ao62
```

where `/usr/local` is a standard UNIX name for local directories. Many sites do not follow this standard for installing products. You must determine the correct location for your environment. If a directory level is missing, the installation EXEC automatically creates it for you.

Group ID name: UNIX security uses group IDs. These group IDs are RACF-defined groups with an OMVS segment. The group ID that you specify is used for any new directories and all files that are created. Users in this group ID have access to the protected and non-protected applications.

Superuser authority: Superuser, also referred to as system administrator or root user, has access to write and delete all files and directories. To be defined as superuser, you must have BPX.ADMIN authority, or the UID number must be 0. If you are using BPX.ADMIN authority, you might also need to change the owner of the HTTP server configuration file.

You do not need superuser authority to install the AutoOPERATOR Web samples. Use of superuser authority simply makes the job easier. If you are not superuser, verify that the user ID used for installation can create the installation directory and create files and subdirectories in this directory. Write authority is also required for the HTTP server configuration file.

What Software Must Be Installed

IBM's HTTP Server for OS/390 (WebSphere) must be installed and operational. You must also have access to USS.

Note: You should be aware of the following information:

- These samples have been tested on IBM HTTP server and they function in this environment.
- These samples should also function for any HTTP server available on OS/390 that supports REXX as a CGI language.
- These samples should also work with IBM WebSphere Application Server, IBM Domino Go Webserver, or APACHE without any modifications. All other HTTP servers might require some minor modifications.

Installing and Customizing the Samples

This section describes the installation procedure.

Running the Installation EXEC

To run the installation EXEC, you must be logged on as a TSO user who has access to USS.

1. To start the EXEC, issue the following command from the ISPF command line:

```
TSO EXEC 'hi l vl . BBSAMP(QAOWEBI N)'
```

where `hilvl` is the data set prefix for the BBSAMP member.

The first screen prompts you for a function.

2. Enter `INSTALL` to install the product (or enter `REMOVE` to remove the product from USS).
3. Enter the name of the installation directory and group ID.

Use the information that you obtained from “What Information to Gather” on page 113.

4. Press `ENTER`.

Messages about the installation are displayed.

5. Review these messages and ensure that the successful installation message is displayed.

Configuring the HTTP Server

During the installation, the name of a USS file is displayed, `http.conf.changes`. This file resides in the `secure` subdirectory of the installation directory and contains directions for applying necessary changes to the HTTP server configuration file.

The HTTP server uses a pound sign (#) in column 1 to identify comments. The directions in the `http.conf.changes` file are denoted with a pound sign in column 1 so that they are not interpreted as configuration statements.

Enter the following information as instructed in the file:

- **PASS**—Defines a directory (including subdirectories) where HTML and graphic files are located. A virtual directory name is related to this real directory. The virtual directory name must be used in the web browser to access the files in the real directory.
- **EXEC**—Defines a directory where the CGI scripts are located. A virtual directory name is related to the real directory. The virtual directory name must be used in the web browser to access the files in the real directory.
- **PROTECT**—Defines which virtual directory names have a security method applied. The user ID is determined through the **PROTECTION** clause. Virtual directories that are not covered by a **PROTECT** clause will use the **PUBLIC** user ID specified.
- **PROTECTION**—Defines the security method and how the user IDs are determined. Many security methods are available in UNIX but in this case, the RACF sign-on method is implemented. Your RACF user ID and password will be used. When a web browser accesses any directory covered by a **PROTECT** clause that specifies this **PROTECTION** clause, the browser will be prompted for a valid RACF user ID and password before access is allowed.

You must restart the HTTP server after these changes have been made.

Making BBSAMP Available to AutoOPERATOR

In addition to the CGI scripts located in USS, the AutoOPERATOR Web samples have EXECs that execute in the AutoOPERATOR address space. These EXECs must be made available by ensuring that the BBSAMP data set is in the SYSPROC concatenation of the AutoOPERATOR PROC.

Meeting AOAnywhere Requirements

AOAnywhere syntax is used throughout the CGI scripts. For this process to occur, BBLINK must be available to the CGI scripts. The CGI scripts run under USS and therefore, it might be easier to add BBLINK to LNKLST rather than trying to define a STEPLIB to a USS task.

When BBLINK is added to LNKLST, there might be a requirement to authorize BBLINK by using APF, depending on the BMC Software products in use. For details, see “Performing Automation Using AOAnywhere” in the *MAINVIEW AutoOPERATOR Advanced Automation Guide for CLIST EXECs* or the *MAINVIEW AutoOPERATOR Advanced Automation Guide for REXX EXECs* manuals.

Reviewing Security

BMC Software recommends that your UNIX system administrator review the security that has been implemented. The administrator will be familiar with areas that might cause problems or areas that might be more vulnerable.

Subdirectory `secure` (and all of its subdirectories) in your installation directory are covered by one `PROTECT` clause that you added. A valid user ID and password must be entered before access is allowed to these secured applications. Subdirectory `secure` defaults to allowing access to users in the group ID specified during installation. If a user has access to `secure`, that user has access to all files and subdirectories in this directory. If `secure` existed prior to running the installation `EXEC` or was modified after installation, access to the files might be different than documented here.

All other files and subdirectories that are not in the `secure` subdirectory have been made public. These applications are display only.

You can limit public access to display-only applications by increasing the scope of the `PROTECT` clause in the HTTP server configuration. See “Configuring the HTTP Server” on page 115.

You might also consider limiting user access to directories by changing directory access modes. Because the installation process does not change directory access modes if they exist prior to installation, changes to directories are retained when updates are applied. Files on the other hand are replaced. Their file modes are lost. All changes to files will need to be re-applied when installing new versions.

For the best security results, consult with your UNIX system administrator.

Accessing Multiple Systems

The samples were written with the AOAnywhere XCF capability to access AutoOPERATOR subsystems across the SYSPLEX. To implement this feature, at least one AutoOPERATOR 6.1 (or later) must be running on the same system as the HTTP server. If you have systems in different SYSPLEXs, multiple HTTP servers that will run these samples, or a configuration where the samples must run under different HTTP servers, you need to install the samples on each affected system. Most installations of USS do not share Hierarchical File System (HFS) directories. If this is your case, you need to install on each system where the AutoOPERATOR Web samples run under a HTTP server.

Accessing the Samples

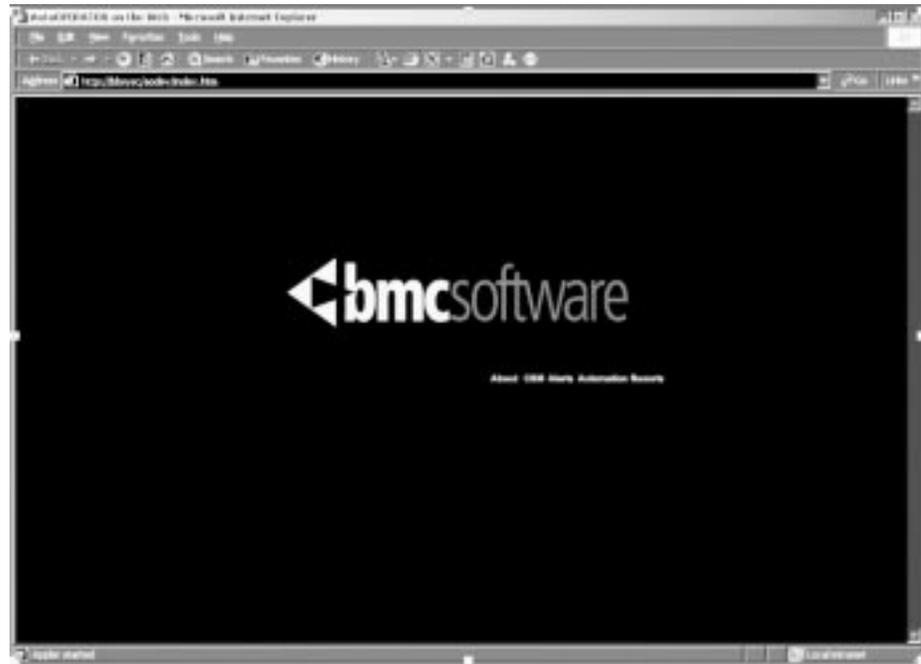
To access the samples, start a web browser on your PC or UNIX system. Change the address in the web browser to the DNS-defined name of your OS/390 TCP/IP address that is running the HTTP server, followed by `/ao`. If you do not know the DNS-defined name, substitute the TCP/IP address for that system (for example, `bmcsysa/ao` or `192. 0. 0. 1/ao`). If you did not change the HTTP server `WELCOME` statement, `index.html` will be automatically displayed. If `index.html` was removed, add `/i ndex. html` to the address in the browser.

Viewing the Entry Page

The entry page consists of a BMC Software logo and hyperlinks (see Figure 2).

Below the logo is a small navigation bar with hyperlinks. The hyperlink text changes colors when the cursor passes over it. These same hyperlinks are present on all pages. By clicking one of these hyperlinks, you can access the other pages.

Figure 2. AutoOPERATOR Web: Entry Page



Viewing the About Page

The About page shows the WebAO heading at the top of the page, which is common to all pages except the entry page. Using the navigation bar on the right side, you can link to all other pages on the site. The WebAO heading in the upper left is also a active hyperlink that changes color as the cursor passes over it. You can click on this heading to access the entry page.

The About page (see Figure 3) introduces the fundamentals of the site and AOAnywhere.

Figure 3. AutoOPERATOR Web: About Page

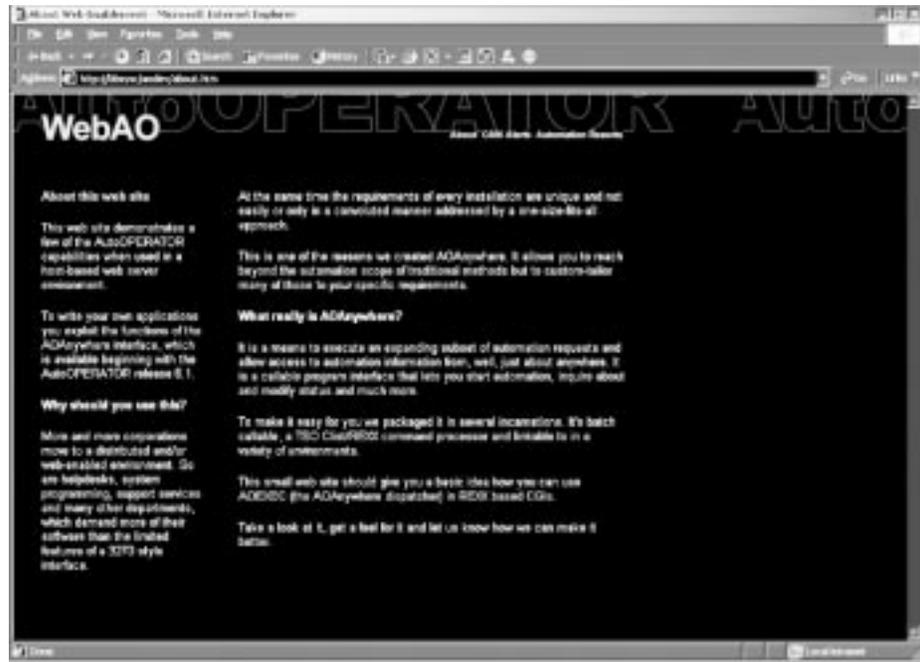


Figure 5. AutoOPERATOR Web: CSM Page with Objects



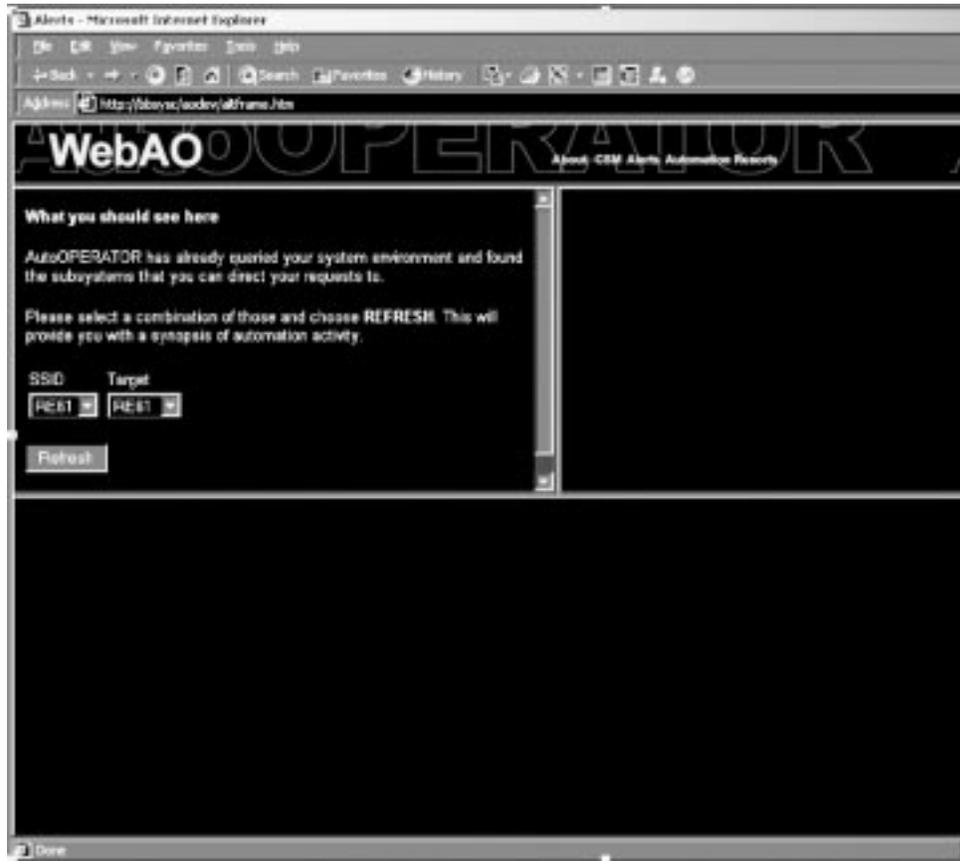
In Figure 5, the bottom pane shows CSM objects arranged hierarchically, where the indented position denotes the parent-child relationships. The pane is scrollable. The right portion of the pane shows the descriptions that are associated with each object. The descriptions are color-coded according to their hierarchy level.

The objects themselves can be hyperlinks and are DHTML animated (where they change color as the cursor passes over them). As hyperlinks, they are not color-coded according to their hierarchy level, but they change colors according to whether they have been viewed. If you click on one of the objects, detailed data is displayed in the middle right pane (see Figure 6 on page 121).

Viewing the ALERTs Page

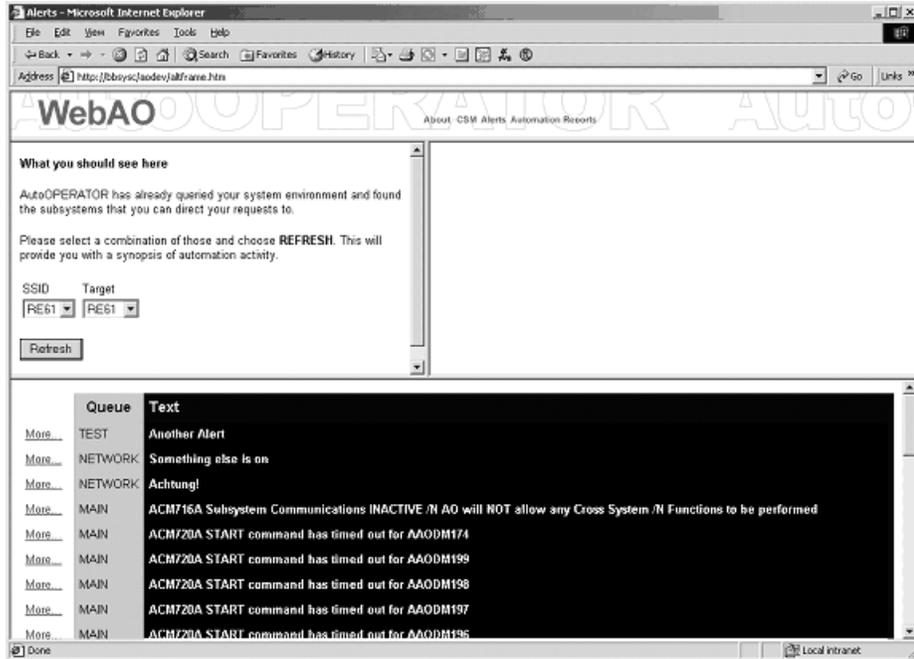
The initial ALERTs page is identical to the CSM page and Automation Reporter page, except for the name (“ALERTs”) in the title bar of the browser window (see Figure 7).

Figure 7. AutoOPERATOR Web: ALERTs Page



A Refresh button activates the actual retrieval (and refresh) of data. The data appears in the bottom pane (see Figure 8 on page 123).

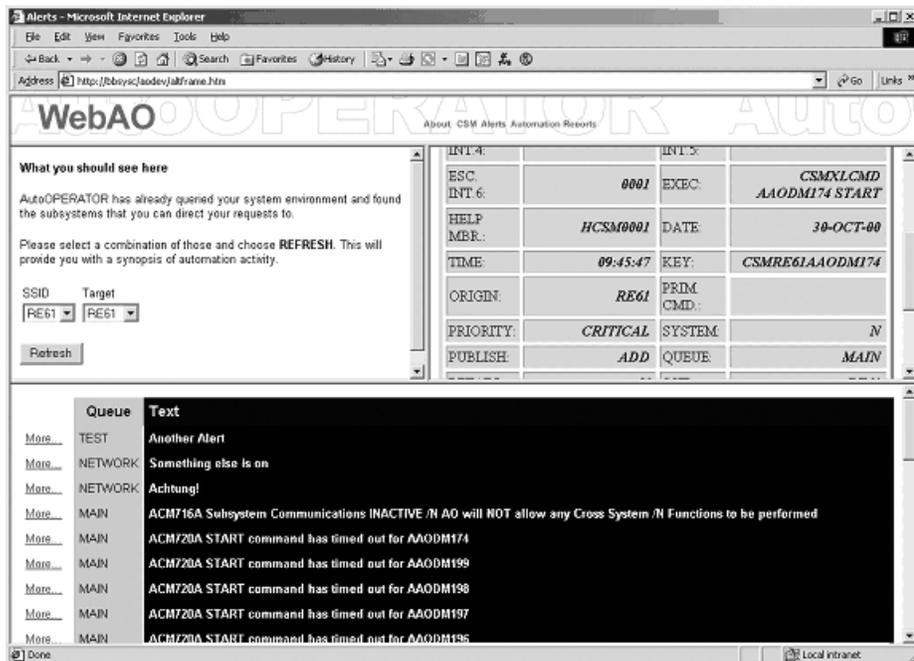
Figure 8. AutoOPERATOR Web: ALERTs Page with Data



The first column of the bottom pane consists of [More...](#) hyperlinks that link to further information. The Queue column shows the name of the ALERT queue.

The third column uses a black background for better visibility of the ALERT colors, which denote priority and severity. Clicking the [More...](#) hyperlink displays a scrollable middle right pane where all data that is associated with this particular ALERT is displayed.

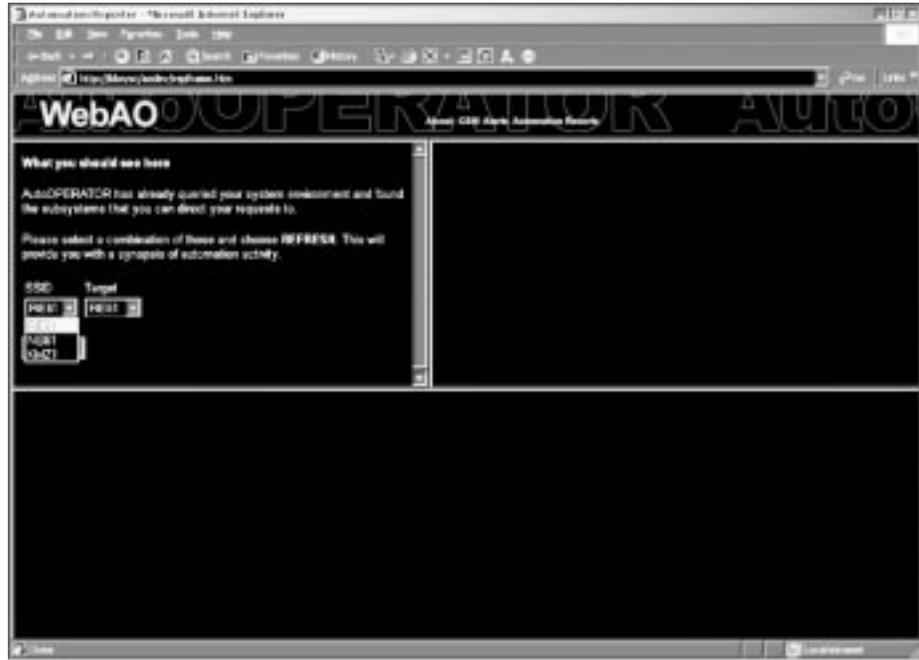
Figure 9. AutoOPERATOR Web: Scrolling ALERTs



Viewing the Automation Reporter Page

This sample requires Automation Reporter to be active on the targeted system. The initial Automation Reporter page is identical to the CSM page and ALERTs page, except for the name (“Automation”) in the title bar of the browser window (see Figure 10).

Figure 10. AutoOPERATOR Web: Automation Reporter Page



When you specify subsystems and targets from the drop-down boxes, automation statistics are displayed, sorted from most recent statistics oldest statistics, as shown in Figure 11 on page 125. This data is directly read from the Automation Reporter data gatherer.

Figure 11. AutoOPERATOR Web: Automation Reporter Data Displayed



If you click on an interval in the first column, the full set of statistics is displayed in a scrollable middle left pane, as shown in Figure 12:

Figure 12. AutoOPERATOR Web: Automation Reporter Statistics Displayed



Viewing the Commands Page

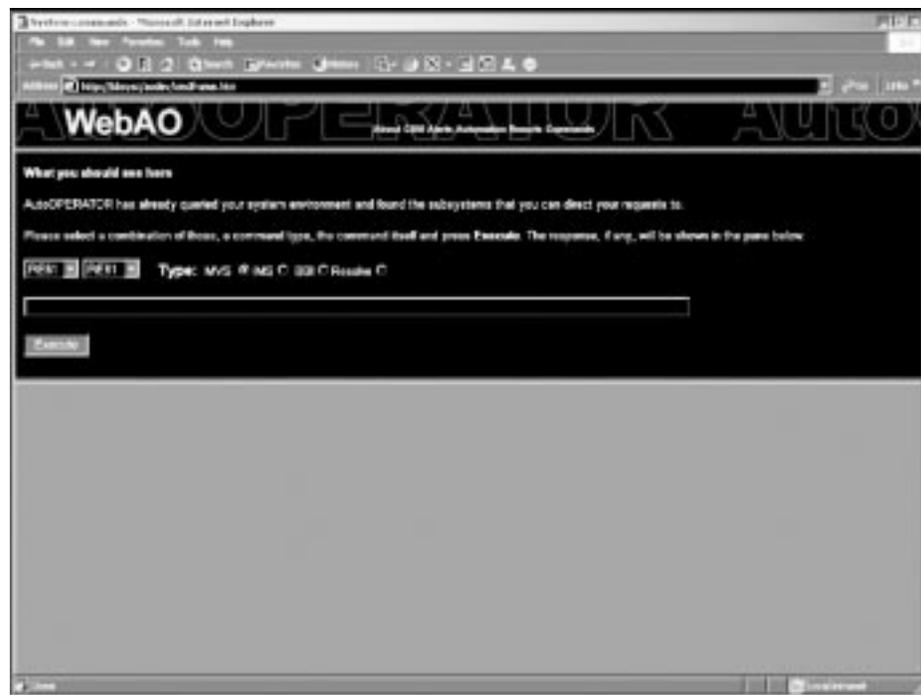
The initial Commands page is divided into three panes:

- WebAO heading and hyperlinks in the top pane
- various input fields in the middle pane
- an output pane at the bottom

The output pane is intentionally silver. In the middle pane, you can use the drop-down boxes to select local and target subsystems, and you can specify the type of command to be issued (prefix characters such as # and. are acceptable). Currently, MVS, IMS, BBI, and Resolve commands may be selected.

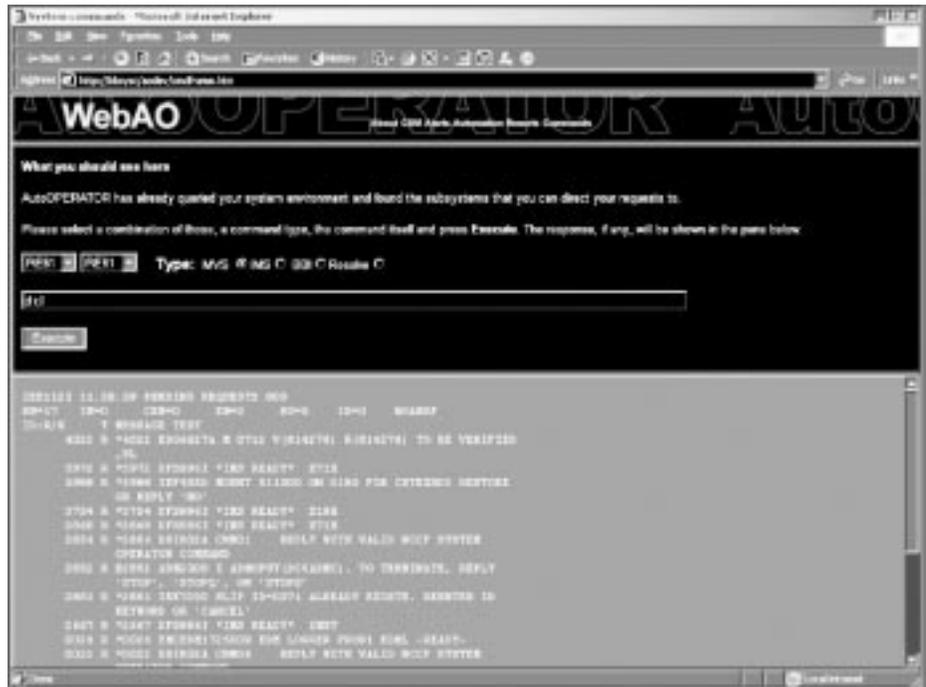
The text box (below the drop-down boxes) is designated as the input area for the command itself, and the Execute button submits the command.

Figure 13. AutoOPERATOR Web: Commands Page



When you make selections, output is displayed at the bottom (see Figure 14 on page 127):

Figure 14. AutoOPERATOR Web: Command Output Displayed



The output is scrollable.

Debugging the Distributed CGI Scripts

All of the CGI scripts were created to be as independent of their environment as possible. With the exception of parameter parsing, these EXECs will run successfully in any environment.

The following sections contain tips for understanding and debugging the sample CGI scripts.

Understanding CGIs

The AutoOPERATOR Web samples consist of HTML tagging and CGI scripts. The CGI scripts are created as REXX EXECs. These EXECs obtain their input from the stack and then parse the information according to CGI requirements. All other statements in the AutoOPERATOR Web samples are compatible with any available REXX environment.

A CGI runs in the same way as a normal REXX EXEC except that all SYSTSPRT output is directed to the web browser (either Netscape or Microsoft Internet Explorer). The 'SAY "Content-type: text/html "' statement is required because it tells the browser what to do with the SYSTSPRT data. In the sample code, it is coded to tell the browser to expect HTML from SYSTSPRT.

Debugging the CGI Script in a TSO Environment

All sample CGIs can be copied to a CLIST library and run as TSO CLISTs. The only modification required is parsing the parameters and getting the test parameters into the EXEC. To do this, disable the parse routine and manually set the required variables. The output that would be directed to the browser now appears on your TSO session. Be aware that HTML formatted data will appear on your screen, enclosed in < > characters.

Debugging the CGI Script without HTML Decoding

You can use the REXX TRACE command when running a CGI script. If TRACE is specified, only SYSTSPRT data occurring after the SAY "Content-type: text/html" statement appears in the browser window. One problem with this process is that trace data will be displayed as a continuous stream of data. To avoid this problem, turn off HTML decoding by changing the content type to "Content-type: text/plain".

Avoiding Frames in the Display

Parts of the sample code contain HTML frames. Frames split the browser screen into windows. If the screen is split into frames, you can determine the frame definitions by changing the address specified in the address field of the browser. Each frame has HTML or CGI coding that has been executed. Determine which HTML or CGI code was executed for that frame and use that code in the browser's address field. This process separates the executable data into separate browser screens to make problem diagnosis a little simpler.

Displaying the HTML Source in a Frame

You can choose to display only the HTML source for a frame definition. To display the source for a frame, right-click anywhere in the frame and choose Source or choose Source from the View menu.

Refreshing HTML Source Code

HTML source code is stored in the browser's cache. To refresh changes that are made to the HTML source code:

- For Internet Explorer, hold down the Ctrl key and click on the browser's Refresh button.
- For Netscape, hold down the Shift key and click refresh.

CGI output is forced to refresh by using the PRAGMA directive with NO-CACHE into its output.

Common Error Messages

This section contains common error messages that you might encounter and describes what they mean.

Table 4. Common Error Messages

Message	Description
Can't locate HTTP configuration file.	The configuration file is listed in the JCL startup proc for your HTTP server, after the -r option. If you are using the free IBM HTTP server and if you have only one copy, the name of the configuration file will be /etc/http.conf. If a period precedes the file name (/etc/http.conf), you need to check the LEPARM variable in the proc for the specification of _CEE_ENVFILE. The file specified here contains a home directory name. The file will be located relative to this home directory name if the configuration file name was specified beginning with a period.
"* - * address linkmvs "AOEXEC P1" +++ RC(-3) +++"	If this message is displayed in a frame, it means that AOAnywhere is not available in linklist or in the USS steplib variable.
Web browser 403 - 403	A security problem has occurred. You should review each directory level of your installation directory. Ensure that READ and EXECUTE are on of the GROUP and OTHER modes of each subdirectory.

Table 4. Common Error Messages (Continued)

Message	Description
Web browser 404 - 404	This message indicates an “address cannot be found” condition. If this message occurs with the first screen, you need to verify that the PASS statement that you added is not preceded by the PASS /* in your HTTP configuration file. The address search starts from the top of the configuration file and stops when the first PASS or EXEC address is true for this address (for example, if /* precedes /ao/*, execution would stop at /* because it matches all address).
Frame is missing the available system list and the REFRESH button	The server side includes are not working in the HTTP server. Ensure that your HTTP configuration file specifies at least "imbeds on SSIOOnly" and the ADDTYPE for ".shmtl", and ensure that text/x-ssi-html is specified.

Part 6. Appendixes

This section contains five appendixes.

Appendix A. Sample EXECs	133
@TIMER	133
JESDOWN	144
Appendix B. Sample REXX EXECs	151
REXX Compound Variable Initialization EXECs	152
REXX DASD Reserve Analysis EXEC	154
REXX QUEUE/PULL and VPUT/VGET Usage	157
Appendix C. MVS Solution Variables	159
Appendix D. CICS Solution Variables	161
Appendix E. DB2 Solution Variables and Parameters	163
Data Sets Containing Variables	163
Data Sets Containing Threshold Parameters	163
Tables That Describe Variables and Parameters	164

Appendix A. Sample EXECs

This appendix shows and explains two EXECs: @TIMER and JESDOWN.

@TIMER

The following discussion refers to [Figure 15 on page 136](#).

The @TIMER EXEC is an interface utility to the timer queue. Timer queue elements are processed by the Interval Services Application.

When the @TIMER EXEC is entered, parameters are parsed and passed to it by the EXEC Management application. The caller-supplied parameters are then used during the processing of the request.

All input parameters use keywords (**lines 1-2**).

Lines 2-88 contain the internal documentation for the EXEC. “[EXEC Distribution](#)” on page 3 contains a detailed discussion of the documentation box.

After the documentation box, **lines 90-91** declare the TMRELEM variable (VDCL). The variable, NULL, is declared for use in later comparisons (**line 93**). If the process name was not passed, an error message is issued (**lines 95-102**). Defaults for the parameters RETAIN, INTERVAL, and GOODFOR are defined (**lines 106-144**); error messages are issued as appropriate.

The STOPTIME parameter is processed in **lines 146-160**. If the parameter is not specified, the default value is set. If the parameter is specified, an edit is done to ensure that the length is eight bytes; if not, an error message is issued and the processing is terminated.

Lines 162-172 edit the TOD parameter. If it is specified, the length must be eight bytes. If it is not specified, an error message is issued and processing is terminated.

Line 178 tests the function for an ADD request. If the test is positive, the logic in **lines 180-336** is performed.

The logic in **lines 180-336** processes the ADD request. First, a check is made to ensure that this is not a duplicate request (**lines 180-187**). The check for duplication is made by getting the variable &SPROC into the TMRELEM list variable. List variables are defined with positional, subordinate variables. If the return code from the variable get (VGET) is 0, then a duplicate was found and a message is issued and processing ends.

Lines 189-208 validate the positional parameters TOD and NEXTTIME; one of the parameters must be specified, but not both. Appropriate error messages are issued before the EXIT is taken.

Line 212 issues an ENQ on the resource TMRCTR; the resource is unique and the ENQ ensures data integrity by serializing its modification. When the ENQ is completed, a VGET is issued for TMRCTR and TMRNEXT from the PROFILE POOL. If the return code is not 0, the variables have not been initialized and the variables are set to 0 (**lines 214-217**).

Lines 218-220 increment the TMRCTR and TMRNEXT values; the current request ID &SID is set to TMRNEXT.

Lines 224-234 set the TOD (time of day) value from the NEXTTIME parameter. The NEXTTIME parameter can be specified by the caller to indicate that the process is to be scheduled in x number of minutes, rather than specifying an exact time of day (TOD). Because the NEXTTIME parameter can be four bytes long, calculations are performed to round up the hours (**lines 225-228**). **Line 229** ensures that the resulting minutes, MM, is two characters long by placing a leading 0. **Lines 230-233** calculate the hours, HH, and ensure that the resulting value is two bytes long.

Line 238 sets the variable &STIME, which is used to reschedule the timer element until the GOODFOR value has elapsed. The timer queues are kept intact across initializations. Therefore, it is possible to set up a timer element that is good for MMMM minutes (the maximum value of 9999 minutes allows for six days of prescheduling).

Lines 242-245 process the STOPTIME parameter. If specified, the variable &XSTOP is set.

Lines 249-271, manipulate and process the GOODFOR value to calculate the ending year (yy), day (ddd), hour (hh), and minute (mm); the process that is scheduled is retained across all restarts until GOODFOR (>IME) date and time have passed.

Lines 275-278 set values for the list variables that were declared in line 91. The variables are put (VPUT) into the PROFILE POOL.

Lines 280-290 process the TARGETSS parameter. If not specified, the list variable &XSSID is set to &IMFORGSS; otherwise &XSSID is set to &TARGETSS.

Lines 294-300 call the interface to the Interval Services Application to process the timer request for a one-time-only event. Because the caller did not specify an interval, the @TIMER EXEC assumes that the process is to be stopped after it is scheduled for execution the first time (STOPCNT=1).

Lines 301-307 call the interface to the Interval Services Application to process the timer request that is rescheduled at specified intervals indefinitely; in other words, a STOPTIME was not specified. If the STOPTIME was specified by the caller, the logic in **lines 308-314** is executed.

Lines 316-331 check return codes, issue appropriate messages, and put the variables (VPUT) into the PROFILE pool.

Line 334 dequeues (VDEQ) the TMRCTR resource.

Lines 338-415 process the request to DELETE an element from the timer queue. A check is made to ensure that the element to be deleted exists (lines 346-353).

Lines 355-365 process the TARGETSS parameter; if not specified, &TARGETSS is set to &IMFORGSS.

Lines 367-379 call the interface to the Interval Services Application to delete the element from the timer queue if the caller is not the Interval Services Program, ISRBPT.

Lines 381-384 check return codes and issue appropriate messages.

Lines 386-415 delete all references to the element from the PROFILE POOL and reset the timer queue counter, TMRCTR.

Lines 417-424 are executed only if an invalid function was passed by the caller. An error message is issued indicating that an invalid function was specified and the processing is terminated.

```

PROC 1 TIMER FUNC() PROC() NEXTTIME() TOD() INTERVAL() STOPTIME()+1
GOODFOR() RETAIN() TARGETSS() ARBSRV() 2
/*****/3
/* DOC GROUP(MVS) FUNC(UTILITIES) CODE(UT) */
/* DOC DISP(YES) AUTHOR(B&B) */
/* DOC DESC(SCHEDULE A TIMER EVENT) */
/*-----*/
/* NAME: */
/* @TIMER */
/* */
/* DESCRIPTION: */
/* SET A TIMED EVENT */
/* */
/* SEE ALSO: */
/* MJS013C */
/* MJS014C */
/* */
/* INVOKED BY: */
/* CALLING EXEC */
/* */
/* INPUT PARAMETERS: */
/* NAME: DESCRIPTION: DEFAULT VALUE: */
/* FUNC FUNCTION TO PERFORM NONE */
/* PERFORM (ADD|DEL) */
/* PROC NAME OF PROCEDURE NONE */
/* TO SCHEDULE */
/* NEXTTIME TIME IN MINUTES TO NONE */
/* SCHEDULE */
/* (MMM) */
/* TOD TIME TO SCHEDULE NONE */
/* (HH: MM: SS) */
/* INTERVAL REPETITION INTERVAL NONE */
/* (HH: MM: SS) */
/* STOPTIME TIME TO STOP NONE */
/* SCHEDULING (HH: MM: SS) */
/* GOODFOR NUMBER OF MINUTES NONE */
/* THIS EVENT IS GOOD */
/* FOR (MMM) */
/* RETAIN RETAIN FOR CATCHUP Y */
/* (Y|N) */
/* TARGETSS SSID OF THE TARGET NONE */
/* FOR EXECUTION */
/* ARBSRV INTERNAL INVOCATION NONE */
/* INDICATOR */
/* */
/* OUTPUT: */
/* NONE */
/* */
/* OUTBOARD CALLS: */
/* NONE */
/* */
/* EXTERNAL ROUTINES CALLED: */
/* NONE */
/* */
/* ALERTS ISSUED: */
/* NONE */
/* */

```

Figure 15. @TIMER EXEC

```

/* VARIABLES USED: */
/* NAME: DESCRIPTION: DEFAULT VALUE: */
/* TMRCTR NUMBER OF ACTIVE 0 */60
/* TIMERS */
/* TMRNEXT LAST ASSIGNED TIMER 0 */
/* ELEMENT NUMBER */
/* SID ELEMENT NUMBER NONE */
/* FOR THE CURRENT */
/* PROCESS */
/* STIME START TIME NONE */
/* GTIME GOOD FOR TIME NONE */
/* XPROC PROCESS NAME NONE */
/* XRETAIN RETAIN PROCESS? NONE */70
/* XINTEVAL SCHEDULING TIME NONE */
/* INTERVALS */
/* XSTOP TIME TO STOP NONE */
/* SCHEDULING */
/* XSSID SSID ON WHICH TO NONE */
/* SCHEDULE */
/*
/* TEST SCRIPT INFO: */
/*
/* RETURN CODES: */80
/* 00 - ELEMENT WAS ADDED/DELETED (FOR ADD, IMFRC=ID) */
/* 04 - DUPLICATE FOUND - ADD */
/* 08 - ELEMENT NOT FOUND - DEL */
/* 16 - PARAMETER ERROR */
/*
/* CHANGE LOG: */
/* 04-02-90 BY MDR: ADDED DOCUMENTATION BOX */
/******

IMFEXEC VDCL + 90
  TMRELEM LIST(SID STIME GTIME XPROC XRETAIN XINTEVAL XSTOP XSSID)

SET &NULL = &STR( ) /* SET NULL VALUE FOR LATER */

/*-----*/
/* IF PROC NOT SPECIFIED, SERIOUS ERROR; TERMINATE PROCESSING */
/*-----*/
IF &STR(&PROC) = &NULL THEN DO /* PROC MUST BE SPECIFIED*/
  IMFEXEC MSG '.@TIMER PROC MISSING, REQUIRED'
  IMFEXEC EXIT CODE(16) 100
  EXIT
END

```

Figure 15. @TIMER EXEC (Continued)

```

SET &SPROC = &PROC
/*-----*/
/* IF RETAIN NOT SPECIFIED, DEFAULT TO RETAIN=Y */
/*-----*/
IF &STR(&RETAIN) = &NULL THEN DO
    SET &RETAIN = &STR(Y) 110
END
ELSE DO
    IF &STR(&RETAIN) NE Y THEN DO
        IF &STR(&RETAIN) NE N THEN DO
            IMFEXEC MSG '.@TIMER VALUE FOR RETAIN MUST BE Y OR N'
            IMFEXEC MSG '.@TIMER PASSED VALUE FOR RETAIN=&RETAIN'
            IMFEXEC EXIT CODE(16)
            EXIT
        END
    END
END 120

/*-----*/
/* IF INTERVAL NOT SPECIFIED, SET TO 00:00:00; */
/* IF INTERVAL SPECIFIED, MUST BE IN THE HH:MM:SS FORMAT */
/*-----*/
IF &STR(&INTERVAL) = &NULL THEN DO
    SET &INTERVAL = &STR(00:00:00)
END
ELSE DO 130
    IF &LENGTH(&INTERVAL) NE 8 THEN DO
        IMFEXEC MSG '.@TIMER INTERVAL MUST BE IN HH:MM:SS FORMAT'
        IMFEXEC MSG '.@TIMER PASSED VALUE FOR INTERVAL=&INTERVAL'
        IMFEXEC EXIT CODE(16)
        EXIT
    END
END

/*-----*/
/* IF GOODFOR NOT SPECIFIED, SET TO 0000 */ 140
/*-----*/
IF &STR(&GOODFOR) = &NULL THEN DO
    SET &GOODFOR = &STR(0000)
END

/*-----*/
/* IF STOPTIME NOT SPECIFIED, SET TO 00:00:00 */
/* IF STOPTIME SPECIFIED, MUST BE IN HH:MM:SS FORMAT */
/*-----*/
IF &STR(&STOPTIME) = &NULL THEN DO 150
    SET &STOPTIME = &STR(00:00:00)
END
ELSE DO
    IF &LENGTH(&STOPTIME) NE 8 THEN DO
        IMFEXEC MSG '.@TIMER STOPTIME MUST BE IN HH:MM:SS FORMAT'
        IMFEXEC MSG '.@TIMER PASSED VALUE FOR STOPTIME=&STOPTIME'
        IMFEXEC EXIT CODE(16)
        EXIT
    END
END 160

```

Figure 15. @TIMER EXEC (Continued)

```

/*-----*/
/*   IF TOD SPECIFIED, MUST BE IN HH:MM:SS FORMAT   */
/*-----*/
IF &STR(&TOD) NE &NULL THEN DO
  IF &LENGTH(&TOD) NE 8 THEN DO
    IMFEXEC MSG '.@TIMER TOD MUST BE IN HH:MM:SS FORMAT'
    IMFEXEC MSG '.@TIMER PASSED VALUE FOR TOD=&TOD'
    IMFEXEC EXIT CODE(16)
    EXIT
  END
END
170

/*-----*/
/*   TEST FOR FUNCTION = ADD   */
/*-----*/

IF &STR(&FUNC) = ADD THEN DO

/*== LOOK FOR DUPLICATE TIMER ELEMENT */
180

  IMFEXEC VGET &SPROC INTO(TMRELEM) PROFILE
  IF &IMFCC EQ 0 THEN DO
    IMFEXEC MSG '.@TIMER DUPLICATE ELEMENT FOUND - &SPROC'
    IMFEXEC EXIT CODE(4)
    EXIT
  END
/*== REQUIRES TOD() OR NEXTTIME() TO BE SPECIFIED */
190

  IF &STR(&TOD) = &NULL THEN DO
    IF &STR(&NEXTTIME) = &NULL THEN DO
      IMFEXEC MSG '.@TIMER TOD() OR NEXTTIME() REQUIRED'
      IMFEXEC EXIT CODE(16)
      EXIT
    END
  END
/*== TOD() AND NEXTTIME() CAN NOT BE SPECIFIED */
200

  IF &STR(&TOD) NE &NULL THEN DO
    IF &STR(&NEXTTIME) NE &NULL THEN DO
      IMFEXEC MSG +
        '.@TIMER TOD() AND NEXTTIME() ARE MUTUALLY EXCLUSIVE'
      IMFEXEC EXIT CODE(16)
      EXIT
    END
  END
/*== LOCK THE TIME COUNTER AND QUEUES */
210

  IMFEXEC VENQ 'TMRCTR' EXC
  IMFEXEC VGET (TMRCTR TMRNEXT) PROFILE
  IF &IMFCC NE 0 THEN DO
    SET &TMRCTR = 0
    SET &TMRNEXT = 0
  END
  SET &TMRCTR = &TMRCTR + 1
  SET &TMRNEXT = &TMRNEXT + 1
  SET &SID = &TMRNEXT
220

```

Figure 15. @TIMER EXEC (Continued)

```

/*== CREATE TOD FROM NEXTTIME IN HH:MM:SS FORMAT ==*/

IF &STR(&NEXTTIME) NE &NULL THEN DO
  SET MM = &SUBSTR(4:5, &SYSTIME)
  SET MM = &MM + &NEXTTIME
  SET HR = &MM / 60
  SET MM = &MM - (&HR * 60)
  IF &MM LT 10 THEN SET MM = &STR(0&MM)
  SET HH = &SUBSTR(1:2, &SYSTIME)
  SET HH = &HH + &HR
  IF &HH LT 10 THEN SET HH = &STR(0&HH)
  SET &TOD = &STR(&HH: &MM: 00)
END

/*== CREATE STIME IN YY.DDD/HH:MM */

SET &STIME = &STR(&SYSJDATE/&TOD)

/*== SET XSTOP TIME */

SET &XSTOP = &STR(00.000/00:00:00)
IF &STR(&STOPTIME) NE &STR(00:00:00) THEN DO
  SET &XSTOP = &STR(&SYSJDATE/&STOPTIME)
END

/*== CREATE GTIME IN YY.DDD/HH:MM */

SET &GTIME = &STR(00.000/00:00)
IF &STR(&GOODFOR) NE &STR(0000) THEN DO
  SET MM = &SUBSTR(4:5, &TOD)
  SET MM = &MM + &GOODFOR
  SET HR = &MM / 60
  SET MM = &MM - (&HR * 60)
  IF &MM LT 10 THEN SET MM = &STR(0&MM)
  SET HH = &SUBSTR(1:2, &TOD)
  SET HH = &HH + &HR
  SET DAY = &HH / 24
  SET HH = &HH - (&DAY * 24)
  IF &HH LT 10 THEN SET HH = &STR(0&HH)
  SET DDD = &SUBSTR(4:6, &SYSJDATE)
  SET DDD = &DDD + &DAY
  SET YR = &DDD / 365
  SET DDD = &DDD - (&YR * 365)
  IF &DDD LT 100 THEN DO
    SET DDD = &STR(0&DDD)
  END
  SET YY = &SUBSTR(1:2, &SYSJDATE)
  SET YY = &YY + &YR
  SET &GTIME = &STR(&YY. . &DDD/&HH: &MM)
END

/*== CREATE PROFILE VARIABLE */

SET &XPROC = &SPROC
SET &XRETAIN = &RETAIN
SET &XINTERVAL = &INTERVAL
SET &TMR&SID = &SPROC

```

Figure 15. @TIMER EXEC (Continued)

```

-----*/280
/*  IF TARGETSS SPECIFIED, SET XSSID TO TARGETSS */
/*  OTHERWISE, USE IMFORGSS */
/*-----*/
  IF &STR(&TARGETSS) NE &NULL THEN DO
    SET &XSSID = &STR(&TARGETSS)
  END
  ELSE DO
    SET &XSSID = &IMFORGSS
    SET &TARGETSS = &IMFORGSS
  END
  290

/*  SET THE TIMER */

SET &IMFRC = 0
IF &INTERVAL = &STR(00:00:00) THEN DO /* ONE TIME ONLY... */
  IMFEXEC IMFC SET +
    REQ=CALLX &SPROC, +
    START=&TOD, STOPCNT=1, +
    TARGET=&TARGETSS
END
  300
ELSE DO
  IF &STOPTIME = &STR(00:00:00) THEN DO /*STOP TIME SPECIFIED?*/
    IMFEXEC IMFC SET +
      REQ=CALLX &SPROC, +
      START=&TOD, I=&INTERVAL, +
      TARGET=&TARGETSS
  END
  ELSE DO
    IMFEXEC IMFC SET +
      REQ=CALLX &SPROC, +
      START=&TOD, I=&INTERVAL, +
      STOP=&STOPTIME, +
      TARGET=&TARGETSS
  END
  310
END
END
IF &IMFCC NE 0 THEN DO
  IMFEXEC MSG '.@TIMER ERROR IN SETC CALLX TMR&SID'
END
ELSE DO
  IF &IMFCC GT 4 THEN DO
    IMFEXEC MSG +
      '.@TIMER ERROR ADDING ELEMENT TMR&SID TO PROFILE POOL'
  END
  ELSE DO
    IMFEXEC VPUT TMR&SID PROFILE
    IMFEXEC VPUT &SPROC FROM(TMRELEM) PROFILE
    IMFEXEC VPUT (TMRCTR TMRNEXT) PROFILE
    IMFEXEC MSG +
      '.@TIMER ADDED ELEMENT - &SPROC - CONTROL(TMR&SID)'
  END
  320
END
END
/*== UNLOCK TIMER COUNTER AND QUEUES */
IMFEXEC VDEQ 'TMRCTR'
EXIT
END
  330

```

Figure 15. @TIMER EXEC (Continued)

```

/*-----*/
/* TEST FOR FUNCTION = DEL */
/*-----*/340

IF &STR(&FUNC) = &STR(DEL) OR IF &STR(&FUNC) = &STR(DELETE) THEN DO

SET &SPROC = &PROC

/*== GET THE VARIABLE FROM THE PROFILE */

IMFEXEC VGET &SPROC INTO(TMRELEM) PROFILE
IF &IMFCC NE 0 THEN DO
IMFEXEC MSG '.@TIMER ELEMENT NOT FOUND - &SPROC'
350
IMFEXEC EXIT CODE(8)
EXIT
END

/*-----*/
/* IF TARGETSS SPECIFIED, SET XSSID TO TARGETSS */
/* OTHERWISE, USE IMFORGSS */
/*-----*/
IF &STR(&TARGETSS) NE &NULL THEN DO
SET &XSSID = &STR(&TARGETSS) 360
END
ELSE DO
SET &XSSID = &IMFORGSS
SET &TARGETSS = &IMFORGSS
END

/*-----*/
/*KILL THE TIMER: SINCE THIS EXEC IS ALSO CALLED BY ISRBPT */
/* PROGRAM TO DELETE THE INFORMATION FROM THE */
/* PROFILE POOL WHEN A TIMER ELEMENT EXPIRES, */370
/* THE ARBSRV PARM IS CHECKED FOR CALLER ID. */
/* IF ARBSRV IS NON-BLANK, THE CALL TO PRG=CALLX*/
/* IS BYPASSED..... */
/*-----*/
IF &ARBSRV = &NULL THEN DO
IMFEXEC IMFC SET +
PRG=CALLX &SPROC +
TARGET=&TARGETSS
END
380

IF &IMFCC NE 0 THEN DO
IMFEXEC MSG '.@TIMER ERROR IN SETC CALLX TMR&SID'
END
IMFEXEC MSG '.@TIMER DELETED ELEMENT - &SPROC'

```

Figure 15. @TIMER EXEC (Continued)

```

/*== DELETE THE CONTROL VARIABLE                               */
IMFEXEC VDEL TMR&SID INTO(TMRELEM) PROFILE

IMFEXEC VENQ 'TMRCTR' EXC                                     390
IMFEXEC VGET (TMRCTR TMRNEXT) PROFILE
SET &TMRCTR = &TMRCTR - 1
IF &TMRCTR = 0 THEN DO
    SET &TMRNEXT = 0
END
IMFEXEC VPUT (TMRCTR TMRNEXT) PROFILE
IMFEXEC VDEQ 'TMRCTR'

IF &IMFCC NE 0 THEN DO
    IMFEXEC MSG '.@TIMER CONTROL VARIABLE NOT FOUND - TMR&SID' 400
    IMFEXEC EXIT CODE(8)
    EXIT
END

/*== DELETE THE TIMER ELEMENT                                 */

IMFEXEC VDEL &SPROC PROFILE
IF &IMFCC NE 0 THEN DO
    IMFEXEC MSG '.@TIMER CONTROL VARIABLE NOT FOUND - TMR&SID' 410
    IMFEXEC EXIT CODE(8)
    EXIT
END

EXIT
END

/*-----*/
/* INVALID FUNCTION                                           */
/*-----*/
DO                                                             420
    IMFEXEC MSG '.@TIMER INVALID FUNCTION SPECIFIED'
    IMFEXEC EXIT CODE(16)
    EXIT
END                                                             424

```

Figure 15. @TIMER EXEC (Continued)

JESDOWN

The following discussion refers to [Figure 16 on page 146](#).

The JESDOWN EXEC is invoked by the operator to shutdown JES2.

Lines 2-74 contain the internal documentation for the EXEC. “[EXEC Distribution](#)” on page 3 contains a detailed discussion of the documentation box.

After the documentation box, the RMTS636, LNES636, ACTS636, PRTS636, and PJES2 variables are declared (**lines 78-86**); these are list variables because they contain subordinates.

Line 87 initializes the &RLPJ variable.

Line 88 gets the value for the JESWAIT PROFILE POOL variable, which was VPUT during MAINVIEW AutoOPERATOR initialization.

Lines 92-105 give the operator an option to cancel the JESDOWN EXEC. A WTOR is issued requesting confirmation to proceed; if the operator replies N, the processing is terminated; otherwise, the processing continues.

Line 109 issues the JES2 command \$PI to drain the initiators.

Line 110 issues the JES2 commands to reset and drain LOGON1 and LOGON2. The asterisk (*) is used in the command to signify that all active LOGONx interfaces should be drained. The reset command (\$E) forces an immediate drain by requeuing the current activity.

Lines 114-123 handle any remotes that are active. **Line 114** issues the following command:

```
IMFEXEC CMD ' $DU, RMTS, L=Z' RESPONSE($HASP636) WAIT(YES)
```

The \$DU,RMTS command is a JES2 command to display all remotes. The L=Z operand tells JES2 to frame the response. The RESPONSE operand indicates the message ID of the expected command response. In case the system is busy, a WAIT time is specified.

Line 115 sets the initial processing to begin with response line 2. The variable in **line 117** &IMFNOL is set by the EXEC Management application to indicate the number of lines returned to JESDOWN EXEC. The variable &RLJP in **line 116** is set to R to indicate that remotes are being processed. **Line 118** VGETs the line number &N into the variable list RMTS636 in the LOCAL POOL. **Line 119** parses the response to determine the status of the remote; if the status is not INACTIVE, the MUT001C EXEC is called to issue the \$E and \$P commands. The MUT001C EXEC is called because if JESDOWN issued the commands, the response from the \$DU command would be lost from the LOCAL POOL. **Line 122** increments &N by two because the information for each remote spans two response lines. **Lines 117-123** are looped until all the response statements have been processed.

Lines 127-136 handle any lines that are active. **Line 127** issues the following command:

```
IMFEXEC CMD ' $DU, LNES, L=Z' RESPONSE($HASP636) WAIT(YES)
```

The \$DU, LNES command is a JES2 command to display all lines. The L=Z operand tells JES2 to frame the response. The RESPONSE operand indicates the message ID of the expected command response. In case the system is busy, a wait time is specified.

Line 128 sets the initial processing to begin with response line 2. The variable in **line 130**, &IMFNOL, is set by the EXEC Management application to indicate the number of lines returned to JESDOWN EXEC. The variable &RLJP in **line 129** is set to L to indicate that remotes are being processed. **Line 131** VGETs the line number &N into the variable list LNES636 in the LOCAL POOL. **Line 132** parses the response to determine the status of the LINE; if the status is not DRAINED, the MUT001C EXEC is called to issue the \$E and \$P commands. The MUT001C EXEC is called because if JESDOWN issued the commands, the response from the \$DU command would be lost from the LOCAL POOL. **Line 135** increments &N by 1. **Lines 130-136** are looped until all the response statements have been processed.

Lines 140-147 handle jobs which are active. **Line 140** issues the following command:

```
IMFEXEC CMD ' $DA, L=Z' RESPONSE($HASP636) WAIT(YES)
```

The \$DA command is a JES2 command to display all active jobs. The L=Z operand tells JES2 to frame the response. The RESPONSE operand indicates the message ID of the expected command response. In case the system is busy, a wait time is specified.

Line 141 sets the initial processing to begin with response line 2. The variable in **line 143**, &IMFNOL, is set by the EXEC Management application to indicate the number of lines returned to JESDOWN EXEC. The variable &RLJP in **line 142** is set to J to indicate that jobs are being processed. **Line 144** VGETs the line number &N into the variable list ACTS636 in the LOCAL POOL. No parsing of the response is required because only information about active jobs is returned. The MUT001C EXEC is called to issue the \$E and \$C commands. The MUT001C EXEC is called because if JESDOWN issued the commands, the response from the \$DA command would be lost from the LOCAL POOL. **Line 146** increments &N by 1. **Lines 143-147** are looped until all the response statements have been processed.

Lines 151-160 handle any printers that are active. **Line 151** issues the following command:

```
IMFEXEC CMD ' $DU, PRTS, L=Z' RESPONSE($HASP636) WAIT(YES)
```

The \$DU,PRTS command is a JES2 command to display all printers. The L=Z operand tells JES2 to frame the response. The RESPONSE operand indicates the message ID of the expected command response. In case the system is busy, a wait time is specified.

Line 152 sets the initial processing to begin with response line 2. The variable in **line 154**, &IMFNOL, is set by the EXEC Management application to indicate the number of lines returned to JESDOWN EXEC. The variable &RLJP in **line 153** is set to P to indicate that printers are being processed. **Line 155** VGETs the line number &N into the variable list PRTS636 in the LOCAL POOL. **Line 156** parses the response to determine the status of the printer; if the status is not DRAINED, the MUT001C EXEC is called to issue the \$E and \$P commands. The MUT001C EXEC is called because if JESDOWN issued the commands, the response from the \$DU command would be lost from the LOCAL POOL. **Line 159** increments &N by 1. **Lines 154-160** are looped until all the response statements have been processed.

JESDOWN waits for a user-specified interval (**line 165**).

Line 170 issues the WRITELOG command for SYSLOG.

Line 171 issues the \$PJES2 command and waits for the \$HASP607 response. The \$HASP607 message indicates that the system is not dormant, any other message indicates that JES2 termination completed. If the system is not dormant, a WTOR is issued to the Operator (**line 175**) to confirm whether JES2 should be forced down with the command: \$PJES2.ABEND. If the Operator responds Y to the WTOR, JES2 is forced down (**line 179**).

Line 185 calls the RESOLVE Service Utility RSTA to verify the status of JES2. If the return code from RSTA is not zero (**line 186**), the JES2 shut down completed WTO is issued; otherwise, a WTO requesting the initiation of manual procedures is issued and the processing is ended.

```

PROC 1 JESDOWN                                                    1
/*****
/2
/* DOC GROUP(MVS) FUNC(SHUT- DOWN)  CODE(SD)                    */
/* DOC DISP(YES)  AUTHOR(B&B)                                           */
/* DOC DESC(SHUT- DOWN JES2)                                           */
/* -----*/
/* NAME:                                                                */
/*     JESDOWN                                                           */
/* DESCRIPTION:                                                         */
/*     SHUT- DOWN JES2 NORMALLY; IF JES2 DOES NOT COME DOWN          */
/*     GRACEFULLY, ASK THE OPERATOR IF HE WANTS TO FORCE IT          */
/*     WITH CMD:  SPJES2, ABEND                                         */
/* SEE ALSO:                                                            */
/*     MSD001C                                                           */
/*     SHASPO98                                                           */
/*     MUT001C                                                           */
/* INVOKED BY:                                                          */
/*     OPERATOR COMMAND                                                */
/* INPUT PARAMETERS:                                                    */
/*     NONE                                                              */
/* OUTPUT:                                                              */
/*     NONE                                                              */
/* OUTBOARD CALLS:                                                      */
/*     NONE                                                              */
/* EXTERNAL ROUTINES CALLED:                                           */
/*     NONE                                                              */
/* ALERTS ISSUED:                                                       */
/*     NONE                                                              */
/*

```

Figure 16. JESDOWN EXEC

```

/* VARIABLES USED: */
/* NAME: DESCRIPTION: DEFAULT VALUE: */
/* IMFNOL NUMBER OF OUTPUT 0 */
/* LINES FROM COMMAND */40
/* RMTNUM JES2 REMOTE NUMB BLANKS */
/* RMSTAT JES2 REMOTE STATUS BLANKS */
/* LNENUM JES2 LINE NUMBER BLANKS */
/* LNSTAT JES2 LINE STATUS BLANKS */
/* JOBNUM JES2 JOB NUMBER BLANKS */
/* JOBNAME JES2 JOB NAME BLANKS */
/* JBSTAT JES2 JOB STATUS BLANKS */
/* CLASS JES2 JOB CLASS BLANKS */
/* PRI JES2 JOB PRIORITY PRI */
/* (LITERAL) */50
/* PRI# JES2 JOB PRIORITY 0 */
/* (NUMBER) */
/* SYSID JES2 SID PARAMETER BLANKS */
/* PRTNUM JES2 PRINTER NUMB BLANKS */
/* PRTUNIT JES2 PRINTER UNIT BLANKS */
/* PRTSTAT JES2 PRINTER STATUS BLANKS */
/* PJESRESP RESPONSE TO SPJES2 BLANKS */
/* COMMAND */
/* JESWAIT WAIT TIME LIMIT 010 */
/* FOR SYSTEM TO DRAIN */60
/* N USED FOR LOOP 1 */
/* CONTROL */
/* RLPJ PASSED TO MUT001C BLANKS */
/* */
/* TEST SCRIPT INFO: */
/* CHANGE ALL NON-DISPLAY COMMANDS TO MSG AND RUN UNDER A TEST*/
/* SS/TS ENVIRONMENT */
/* */
/* RETURN CODES: */
/* 0 - JES2 SHUT-DOWN COMPLETED SUCCESSFULLY */70
/* */
/* CHANGE LOG: */
/* 04-05-90 BY MDR: ADDED THIS SOLUTION */
/* ***** */

```

Figure 16. JESDOWN EXEC (Continued)

```

/*-----*/
/*   DECLARE WORK AREAS FOR MESSAGE HANDLING   */
/*-----*/
IMFEXEC VDCL RMTS636 LIST(RMTNUM RMSTAT) /* OUTPUT OF RMTS DISPLAY*/
/*-----*/
/*-----*/
IMFEXEC VDCL LNES636 LIST(LNENUM LNSTAT)
80
/*-----*/
/*-----*/
IMFEXEC VDCL ACTS636 LIST(JOBNUM JOBNAME +
                          JBSTAT CLASS PRI PRI# SYSID)
/*-----*/
/*-----*/
IMFEXEC VDCL PRTS636 LIST(PRTNUM PRTUNIT PRTSTAT)
IMFEXEC VDCL PJES2 LIST(PJESRESP) /* RESPONSE TO SPJES2 */
SET &RLPJ = &STR( ) /* INIT VARIABLE */
IMFEXEC VGET JESWAIT PROFILE /* GET USER WAIT TIME LIM*/
/*-----*/
/* ASK THE OPERATOR IF THE JES2 SHUT-DOWN PROCESS SHOULD
CONTINUE*/90
/*-----*/
ASKOPER: +
IMFEXEC WTOR +
'. JESDOWN - REPLY "Y" TO CONTINUE JES2 SHUT-DOWN OR "N" TO ABORT' +
REPLY(RESP)
IF &RESP = &STR(N) THEN DO
IMFEXEC WTO '. JESDOWN - JES2 SHUT-DOWN ABORTED'
GOTO JESDEXIT
END
ELSE DO
IF &RESP NE &STR(Y) THEN DO
IMFEXEC WTO '. JESDOWN - INVALID REPLY, PLEASE REPLY "Y" OR "N"'
GOTO ASKOPER
END
END
/*-----*/
/* ISSUE COMMANDS TO DRAIN INITIATORS, LOGON1 AND LOGON2 */
/*-----*/
IMFEXEC CMD SPI
IMFEXEC CMD ' SPLOGON(*) ; ELOGON(*) '
110
/*-----*/
/* DISPLAY THE REMOTES AND ISSUE SERMT(N) AND SPRMT(N) COMMANDS */
/*-----*/
IMFEXEC CMD ' $DU, RMTS, L=Z' RESPONSE($HASP636) WAIT(20)
SET &N = 2
SET &RLPJ = &STR(R) /* INDICATE THIS IS A RMT */
DO WHILE (&N LE &IMFNOL)
IMFEXEC VGET LINE&N INTO(RMTS636) LOCAL
IF &SUBSTR(24: 31, &STR(&RMSTAT)) NE &STR(INACTIVE) THEN DO
IMFEXEC SELECT EXEC(MUT001C &RMTNUM &RLPJ)
END
SET &N = &N + 2
END
END

```

Figure 16. JESDOWN EXEC (Continued)

```

/*-----*/
/*   DISPLAY THE LINES AND ISSUE RESET/DRAIN COMMANDS   */
/*-----*/
IMFEXEC CMD ' $DU, LNES, L=Z' RESPONSE($HASP636) WAIT(20)
SET &N = 2
SET &RLPJ = &STR(L)                               /* INDICATE THIS IS A LINE*/
DO WHILE (&N LE &IMFNOL)
130
    IMFEXEC VGET LINE&N INTO(LNES636) LOCAL
    IF &SUBSTR(17: 23, &LNSTAT) NE &STR(DRAINED) THEN DO
        IMFEXEC SELECT EXEC(MUT001C &LNENUM &RLPJ)
    END
    SET &N = &N + 1
END
/*-----*/
/*   ISSUE THE DISPLAY ACTIVE COMMAND AND ISSUE RESET/CANCEL CMDS*/
/*-----*/
IMFEXEC CMD ' $DA, L=Z' RESPONSE($HASP636) WAIT(20)
140
SET &N = 2
SET &RLPJ = &STR(J)                               /* INDICATE THIS IS A JOB */
DO WHILE (&N LE &IMFNOL)
    IMFEXEC VGET LINE&N INTO(ACTS636) LOCAL
    IMFEXEC SELECT EXEC(MUT001C &JOBNAME &RLPJ)
    SET &N = &N + 1
END
/*-----*/
/*   DISPLAY THE PRINTERS AND ISSUE RESET/DRAIN COMMANDS   */
/*-----*/
*/150
IMFEXEC CMD ' $DU, PRTS, L=Z' RESPONSE($HASP636) WAIT(20)
SET &N = 2
SET &RLPJ = &STR(P)                               /* INDICATE THIS IS A PRT*/
DO WHILE (&N LE &IMFNOL)
    IMFEXEC VGET LINE&N INTO(PRTS636) LOCAL
    IF &PRTSTAT NE &STR(DRAINED) THEN DO
        IMFEXEC SELECT EXEC(MUT001C &PRTNUM &RLPJ)
    END
    SET &N = &N + 1
END
160
/*-----*/
/*   IF THE USER SPECIFIED A WAIT TIME LIMIT FOR SYSTEM TO FINISH */
/*   DRAINING, NOW IS THE TIME TO USE IT..... */
/*-----*/
IMFEXEC WAIT &JESWAIT                               /* WAIT....ZZZZZZZ..... */

```

Figure 16. JESDOWN EXEC (Continued)

Appendix B. Sample REXX EXECs

REXX has many built-in functions, data queues, and stacks that make it a powerful interpretive language. The BMC Software sample REXX EXECs use some of the features of the REXX Language. The REXX samples, which have been distributed, demonstrate these features. For a complete discussion of the constructs and the syntax of the REXX language, refer to the IBM manuals for the version of REXX installed in your environment.

The following sample EXECs are the basis for the comparison and discussion of the REXX procedural language. Although further customization and changes are *required* to fully utilize these EXECs in a production environment, they are intended to illustrate some of the features of the REXX language.

These sample EXECs for REXX are distributed in the BBPROC data set. BMC Software provides the following REXX Sample EXECs.

REXX Compound Variable Initialization

Set values for variables using the REXX STEMs

REXX DASD Reserve Analysis

Provides immediate insight into contention problems in a shared DASD environment

REXX QUEUE/PULL And VPUT/VGET Usage

Use VPUT/VGET and QUEUE/PULL instructions to store and retrieve data

REXX Compound Variable Initialization EXECs

Two sample EXECs, MSU006X and MSU006X2, are distributed to show initialization of REXX compound variables.

In CLISTs, compound variables are defined with ampersands (&) and the REXX equivalent of ampersands is the STEM variable. The REXX compound variables are defined with STEM where the STEM variable is of the form VAR.n and n is an integer.

The following instructions reference values for SYS1 and SYS2 within a CLIST:

```
IMFEXEC VGET (SYSN) PROFILE
SET &N = 1          /* INITIALIZE THE INDEX VALUE      */
DO WHILE &N LE &SYSN /* SET UP THE LOOP                  */
IMFEXEC VGET (SYS&N) PROFILE
SET &SS = &&SYS&N   /* DERIVE THE VALUE                 */
```

The following set of instructions refer to the same values in a REXX EXEC:

```
ADDRESS IMFEXEC      /* SET IMFEXEC AS THE EXTERNAL ENVIRON */
IMFEXEC VGET (SYSN) PROFILE
DO N=1 TO SYSN BY 1 UNTIL N>SYSN
IMFEXEC VGET ('SYS' N) PROFILE
SS = VALUE('SYS' N) /* DERIVE THE VALUE                   */
```

Notice that with CLISTs, the value of a compound variable must be derived using the && expression, whereas REXX uses the built-in function VALUE.

The apostrophes around SYS define it as a literal to REXX and distinguish the literal from the variable N. If a STEM variable was defined at initialization, this example could have used the following instruction to get the values for SYS. N:

```
IMFEXEC VGET (SYS. N) PROFILE
```

With this instruction, there are no apostrophes to distinguish SYS as a literal because 'SYS.' is a STEM and REXX defaults it to a literal.

In [Figure 17 on page 153](#), note that there is no SET command in REXX. Any expression with an equal sign (=) implies an ASSIGNMENT command that functions the same as a CLIST SET command. MSU006X initializes the SYSN variables for reference by both CLIST and REXX EXECs. MSU006X is a REXX EXEC because the STEM variable, SYS. 1, is not a valid expression within a CLIST.

MSU006X illustrates the assignment of values to both SYS1 and SYS. 1 variables but you must code the values within the EXEC. MSU006X2 illustrates the assignment of values to the STEM variables SYS. n by deriving the values of SYS&n within the context of MAINVIEW AutoOPERATOR initialization.

```
/*-----*/
SYS1="NONE"          /* TARGET SYSTEM(NAME#1) FOR CLIST REFS */
SYS2="NONE"          /* TARGET SYSTEM(NAME#2) FOR CLIST REFS */

SYS. 1="NONE"        /* TARGET SYSTEM(NAME#1) FOR REXX REFS */
SYS. 2="NONE"        /* TARGET SYSTEM(NAME#2) FOR REXX REFS */
/*-----*/
```

Figure 17. REXX EXEC - MSU006X

At MAINVIEW AutoOPERATOR initialization, the values for SYS&n were specified to match the number of target systems within your environment. To build the REXX STEM variables, you can either change the whole start up EXEC, MSU002C or use MSU006X2, which dynamically assigns the values.

EXECs for REXX Compound Variable Initialization

- | | |
|-----------------|---|
| MSU006X | Assigns values for SYS1, SYS2, SYS.1 and SYS.2 statically |
| MSU006X2 | Assigns values for SYS.1 and SYS.2 dynamically by deriving the values for SYS1 and SYS2 |

REXX DASD Reserve Analysis EXEC

The sample REXX DASD Reserve Analysis EXEC is modeled after the CLIST version of the same EXEC. This discussion concentrates on the use of REXX functions and instructions. For the complete discussion of this EXEC, see [“DASD Reserve Analysis Solution” on page 35](#).

For the EXEC to function properly in your environment, you should establish values for the following variables.

Variable	Description
SYSN	Number of MVS systems in the shared DASD configuration
SYS1-SYSn	Names (for example, SMFID) of each system in a shared DASD configuration

This EXEC is initiated by a Rule for the following IOS message:

```
IOS071I  ucb, chpid, jobname, start pending.
```

When REXX EXECs are started based on a TEXT ID, MAINVIEW AutoOPERATOR does not parse the message text prior to scheduling the specified EXEC. The REXX EXEC must do the parsing using the REXX instruction, PARSE.

The following instruction parses the IOS071I message in the distributed EXEC, IOS071IX.

```
PARSE ARG IOS071IX P1 ',' P2 ',' P3 ',' P4 P5 .
```

The PARSE instruction has other keyword operands besides ARG.

You should consult the *TSO Extensions Version 2 Reference* for a full discussion of the PARSE instruction. In this EXEC, the ARG keyword indicates that parsing should be done on the passed parameter strings.

The parameter string contains commas and blanks; in REXX EXECs, the EXEC must specify the parsing pattern. This example has commas in the pattern to subdivide the message. The following values would be placed in the variables:

Variable	Value
IOS071IX	IOS071I
P1	ucb
P2	chpid
P3	jobname
P4	start
P5	pending

Because this discussion does not go beyond the P5 variable, the EXEC has a period (.) as the last variable.

IOS071IX uses the ADDRESS command to inform REXX that non-REXX language instructions should be passed on to the external environment. After a VGET is completed to derive the value for SYSN (the number of shared systems), IOS071IX goes into the following loop:

```

DO N=1 TO SYSN BY 1 UNTIL N>SYSN      /* LOOK AT ALL SYSTEMS */
  IMFEXEC VGET (' SYS' N) PROFILE      /* START GETTING SS IDS*/
  TARG = VALUE(' SYS' N)              /* SET TARGET SYS NAME */
  IF TARG \= IMFORGSS THEN DO         /* DON' T PROCESS THIS */
    ADDRESS IMFEXEC
    IMFEXEC SELECT,                   /* COMMAS ARE USED FOR */
                                     /* CONTINUATION IN REXX*/
    "EXEC(MPE001X" IMFORGSS P1 P3 ") TARGET("TARG")"
  END
END

```

Since REXX does not permit the use of ampersands, this example has apostrophes to distinguish a literal from a variable when the VGET was issued for the SS IDs. Remember that SYSN is the total number of subsystems in a shared DASD complex and ' SYS' N is the subsystem ID. The built-in function VALUE is used to assign TARG; if this value function was not used, the following IF clause would fail every time:

```
IF ' SYS' N = IMFORGSS THEN DO
```

The next instruction could have been used successfully in the IF construction:

```
IF IMFORGSS = VALUE(' SYS' N) THEN DO
```

Because follow-up EXEC MPE001X needs to be scheduled on the TARGET system, it is easier to use the ASSIGNMENT clause to set TARG. Note that when MPE001X is scheduled, double quotes (" ") are used to distinguish the literals from the variables which are then passed to the IMFEXEC environment. If double quotes were not used, REXX would have interpreted EXEC(MPE001X) as a built-in function.

MPE001X is the REXX counterpart to MPE001C. The MPE001X REXX EXEC uses the REXX PARSE and ADDRESS commands along with the VALUE built-in function. MPE001X issues a VGET for IMFRC after the call to the RRES utility and then MPE001X VGETs the number of lines passed back from RRES utility. For this discussion, the following command was added:

```

IF DATATYPE(RRESROL1) 1 = ' NUM' THEN DO /* IS IT NUMERIC? */
  IMFEXEC MSG "' RRESROL1 NOT NUMERIC, RRESROL1="RRESROL1"'"
  SIGNAL MPEXIT /* EXIT */
END

```

The REXX built-in function, DATATYPE, ensures that the number of lines is numeric. The DATATYPE function also can verify other characteristics. The IMFEXEC MSG facility requires the text of the message to be enclosed in single quotes, but within REXX, double and single quotes are interchangeable.

The samples use double quotes to tell REXX that the contents are literals so that REXX passes on the single quotes to the external environment, IMFEXEC. The SIGNAL command of REXX can be used to branch or trap certain conditions. REXX does not have a GOTO command so loops or branches must be controlled by SIGNAL, LEAVE, or CALL commands.

If the UCB in the IOS0711 message has an outstanding reserve, MPE001X schedules the MPE002X EXEC. MPE002X issues an ALERT back to the system requesting the information about outstanding reserves. MPE002X is functionally equivalent to MPE002C.

BMC Software distributes two REXX samples, MPE002X and MPE002X2. Both use the REXX PARSE and ADDRESS instructions and the difference between MPE002X and MPE002X2 is how they build text for ALERTs. MPE002X uses single and double quotes and MPE002X2 uses string concatenations.

An ALERT KEY is built with the following instruction:

```
ALERTKEY=INSERT(SYSID,UCB)
```

Using the REXX INSERT function, a unique ALERT KEY is built. This instruction inserts the value of SYSID before the value of UCB, which is placed subsequently in a variable called ALERTKEY.

MPE002X issues the ALERT with the following instruction:

```
"IMFEXEC ALERT" ALERTKEY,  
". MPE002X JOB "JOBNAME " ON SYSTEM "SYSID" USING "  
UCB" HELD BY SYSTEM "SS" " "COLOR(WHITE)"
```

Note: Commas indicate continuation in REXX.

There are seven pairs of double quotes and one pair of single quotes in the above instruction. The same ALERT is issued with the following set of instructions in MPE002X2.

```
WORK1=" " /* DEFINE LEADING/TRAILING QUOTE */  
WORK2='. MPE002X2 JOB ' /* FIRST PART OF THE MESSAGE */  
  
/* CONCATENATE WORK1, WORK2 AND MORE OF THE TEXT */  
WORK3=WORK1|WORK2|JOBNAME|' ON SYSTEM 'SYSID' USING 'UCB  
  
/* BUILD REST OF THE TEXT WITH THE DELIMITING QUOTE AT END */  
WORK4=', HELD BY SYSTEM '|SS|WORK1  
  
TEXT=INSERT(WORK3,WORK4) /* INSERT WORK3 INFRONT OF WORK4 */  
"IMFEXEC ALERT" ALERTKEY TEXT "COLOR(WHITE)"
```

MPE002X and MPE002X2 are designed to demonstrate the use of quotes and string concatenations in REXX. The choice is really stylistic; both results are the same.

EXECs for REXX DASD Reserve Analysis

IOS071IX	Schedules remote EXECs threshold counters
MPE001X	Examines currently held reserves (runs on remote systems)
MPE002X	Issues ALERT back to requesting system
MPE002X2	Issues ALERT back to requesting system

REXX QUEUE/PULL and VPUT/VGET Usage

Two sample EXECs, MUT002X and MUT003X, are distributed to show the use of REXX QUEUE/PULL and IMFEXEC's VPUT/VGET instructions. As with all REXX EXECs, the PARSE and ADDRESS commands are used. After these commands, MUT002X calls MUT003X to issue the MVS commands, D R, L. When entered, MUT003X establishes the IMFEXEC as the external environment with the ADDRESS instruction.

The following command displays the outstanding replies:

```
"IMFEXEC CMD 'D R, L' RESPONSE(IEE112I) WAIT(20) "
```

Upon return, the response lines are in the local pool. The variable IMFNOL contains the number of lines returned. The variable NUMLINES is ASSIGNED the values of IMFNOL and VPUT to the local pool.

MUT003X places information into the MAINVIEW AutoOPERATOR local pool and the REXX EXTENDED DATA QUEUE with the following set of instructions:

```
DO N=1 TO IMFNOL BY 1 UNTIL N>IMFNOL
  "IMFEXEC VDCL REXXL"N" LIST(TEXT B C D E F) "
  "IMFEXEC VGET LINE"N" INTO(REXXL"N") LOCAL"
  "IMFEXEC VPUT LINE. "N" FROM(REXXL"N") LOCAL"
  QUEUE 'LINE. 'N TEXT B C D E F
END
```

Although 'LINE' . N looks like a REXX STEM, a specific queued line cannot be referenced by the PULL command; therefore the pseudo stem, 'LINE. ' N, becomes a part of the data that queued.

The QUEUE command builds the extended data queue. After the loop is completed, MUT003X issues the RETURN command and processing is continued in MUT002X.

Upon return from MUT003X, MUT002X uses the REXX built-in function QUEUED to establish the number of lines queued and displays this information before setting up the loop to PULL from the extended data queue. The following sequence of instructions is executed:

```
CNT=QUEUED()
IMFEXEC MSG '. MUT002X: THE NUMBER OF QUEUED LINES IS: ' CNT
DO N=1 TO CNT BY 1 UNTIL N>CNT
  PULL LINE TEXT B C D E F
  IMFEXEC MSG '. MUT002X: PULL VARIABLES ARE: ' LINE TEXT B C D E F
END
```

Notice that LINE is considered a part of the data that is queued and not a REXX STEM. REXX discards the data once it has been pulled from the extended data queue. If you want to erase the data queue, build a PULL loop without specifying an argument template. The PULL instruction also can be specified as PARSE PULL.

MUT002X continues execution by getting NUMLINES from the LOCAL POOL and setting up a similar loop to issue VDCL and VGET instructions to derive the values that were stored in the LOCAL POOL by MUT003X.

EXECs for REXX QUEUE/PULL and VPUT/VGET Usage

MUT002X Invokes MUT003X to gather data then VGET the data from the LOCAL POOL and PULL the data from the REX DATA QUEUE

MUT003X Issues the MVS command D R, L and places the data from the response in the LOCAL POOL and the REXX DATA QUEUE

Appendix C. MVS Solution Variables

To ensure that the MVS Solutions function properly in your environment, you must establish values for several variables.

[Table 5](#) lists all MVS Solution variables, their default values, and the solutions each one affects.

Edit member MSU002C in the BBPROC data set, distributed in MAINVIEW AutoOPERATOR, to set values for the appropriate variables.

A Rule for message PM0010I is required to schedule MSU002C upon PAS initialization.

Table 5. MVS Solution Variables

Name	Description	Default	Solutions
DSMM1MIG	ML1 migration in process flag for DASD/Space Monitoring	n	Dump Data Set Monitoring
DUMPCLER	Task to clear dump data sets	None	Dump Data Set Monitoring
DUMPMF	Minimum number of dump data sets to keep clear	1	Dump Data Set Monitoring
DUMPNDS	Total number of dump data sets	4	Dump Data Set Monitoring
LBGCLAS	Class priority list from high to low	MDRAI53	Load Balancing
LBGCPUH	CPU utilization to decrease workload	90	Load Balancing
LBGCPUL	CPU utilization to increase workload	50	Load Balancing
LBGIBEG	Beginning initiator number to manage	None	Load Balancing
LBGIDEC	Number of initiators to stop	1	Load Balancing
LBGIEND	Ending initiator number to manage	None	Load Balancing
LBGIINC	Number of initiators to start	2	Load Balancing
LBGPAGE	Page rate to decrease workload	xxx	Load Balancing
LOGREC	Task name to clear SYS1.LOGREC	None	Logrec Data Set Monitoring
MAXACSA	Maximum extended CSA utilization	80	Performance Management
MAXBCSA	Maximum CSA utilization below the limit	80	Performance Management
MAXCCPU	Maximum complex CPU utilization	95	Performance Management
MAXDEVU	Maximum device utilization	80	Performance Management
MAXTCPU	Maximum TSO CPU utilization	50	Performance Management

Table 5. MVS Solution Variables (Continued)

Name	Description	Default	Solutions
MVSCALL	Name of the support person to be paged; this name must be defined to the Elan Workstation	None	LOGREC Data Set Monitoring VSAM Failures
SMFALT	Suffix of alternate SMF parameters	None	SMF Data Set Monitoring
SMFCLEAR	Task to dump SMF data sets	None	SMF Data Set Monitoring
SYSBEEP	Information to be placed on pager	None	WTO Buffer Monitoring
SYSJES	Version, release, and modification level of JES	SP3.1.3	Solutions that depend upon release-specific messages
SYSMVS	Version, release, and modification level of MVS	SP4.2.2	Solutions that depend upon release-specific messages
SYSN	Number of MVS systems in shared DASD complex	2	DASD Reserve Analysis Shared DASD Control
SYSPROG	TSO USERID of primary system programmer to receive warning messages	None	WTO Buffer Monitoring Logrec Data Set Monitoring
SYS1 - SYSN	System IDs of systems in a shared DASD configuration	None	DASD Reserve Analysis

Appendix D. CICS Solution Variables

To ensure that the CICS Solutions function properly in your environment, you must establish values for several variables.

Table 6 lists all CICS Solution variables, their default values, and the solutions each one affects.

Edit member CSUNIT in the BBPROC data set, distributed with MAINVIEW AutoOPERATOR, to set values for the appropriate variables.

If JRNLEXEC=YES is specified in BBPARM member BBISSP00, MSU002C is automatically scheduled to execute when the PM0010I BBI/PM Initialization Complete message is received. Therefore, it sets the variable values each time the MAINVIEW AutoOPERATOR subsystem is started.

If JRNLEXEC=N is specified, or JRNLEXEC is not specified, a Rule is required for PM0010I.

The CICS Solution variables also can be set by an operator. To reset the CICS Solution variables, enter:

```
%CSUNIT
```

Table 6. CICS Solution Variables

Name	Description	Default	Solutions
FOCALPT	SSID of MAINVIEW AutoOPERATOR subsystem to receive all ALERTs	None	<ul style="list-style-type: none"> Error Recovery Performance Management
ONLCNT01 - ONLCNT0n	Name of the controlling subsystem	None	<ul style="list-style-type: none"> Error Recovery File Degradation Analysis VSAM Control Split Monitor Transaction Response Time Monitor
ONLALT01 - ONLALT0n	Color of ALERTs messages for target system n	RED	<ul style="list-style-type: none"> Error Recovery File Degradation Analysis VSAM Control Split Monitor Transaction Response Time Monitor
ONLTYP01 - ONLTYP0n	Type of target system n or BBI-SSn	None	<ul style="list-style-type: none"> Error Recovery File Degradation Analysis VSAM Control Split Monitor Transaction Response Time Monitor

Table 6. CICS Solution Variables (Continued)

Name	Description	Default	Solutions
ONLALM01 - ONLALM0n	Alarm indicator for ALERT in target system n	Y	<ul style="list-style-type: none"> • Error Recovery • File Degradation Analysis • VSAM Control Split Monitor • Transaction Response Time Monitor
ONLSSN	Number of online subsystems	4	<ul style="list-style-type: none"> • Error Recovery • File Degradation Analysis • VSAM Control Split Monitor • Transaction Response Time Monitor
ONLSS01 - ONLSS0n	Names of the target subsystems	None	<ul style="list-style-type: none"> • Error Recovery • File Degradation Analysis • VSAM Control Split Monitor • Transaction Response Time Monitor

Appendix E. DB2 Solution Variables and Parameters

To ensure that the DB2 Solutions function properly in your environment, you must establish values for several variables and determine threshold parameters.

Data Sets Containing Variables

DB2 Solution variables are found in the following data sets¹:

hilevel.UBBPROC(DSU001C)
hilevel.UBBPROC(DSU101C)
hilevel.UBBPARM(BBIISP00)
hilevel.UBBPARM(BBISSP00)

- | | |
|-----------------|---|
| DSU001C | An EXEC that initializes DB2 variables required for all DB2 Solutions. |
| DSU101C | An EXEC that initializes variables per DB2 subsystem. You can have as many as nine DB2 subsystems. Variables for these additional subsystems are set by copying member DSU101C and renaming the member:

DSU10 <i>n</i> C

where <i>n</i> is the DB2 subsystem (1-9) for which you are setting variables. |
| BBIISP00 | Contains a startup list to which you add BLK members DPE002B and DPE030B during initial customization. |
| BBISSP00 | Contains Rules parameters that you update during initial customization. |

Data Sets Containing Threshold Parameters

DB2 threshold parameters are found in these data sets:

hilevel.UBBPARM(DPE002B)
hilevel.UBBPARM(DPE030B)
hilevel.UBBPARM(DMRBEX00)

- | | |
|-----------------|--|
| DPE002B | DPE002B and DPE030B contain parameters that specify the names of monitors you add during initial customization of the DB2 Solutions. |
| DPE030B | See DEP002B. |
| DMRBEX00 | Contains parameters for setting threshold levels in the DB2RNWY Solution during initial customization of the DB2 Solutions. |

¹ See the *MAINVIEW Common Customization Guide* for information on the installation of UBBPARM and UBBPROC data sets.

Tables That Describe Variables and Parameters

Variables and threshold parameters for the DB2 Solutions are organized in tables according to the solutions each one affects:

All DB2 Solutions	Table 7 on page 165
DB2DLOK	Table 8 on page 166
DB2RESP Variables	Table 9 on page 167
DB2RESP Thresholds	Table 10 on page 169
DB2RNWY	Table 11 on page 169
DB2TFUL	Table 12 on page 170
DB2THRD	Table 13 on page 171

Note: You must set the following two variables listed in the **All DB2 Solutions** (see [Table 7 on page 165](#)):

- DB2
- D_DB2NUM

These variables are part of the initial environment set up and it is mandatory that you set both of them before using any of the DB2 Solutions.

The rest of the variables found in the following five tables are optional, and defaults are provided for them.

All DB2 Solutions

Table 7. Variables for All DB2 Solutions in hilevel.UBBPROC

Variable Name	Description	Default	Data Set Member
D_DB2NUM ^A	Number of DB2s monitored by this subsystem	1	DSU001C
D_ALERT_IDS	USERIDs to be notified of DB2 Alerts	Null	DSU001C
D_ALERT_Q	Queue name to use for Alerts	DB2	DSU001C
D_ALERT_SEND	Indicates (Y N) that SEND command is to be issued	N	DSU001C
D_ALERT_SENDOPT	SEND option: either LOGON or NOW	NOW	DSU001C
DB2 ^A	Name you supply to identify each DB2 subsystem	DB2D	DSU101C
D_ALERT_TGT	Target for MAINVIEW Alerts	&IMFORGSS	DSU001C
MSTR	DB2 MSTR address space name associated with this DB2 (for example, DB2AMSTR, DSNBMSTR, and so on)	&DB2.MSTR	DSU101C
DBM1	DB2 DBM1 address space name associated with this DB2 (for example, DB2ADB1, DSNBDBM1, and so on)	&DB2.DBM1	DSU101C
DIST	DB2 DIST address space name associated with this DB2 if applicable at your site (for example DB2ADIST, DSNBDIST, and so on)	&DB2.DIST	DSU101C
IRLM	IRLM PROC name associated with this DB2 subsystems (for example DB2PROC, IRLMPROC, and so on)	&DB2.PROC	DSU101C
IMST	Optional IMS target name associated with this DB2 subsystem (null if not applicable)	Null	DSU101C

^a It is mandatory that you set this variable

DB2DLOK

Table 8. DB2DLOK Variables in hilevel.UBBPROC

Variable Name	Description	Default	Data Set Member
CNT	Total number of deadlock/timeout history entries to be maintained in the dynamic Deadlock History Table for this DB2 subsystem	20	DSU101C
INTVL	Length of time in seconds to wait between LOCKD displays; this should be half your IRLM deadlock time	5	DSU101C
MAX_AGE	Discard deadlock events over this age (rolling time interval in minutes that a particular plan involved in deadlock/time can be considered a candidate for analysis for this DB2 subsystem)	30 minutes	DSU101C
MAXEITHER	Maximum number of times that a plan can be EITHER a lock owner or a lock requestor in a deadlock/timeout within the MAX_AGE time interval before triggering the Deadlock Analysis Solution	3	DSU101C
MAXOWNER	Maximum number of times that a plan can be a lock OWNER in a deadlock/timeout within the MAX_AGE time interval before triggering the Deadlock Analysis Solution	2	DSU101C
MAXREQ	Maximum number of times that a plan can be a lock REQUESTOR in a deadlock/timeout within the MAX_AGE time interval before triggering the Deadlock Analysis/Timeout Solution	2	DSU101C
STOPTIME	Length of time in minutes to run the LOCKD data extractor to analyze lock contentions	10 minutes	DSU101C

DB2RESP Variables

Table 9. DB2RESP Variables in hilevel.UBBPROC

Variable Name	Description	Default	Data Set Member
MAX_TSOELAP	Maximum elapsed time per TSO user	00:01:00 (1 minute)	DSU101C
MAX_TSOSTMT	Maximum number of SQL statements issued per TSO user	500	DSU101C
MAX_TSOLOCK	Maximum number of LOCKS held per TSO user	500	DSU101C
MAX_BATELAP	Maximum elapsed time per BATCH transaction	01:00:00 (1 hour)	DSU101C
MAX_BATSTMT	Maximum number of SQL statements issued per BATCH transaction	100,000	DSU101C
MAX_BATLOCK	Maximum number of LOCKs held per BATCH transaction	2,000	DSU101C
MAX_UTLELAP	Maximum elapsed time per UTILITY transaction	00:30:00 (30 minutes)	DSU101C
MAX_UTLSTMT	Maximum number of SQL statements issued per UTILITY transaction	0	DSU101C
MAX_UTLLOCK	Maximum number of LOCKs held per UTILITY transaction	50	DSU101C
MAX_CAFELAP	Maximum elapsed time per CAF transaction	00:30:00 (30 minutes)	DSU101C
MAX_CAFSTMT	Maximum number of SQL statements issued per CAF transaction	100,000	DSU101C
MAX_CAFLOCK	Maximum number of LOCKs held per CAF transaction	2,000	DSU101C
MAX_TPELAP	Maximum elapsed time per CICS or IMS transaction	00:00:05 (5 seconds)	DSU101C
MAX_TPSTMT	Maximum number of SQL statements issued per CICS or IMS transaction	100	DSU101C
MAX_TPLOCK	Maximum number of LOCKs held per CICS or IMS transaction	100	DSU101C
MAX_EDMFAILS	Maximum number of EDM pool failures per minute	0	DSU101C
MAX_EDMLOADS	Maximum percent of EDM pool requests needing loads	20%	DSU101C
MAXBFR_UNAVAIL	Maximum number of buffer unavailable conditions per minute	0	DSU101C
MAXALLOC_DELAY	Maximum number of allocation delay conditions per minute	0	DSU101C
MAX_RIDUT	Maximum RID pool percent utilization	80%	DSU101C

Table 9. DB2RESP Variables in hilevel.UBBPROC (Continued)

Variable Name	Description	Default	Data Set Member
MAX_BFREXP	Maximum number of buffer pool expansions per minute	0	DSU101C
MAX_BFRDMC	Maximum number of Data Manager Critical conditions per minute	0	DSU101C
MAX_BFRIW	Maximum number of Immediate Write conditions per minute	0	DSU101C
MIN_GETRIO	Minimum number of GETPAGES per read I/O in the INTVL	2.0	DSU101C
MIN_SWSPWS	Minimum number of system page updates per system pages written in the INTVL	1.0	DSU101C
MIN_PWSWIO	Minimum number of system pages written per write I/O in the INTVL	5.0	DSU101C
RESP_MIN	Number of minutes between monitor warning messages before triggering the EXEC again if necessary	1	DSU101C
MAX_SUSPCT	Percent suspended threads (total active threads divided by total suspended threads)	25%	DSU101C
MAX_THDUTL	Thread utilization (total active threads divided by maximum threads allowed)	90%	DSU101C
MAX_#SUSP	Maximum suspensions per active thread per minute (suspensions per minute divided by number of active threads)	1	DSU101C
MAX_#DEAD	Maximum deadlocks per minute in the interval	0	DSU101C
MAX_#TIMEOUT	Maximum timeouts per minute in the interval	0	DSU101C
MAX_PAGEPCT	Maximum DB2 paging as a percent of total system (DB2 paging divided by total system paging)	10%	DSU101C
MAX_EDMUT	Maximum EDM pool utilization	80%	DSU101C
MAX_OPENPCT	Maximum open data set utilization	90%	DSU101C

DB2RESP Monitor Thresholds

Table 10. Monitor Thresholds for DB2RESP in hilevel.UBBPARAM

Monitor	Connection	Description	Threshold	Data Set Member
@ELAP	IMS	Average elapsed time	5 seconds	DPE002B
@ELPD	IMS	Average elapsed time in DB2	3 seconds	DPE002B
@CPU	IMS	Average CPU time	0.5 seconds	DPE002B
@CPUD	IMS	Average CPU time in DB2	0.3 seconds	DPE002B
@ELAP	TSO	Average elapsed time	15 seconds	DPE002B
@ELPD	TSO	Average elapsed time in DB2	5 seconds	DPE002B
@CPU	TSO	Average CPU time	1 second	DPE002B
@CPUD	TSO	Average CPU time in DB2	0.5 seconds	DPE002B
@ELAP	CICS	Average elapsed time	5 seconds	DPE002B
@ELPD	CICS	Average elapsed time in DB2	3 seconds	DPE002B
@CPU	CICS	Average CPU time	0.5 seconds	DPE002B
@CPUD	CICS	Average CPU time in DB2	0.3 seconds	DPE002B
@CPU	BATCH	Average CPU time	0.5 seconds	DPE002B
@CPUD	BATCH	Average CPU time in DB2	0.3 seconds	DPE002B

DB2RNWY

Table 11. DB2RNWY Threshold Parameters in hilevel.UBBPARAM

Keyword	Connection	Threshold Parameter	Data Set Member
MSG=DZ0610W	IMS	GPRATE=200 CYCLES=3	DMRBEX00
MSG=DZ0620W	CICS	GPRATE=200 CYCLES=3	DMRBEX00
MSG=DZ0630W	TSO	GPRATE=200 CYCLES=3	DMRBEX00
MSG=DZ0640W	BATCH	GPRATE=500 CYCLES=3	DMRBEX00
MSG=DZ0650W	CAF	GPRATE=500 CYCLES=3	DMRBEX00
MSG=DZ0660W	UTILITY	GPRATE=1000 CYCLES=3	DMRBEX00

DB2TFUL

Table 12. DB2TFUL Variables in hilevel.UBBPROC

Variable Name	Description	Default	Data Set Member
MAX_UTIL	Default maximum data base/tablespace (DB/TS) utilization for all DB/TSs associated with this subsystem; zero means no default threshold This value can be overridden for specific DB/TSs within each DB2 subsystem.	0	DSU101C
MAX_EXTS	Default maximum number of extents for all DB/TSs associated with this DB2 subsystem This value can be overridden for specific DB/TSs within each DB2 subsystem.	0	DSU101C
NUM	Total number of overrides for this DB2 subsystem	4	DSU101C
NAME _n	Nth DBTS name to be overridden for percent space utilization and extent monitoring for this DB2 subsystem Generic DBTS names may be used.	NA	DSU101C
NAME1	First DBTS name	DSN*	DSU101C
NAME2	Second DBTS name	DSNDB06_ DSNAPH01	DSU101C
NAME3	Third DBTS name	DSNDB07_ DSN4K01	DSU101C
NAME4	Fourth DBTS name	DSNDB06_ SYS*	DSU101C
MAX_UTIL _n	Utilization threshold for all DB/TSs that satisfy the NAME _n qualification Zero means ignore percent utilization.	NA	DSU101C
MAX_UTIL1	Utilization threshold for NAME1	0	DSU101C
MAX_UTIL2	Utilization threshold for NAME2	90	DSU101C
MAX_UTIL3	Utilization threshold for NAME3	50	DSU101C
MAX_UTIL4	Utilization threshold for NAME4	50	DSU101C
MAX_EXTS _n	Extent threshold for all DB/TS's that satisfy the NAME _n qualification Zero means ignore percent utilization.	NA	DSU101C
MAX_EXTS1	Extent override for NAME1	0	DSU101C
MAX_EXTS2	Extent override for NAME2	1	DSU101C

Table 12. DB2TFUL Variables in hilevel.UBBPROC (Continued)

Variable Name	Description	Default	Data Set Member
MAX_EXTS3	Extent override for NAME3	20	DSU101C
MAX_EXTS4	Extent override for NAME4	1	DSU101C

DB2THRD

Table 13. DB2TFUL Variables in hilevel.UBBPROC

Variable Name	Description	Default	Data Set Member
Q_VERIFY	Indicates (Y N) whether or not to issue an ALERT to verify the TSO cancel	Y	DSU101C

Part 7. Glossary and Index

This section contains a glossary of terms and an index.

Glossary175

Index187

Glossary

This glossary defines BMC Software terminology. Other dictionaries and glossaries can be used in conjunction with this glossary.

Since this glossary pertains to BMC Software-related products, some of the terms defined might not appear in this book.

To help you find the information you need, this glossary uses the following cross-references:

Contrast with	Indicates a term that has a contrary or contradictory meaning.
See	Indicates an entry that is a synonym or contains expanded information.
See also	Indicates an entry that contains related information.

A

action. Defined operation, such as modifying a MAINVIEW window, that is performed in response to a command. *See* object.

active window. Any MAINVIEW window in which data can be refreshed. *See* alternate window, current window, window.

administrative view. Display from which a product's management tasks are performed, such as the DSLIST view for managing historical data sets. *See* view.

ALT WIN field. Input field that allows you to specify the window identifier for an alternate window where the results of a hyperlink are displayed. *See* alternate window.

Alternate Access. *See* MAINVIEW Alternate Access.

alternate form. View requested through the FORM command that changes the format of a previously displayed view to show related information. *See also* form, query.

alternate window. (1) Window that is specifically selected to display the results of a hyperlink. (2) Window whose identifier is defined to the ALT WIN field. *Contrast with* current window. *See* active window, window, ALT WIN field.

analyzer. (1) Online display that presents a snapshot of status and activity data and indicates problem areas. (2) Component of CMF MONITOR. *See* CMF MONITOR Analyzer.

application. (1) Program that performs a specific set of tasks within a MAINVIEW product. (2) In MAINVIEW VistaPoint, combination of workloads to enable display of their transaction performance data in a single view.

application trace. *See* trace.

ASCH workload. Workload comprising Advanced Program-to-Program Communication (APPC) address spaces.

AutoCustomization. Online facility for customizing the installation of products. AutoCustomization provides an ISPF panel interface that both presents customization steps in sequence and provides current status information about the progress of the installation.

automatic screen update. Usage mode wherein the currently displayed screen is refreshed automatically with new data at an interval you specify. Invoked by the ASU command.

B

batch workload. Workload consisting of address spaces running batch jobs.

BBI. Basic architecture that distributes work between workstations and multiple OS/390 targets for BMC Software MAINVIEW products.

BBI-SS PAS. *See* BBI subsystem product address space.

BBI subsystem product address space (BBI-SS PAS). OS/390 subsystem address space that manages communication between local and remote systems and that contains one or more of the following products:

- Command MQ for S/390
- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for DB2
- MAINVIEW for DBCTL
- MAINVIEW for IMS Online
- MAINVIEW for MQSeries
- MAINVIEW SRM
- MAINVIEW VistaPoint (for CICS, DB2, DBCTL, and IMS workloads)

BBPARM. *See* parameter library.

BBPROC. *See* procedure library.

BBPROF. *See* profile library.

BBSAMP. *See* sample library.

BBV. *See* MAINVIEW Alternate Access.

BBXS. BMC Software Subsystem Services. Common set of service routines loaded into common storage and used by several BMC Software MAINVIEW products.

border. Visual indication of the boundaries of a window.

bottleneck analysis. Process of determining which resources have insufficient capacity to provide acceptable service levels and that therefore can cause performance problems.

C

CA-Disk. Data management system by Computer Associates that replaced the DMS product.

CAS. Coordinating address space. One of the address spaces used by the MAINVIEW windows environment architecture. The CAS supplies common services and enables communication between linked systems. Each OS/390 or z/OS image requires a separate CAS. Cross-system communication is established through the CAS using VTAM and XCF communication links.

CFMON. *See* coupling facility monitoring.

chart. Display format for graphical data. *See also* graph.

CICSplex. User-defined set of one or more CICS systems that are controlled and managed as a single functional entity.

CMF MONITOR. Comprehensive Management Facility MONITOR. Product that measures and reports on all critical system resources, such as CPU, channel, and device usage; memory, paging, and swapping activity; and workload performance.

CMF MONITOR Analyzer. Batch component of CMF MONITOR that reads the SMF user and 70 series records created by the CMF MONITOR Extractor and/or the RMF Extractor and formats them into printed system performance reports.

CMF MONITOR Extractor. Component of CMF that collects performance statistics for CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390, and RMF postprocessor. *See* CMF MONITOR Analyzer, CMF MONITOR Online, MAINVIEW for OS/390.

CMF MONITOR Online. Component of CMF that uses the MAINVIEW window interface to present data on all address spaces, their use of various system resources, and the delays that each address space incurs while waiting for access to these resources. *See* CMF MONITOR, MAINVIEW for OS/390.

CMF Type 79 API. Application programming interface, provided by CMF, that provides access to MAINVIEW SMF-type 79 records.

CMFMON. Component of CMF MONITOR that simplifies online retrieval of information about system hardware and application performance and creates MAINVIEW SMF-type 79 records.

The CMFMON *online facility* can be used to view data in one or more formatted screens.

The CMFMON *write facility* can be used to write collected data as MAINVIEW SMF-type 79 records to an SMF or sequential data set.

CMRDETL. MAINVIEW for CICS data set that stores detail transaction records (type 6E) and abend records (type 6D). Detail records are logged for each successful transaction. Abend records are written when an abend occurs. Both records have the same format when stored on CMRDETL.

CMRSTATS. MAINVIEW for CICS data set that stores both CICS operational statistic records, at five-minute intervals, and other records, at intervals defined by parameters specified during customization (using CMRSOPT).

column. Vertical component of a view or display, typically containing fields of the same type of information, that varies by the objects associated in each row.

collection interval. Length of time data is collected. *See also* delta mode, total mode.

command delimiter. Special character, usually a ; (semicolon), used to stack commands typed concurrently on the COMMAND line for sequential execution.

COMMAND line. Line in the control area of the display screen where primary commands can be typed. *Contrast with* line command column.

Command MQ Automation D/S. Command MQ agents, which provide local proactive monitoring for both MQSeries and MSMQ (Microsoft message queue manager). The Command MQ agents operate at the local node level where they continue to perform functions regardless of the availability of the MQM (message queue manager) network. Functionality includes automatic monitoring and restarts of channels, queue managers, queues and command servers. In cases where automated recovery is not possible, the agents transport critical alert information to a central console.

Command MQ Automation S/390. Command MQ component, which monitors the MQM (message queue manager) networks and intercedes to perform corrective actions when problems arise. Solutions include:

- Dead-Letter Queue management
- System Queue Archival
- Service Interval Performance solutions
- Channel Availability

These solutions help ensure immediate relief to some of the most pressing MQM operations and performance problems.

Command MQ for D/S. Command MQ for D/S utilizes a true client/server architecture and employs resident agents to provide configuration, administration, performance monitoring and operations management for the MQM (message queue manager) network.

Command MQ for S/390. See MAINVIEW for MQSeries.

COMMON STORAGE MONITOR. Component of MAINVIEW for OS/390 that monitors usage and reconfigures OS/390 or z/OS common storage blocks.

composite workload. Workload made up of a WLM workload or other workloads, which are called *constituent workloads*.

constituent workload. Member of a composite workload. Constituent workloads in a composite usually belong to a single workload class, but sometimes are mixed.

contention. Occurs when there are more requests for service than there are servers available.

context. In a Plex Manager view, field that contains the name of a target or group of targets specified with the CONTEXT command. See scope, service point, SSI context, target context.

CONTEXT command. Specifies either a MAINVIEW product and a specific target for that product (see target context) or a MAINVIEW product and a name representing one or more targets (see *SSI context*) for that product.

control statement. (1) Statement that interrupts a sequence of instructions and transfers control to another part of the program. (2) Statement that names samplers and other parameters that configure the MAINVIEW components to perform specified functions. (3) In CMF MONITOR, statement in a parameter library member used to identify a sampler in the extractor or a report in the analyzer, or to describe either component's processing requirements to the operating system.

coupling facility monitoring (CFMON). Coupling facility views that monitor the activity of your system's coupling facilities.

current data. Data that reflects the system in its current state. The two types of current data are realtime data and interval data. Contrast with historical data. See also interval data and realtime data.

current window. In the MAINVIEW window environment, window where the main dialog with the application takes place. The current window is used as the default window destination for commands issued on the COMMAND line when no window number is specified. Contrast with alternate window. See active window, window.

D

DASD. Direct Access Storage Device. (1) A device with rotating recording surfaces that provides immediate access to stored data. (2) Any device that responds to a DASD program.

data collector. Program that belongs to a MAINVIEW product and that collects data from various sources and stores the data in records used by views. For example, MAINVIEW for OS/390 data collectors obtain data from OS/390 or z/OS services, OS/390 or z/OS control blocks, CMF MONITOR

Extractor control blocks, and other sources. Contrast with extractor.

delta mode. (1) In MAINVIEW for DB2 analyzer displays, difference between the value sampled at the start of the current statistics interval and the value sampled by the current analyzer request. See also *statistics interval*. (2) In CMFMON, usage mode wherein certain columns of data reflect the difference in values between one sample cycle and the next. Invoked by the DELTA ON command. See also collection interval, sample cycle, total mode.

DFSMS. Data Facility Storage Management System. Data management, backup, and HSM software from IBM for OS/390 or z/OS mainframes.

DMR. See MAINVIEW for DB2.

DMS. Data Management System. See CA-Disk.

DMS2HSM. See MAINVIEW SRM DMS2HSM.

DSO. Data Set Optimizer. CMF MONITOR Extractor component that uses CMF MONITOR Extractor data to produce reports specifying the optimal ordering of data sets on moveable head devices.

E

EasyHSM. See MAINVIEW SRM EasyHSM.

EasyPOOL. See MAINVIEW SRM EasyPOOL.

EasySMS. See MAINVIEW SRM EasySMS.

element. (1) Data component of a data collector record, shown in a view as a field. (2) Internal value of a field in a view, used in product functions.

element help. Online help for a field in a view. The preferred term is *field help*.

Enterprise Storage Automation. See MAINVIEW SRM Enterprise Storage Automation.

event. A message issued by Enterprise Storage Automation. User-defined storage occurrences generate events in the form of messages. These events provide an early warning system for storage problems and are routed to user-specified destinations for central viewing and management.

Event Collector. Component for MAINVIEW for IMS Online, MAINVIEW for IMS Offline, and MAINVIEW for DBCTL that collects data about events in the IMS environment. This data is required for Workload Monitor and optional for Workload Analyzer (except for the workload trace service). This data also is recorded as transaction records (X'FA') and program records (X'F9') on the IMS system log for later use by the MAINVIEW for IMS Offline components: Performance Reporter and Transaction Accountant.

expand. Predefined link from one display to a related display. See also hyperlink.

extractor. Program that collects data from various sources and keeps the data control blocks to be written as records. Extractors obtain data from services, control blocks, and other sources. *Contrast with* data collector.

extractor interval. *See* collection interval.

F

fast path. Predefined link between one screen and another. To use the fast path, place the cursor on a single value in a field and press Enter. The resulting screen displays more detailed information about the selected value. *See also* hyperlink.

field. Group of character positions within a screen or report used to type or display specific information.

field help. Online help describing the purpose or contents of a field on a screen. To display field help, place the cursor anywhere in a field and press PF1 (HELP). In some products, field help is accessible from the screen help that is displayed when you press PF1.

filter. Selection criteria used to limit the number of rows displayed in a view. Data that does not meet the selection criteria is not displayed. A filter is composed of an element, an operator, and an operand (a number or character string). Filters can be implemented in view customization, through the PARM/QPARM commands, or through the Where/QWhere commands. Filters are established against elements of data.

fire. The term used to indicate that an event has triggered an action. In MAINVIEW AutoOPERATOR, when a rule selection criteria matches an incoming event and *fires*, the user-specified automation actions are performed. This process is also called *handling* the event.

fixed field. Field that remains stationary at the left margin of a screen that is scrolled either right or left.

FOCAL POINT. MAINVIEW product that displays a summary of key performance indicators across systems, sites, and applications from a single terminal.

form. One of two constituent parts of a view; the other is query. A form defines how the data is presented; a query identifies the data required for the view. *See also* query, view.

full-screen mode. Display of a MAINVIEW product application or service on the entire screen. There is no window information line. *Contrast with* windows mode.

G

global command. Any MAINVIEW window interface command that can affect all windows in the window area of a MAINVIEW display.

graph. Graphical display of data that you select from a MAINVIEW window environment view. *See also* chart.

H

hilevel. For MAINVIEW products, high-level data set qualifier required by a site's naming conventions.

historical data. (1) Data that reflects the system as it existed at the end of a past recording interval or the duration of several intervals. (2) Any data stored in the historical database and retrieved using the TIME command. *Contrast with* current data, interval data and realtime data.

historical database. Collection of performance data written at the end of each installation-defined recording interval and containing up to 100 VSAM clusters. Data is extracted from the historical database with the TIME command. *See* historical data.

historical data set. In MAINVIEW products that display historical data, VSAM cluster file in which data is recorded at regular intervals.

HSM. (Hierarchical Storage Management) Automatic movement of files from hard disk to slower, less-expensive storage media. The typical hierarchy is from magnetic disk to optical disk to tape.

hyperlink. (1) Preset field in a view or an EXPAND line on a display that permits you to

- Access cursor-sensitive help
- Issue commands
- Link to another view or display

The transfer can be either within a single product or to a related display/view in a different BMC Software product. Generally, hyperlinked fields are highlighted. (2) Cursor-activated short path from a topic or term in online help to related information. *See also* fast path.

I

Image log. Collection of screen-display records. Image logs can be created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Image log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Image log stops when both data sets are filled and the first data set is not processed by the archive program.

The TS Image log is a single data set that wraps around when full.

IMSplex System Manager (IPSM). MVIMS Online and MVDBC service that provides Single System Image views of resources and bottlenecks for applications across one or more IMS regions and systems.

interval data. Cumulative data collected during a collection interval. Intervals usually last from 15 to 30 minutes depending on how the recording interval is specified during product customization. *Contrast with* historical data.

Note: If change is made to the workloads, a new interval will be started.

See also current data and realtime data.

InTune. Product for improving application program performance. It monitors the program and provides information used to reduce bottlenecks and delays.

IRUF. IMS Resource Utilization File (IRUF). IRUFs can be either detail (one event, one record) or summarized (more than one event, one record). A detail IRUF is created by processing the IMS system log through a program called IMFLEDIT. A summarized IRUF is created by processing one or more detail IRUFs, one or more summarized IRUFs, or a combination of both, through a sort program and the TASCOSTR program.

J

job activity view. Report about address space consumption of resources. *See* view.

journal. Special-purpose data set that stores the chronological records of operator and system actions.

Journal log. Collection of messages. Journal logs are created for both the BBI-SS PAS and the BBI terminal session (TS).

The BBI-SS PAS Journal log consists of two data sets that are used alternately: as one fills up, the other is used. Logging to the BBI-SS PAS Journal log stops when both data sets are filled and the first data set is not being processed by the archive program.

The TS Journal log is a single data set that wraps around when full.

L

line command. Command that you type in the line command column in a view or display. Line commands initiate actions that apply to the data displayed in that particular row.

line command column. Command input column on the left side of a view or display. *Contrast with* COMMAND line.

Log Edit. In the MAINVIEW for IMS Offline program named IMFLEDIT, function that extracts transaction (X'FA') and program (X'F9') records from the IMS system log. IMFLEDIT also extracts certain records that were recorded on the system log by IMS. IMFLEDIT then formats the records into a file called the IMS Resource Utilization File (IRUF).

M

MAINVIEW. BMC Software integrated systems management architecture.

MAINVIEW Alarm Manager. In conjunction with other MAINVIEW products, notifies you when an exception condition occurs. MAINVIEW Alarm Manager is capable of monitoring multiple systems simultaneously, which means that MAINVIEW Alarm Manager installed on one system keeps track of your entire sysplex. You can then display a single view that show exceptions for all MAINVIEW performance monitors within your OS/390 or z/OS enterprise.

MAINVIEW Alternate Access. Enables MAINVIEW products to be used without TSO by providing access through EXCP and VTAM interfaces.

MAINVIEW Application Program Interface. REXX- or CLIST-based, callable interface that allows MAINVIEW AutoOPERATOR EXECs to access MAINVIEW monitor product view data.

MAINVIEW AutoOPERATOR. Product that uses tools, techniques, and facilities to automate routine operator tasks and provide online performance monitoring, and that achieves high availability through error minimization, improved productivity, and problem prediction and prevention.

MAINVIEW control area. In the MAINVIEW window environment, first three lines at the top of the view containing the window information line and the COMMAND, SCROLL, CURR WIN, and ALT WIN lines. The control area cannot be customized and is part of the information display. *Contrast with* MAINVIEW display area, MAINVIEW window area.

MAINVIEW display area. *See* MAINVIEW window area.

MAINVIEW Explorer. Product that provides access to MAINVIEW products from a Web browser running under Windows. MAINVIEW Explorer replaces MAINVIEW Desktop.

MAINVIEW for CICS. Product (formerly MV MANAGER for CICS) that provides realtime application performance analysis and monitoring for CICS system management.

MAINVIEW for DB2. Product (formerly MV MANAGER for DB2) that provides realtime and historical application performance analysis and monitoring for DB2 subsystem management.

MAINVIEW for DBCTL. Product (formerly MV MANAGER for DBCTL) that provides realtime application performance analysis and monitoring for DBCTL management.

MAINVIEW for IMS (MVIMS) Offline. Product with a Performance Reporter component that organizes data and prints reports used to analyze IMS performance and a Transaction Accountant component that produces cost accounting and user charge-back records and reports.

MAINVIEW for IMS (MVIMS) Online. Product that provides realtime application performance analysis and monitoring for IMS management.

MAINVIEW for IP. Product that monitors OS/390 and z/OS mission-critical application performance as it relates to TCP/IP stack usage. Collected data includes availability, connections, response times, routers, service levels, storage, traffic, Web cache, and so on.

MAINVIEW for Linux-Servers. Product that allows you to monitor the performance of your Linux systems from the MAINVIEW windows interface.

MAINVIEW for MQSeries. Delivers comprehensive capabilities for configuration, administration, performance monitoring and operations management for an entire MQM (message queue manager) network.

MAINVIEW for OS/390. System management application (known as MAINVIEW for MVS prior to version 2.5). Built upon the MAINVIEW window environment architecture, it uses the window interface to provide access to system performance data and other functions necessary in the overall management of an enterprise.

MAINVIEW for UNIX System Services. System management application that allows you to monitor the performance of the Unix System Services from a MAINVIEW window interface.

MAINVIEW for VTAM. Product that displays application performance data by application, transaction ID, and LU name. This collected data includes: connections, response time statistics, application availability, and application throughput.

MAINVIEW for WebSphere. Product that provides Web monitoring and management for applications integrated with IBM WebSphere Application Server for OS/390 or z/OS.

MAINVIEW Selection Menu. ISPF selection panel that provides access to all MAINVIEW windows-mode and full-screen mode products.

MAINVIEW SRM. *See* MAINVIEW Storage Resource Manager (SRM).

MAINVIEW SRM DMS2HSM. Product that facilitates the conversion of CA-Disk, formerly known as DMS, to HSM.

MAINVIEW SRM EasyHSM. Product that provides online monitoring and reporting to help storage managers use DFHSM efficiently.

MAINVIEW SRM EasyPOOL. Product that provides control over data set allocation and enforcement of allocation and naming standards. EasyPOOL functions operate at the operating system level to intercept normal job processing, thus providing services without any JCL changes.

MAINVIEW SRM EasySMS. Product that provides tools that aid in the conversion to DFSMS and provides enhancement to the DFSMS environment after implementation. EasySMS consists of the EasyACS functions, the SMSACSTE function, and the Monitoring and Positioning Facility.

MAINVIEW SRM Enterprise Storage Automation. Product that delivers powerful event generation and storage automation technology across the storage enterprise. Used in conjunction with MAINVIEW AutoOPERATOR, automated solutions to perform pool, volume, application, or data set-level manipulation can be created and used in response to any condition or invoked to perform ad hoc requests

MAINVIEW SRM SG-Auto. Product that provides early warning notification of storage anomalies and automated responses to those anomalies based on conditions in the storage subsystem.

MAINVIEW SRM SG-Control. Product that provides real-time monitoring, budgeting, and control of DASD space utilization.

MAINVIEW SRM StopX37/II. Product that provides enhancements to OS/390 or z/OS space management, reducing the incidence of space-related processing problems. The StopX37/II functions operate at the system level to interceptabend conditions or standards violations, thus providing services without any JCL changes.

MAINVIEW SRM StorageGUARD. Product that monitors and reports on DASD consumption and provides historical views to help control current and future DASD usage.

MAINVIEW Storage Resource Manager (SRM). Suite of products that assists in all phases of OS/390 or z/OS storage management. MAINVIEW SRM consists of products that perform automation, reporting, trend analysis, and error correction for storage management.

MAINVIEW SYSPROG Services. *See* SYSPROG Services.

MAINVIEW VistaPoint. Product that provides enterprise-wide views of performance. Application and workload views are available for CICS, DB2, DBCTL, IMS, and OS/390. Data is summarized at the level of detail needed; for example, views can be for a single target, an OS/390 or z/OS image, or an entire enterprise.

MAINVIEW window area. Portion of the information display that is not the control area and in which views are displayed and windows opened. It includes all but the first three lines of the information display. *Contrast with* MAINVIEW control area.

monitor. Online service that measures resources or workloads at user-defined intervals and issues warnings when user-defined thresholds are exceeded.

Multi-Level Automation (MLA). The user-defined, multiple step process in Enterprise Storage Automation that implements solutions in a tiered approach, where solutions are invoked one after another until the condition is resolved.

MVALARM. *See* MAINVIEW Alarm Manager.

MVAPI. *See* MAINVIEW Application Program Interface.

MVCICS. *See* MAINVIEW for CICS.

MVDB2. *See* MAINVIEW for DB2.

MVDBC. *See* MAINVIEW for DBCTL.

MVIMS. *See* MAINVIEW for IMS.

MVIP. *See* MAINVIEW for IP.

MVLNX. *See* MAINVIEW for Linux-Servers.

MVMQ. *See* MAINVIEW for MQSeries.

MVMVS. *See* MAINVIEW for OS/390.

MVScope. MAINVIEW for OS/390 application that traces both CPU usage down to the CSECT level and I/O usage down to the channel program level.

MVSRM. *See* MAINVIEW Storage Resource Manager (SRM).

MVSRMHSM. *See* MAINVIEW SRM EasyHSM.
MVSRMSGC. *See* MAINVIEW SRM SG-Control.
MVSRMSGD. *See* MAINVIEW SRM StorageGUARD.
MVSRMSGP. *See* MAINVIEW SRM StorageGUARD.
MVVP. *See* MAINVIEW VistaPoint.
MVV TAM. *See* MAINVIEW for VTAM.
MVWEB. *See* MAINVIEW for WebSphere.

N

nested help. Multiple layers of help pop-up windows. Each successive layer is accessed by clicking a hyperlink from the previous layer.

O

object. Anything you can manipulate as a single unit. MAINVIEW objects can be any of the following: product, secondary window, view, row, column, or field.

You can issue an action against an object by issuing a line command in the line command column to the left of the object. *See* action.

OMVS workload. Workload consisting of OS/390 OpenEdition address spaces.

online help. Help information that is accessible online.

OS/390 and z/OS Installer. BMC Software common installation system for mainframe products.

OS/390 product address space (PAS). Address space containing OS/390 or z/OS data collectors, including the CMF MONITOR Extractor. Used by the MAINVIEW for OS/390, MAINVIEW for Unix System Services, and CMF MONITOR products. *See* PAS.

P

parameter library. Data set consisting of members that contain parameters for specific MAINVIEW products or a support component. There can be several versions:

- The distributed parameter library, called BBPARAM
- A site-specific parameter library or libraries

These can be

- A library created by AutoCustomization, called UBBPARAM
- A library created manually, with a unique name

PAS. Product address space. Used by the MAINVIEW products. Contains data collectors and other product functions. *See* OS/390 product address space (PAS), BBI subsystem product address space (BBI-SS PAS).

performance group workload. Collection of address spaced defined to OS/390 or z/OS. If you are running OS/390 or z/OS with WLM in compatibility mode, MAINVIEW for OS/390 creates a performance group workload instead of a service class. *See* service class workload, workload definition.

PERFORMANCE MANAGER. MAINVIEW for CICS online service for monitoring and managing current performance of CICS regions.

Performance Reporter (MVIMS Offline). MVIMS Offline component that organizes data and prints reports that can be used to analyze IMS performance.

Performance Reporter. Product component that generates offline batch reports. The following products can generate these reports:

- MAINVIEW for DB2
- MAINVIEW for CICS

Plex Manager. Product through which cross-system communication, MAINVIEW security, and an SSI context are established and controlled. Plex Manager is shipped with MAINVIEW window environment products as part of the coordinating address space (CAS) and is accessible as a menu option from the MAINVIEW Selection Menu.

PRGP workload. In MVS/SP 5.0 or earlier, or in compatibility mode in MVS/SP 5.1 or later, composite of service classes. MAINVIEW for OS/390 creates a performance group workload for each performance group defined in the current IEAIPS.xx member.

procedure library. Data set consisting of members that contain executable procedures used by MAINVIEW AutoOPERATOR. These procedures are execute command lists (EXECs) that automate site functions. There can be several versions:

- The distributed parameter library, called BBPROC
- A site-specific parameter library or libraries

These can be

- A library created by AutoCustomization, called UBBPROC
- A library created manually, with a unique name

The site-created EXECs can be either user-written or customized MAINVIEW AutoOPERATOR-supplied EXECs from BBPROC.

product address space. *See* PAS.

profile library. Data set consisting of members that contain profile information and cycle refresh definitions for a terminal session connected to a BBI-SS PAS. Other members are dynamically created by MAINVIEW applications. There can be several versions:

- The distributed profile library, called BBPROF
- A site-specific profile library or libraries

These can be

- A library created by AutoCustomization, called SBBPROF
- A library created manually, with a unique name

The site library is a common profile shared by all site users. The terminal session CLIST creates a user profile automatically if one does not exist; it is called userid.BBPROF, where userid is your logon ID. User profile libraries allow each user to specify unique PF keys, CYCLE commands, target system defaults, a Primary Option Menu, and a unique set of application profiles.

Q

query. One of two constituent parts of a view; the other is form. A query defines the data for a view; a form defines the display format. *See also* form, view.

R

realtime data. Performance data as it exists at the moment of inquiry. Realtime data is recorded during the smallest unit of time for data collection. *Contrast with* historical data. *See also* current data and interval data.

Resource Analyzer. Online realtime displays used to analyze IMS resources and determine which are affected by specific workload problems.

Resource Monitor. Online data collection services used to monitor IMS resources and issue warnings when defined utilization thresholds are exceeded.

row. (1) Horizontal component of a view or display comprising all the fields pertaining to a single device, address space, user, etc. (2) Horizontal component of a DB2 table consisting of a sequence of values, one for each column of the table.

RxD2. Product that provides access to DB2 from REXX. It provides tools to query the DB2 catalog, issue dynamic SQL, test DB2 applications, analyze EXPLAIN data, generate DDL or DB2 utility JCL, edit DB2 table spaces, perform security administration, and much more.

S

sample cycle. Time between data samples.

For the CMF MONITOR Extractor, this is the time specified in the extractor control statements (usually 1 to 5 seconds).

For realtime data, the cycle is not fixed. Data is sampled each time you press Enter.

sample library. Data set consisting of members each of which contains one of the following:

- Sample JCL that can be edited to perform specific functions
- A macro that is referenced in the assembly of user-written services
- A sample user exit routine

There can be several versions:

- The distributed sample library, called BBSAMP
- A site-specific sample library or libraries

These can be

- A library created by AutoCustomization, called UBBSAMP
- A library created manually, with a unique name

sampler. Program that monitors a specific aspect of system performance. Includes utilization thresholds used by the Exception Monitor. The CMF MONITOR Extractor contains samplers.

SBBPROF. *See* profile library.

scope. Subset of an SSI context. The scope could be all the data for the context or a subset of data within the context. It is user- or site-defined. *See* SSI context, target.

screen definition. Configuration of one or more views that have been stored with the SAVEScr command and assigned a unique name. A screen includes the layout of the windows and the view, context, system, and product active in each window.

selection view. In MAINVIEW products, view displaying a list of available views.

service class workload. Collection of address spaces defined to OS/390 or z/OS. If you are running Workload Manager (WLM) in goal mode, MAINVIEW for OS/390 creates a service class workload for each service class that you define through WLM definition dialogs.

If you are running MVS 4.3 or earlier, or MVS/SP 5.1 or later with WLM in compatibility mode, MVS creates a performance group workload instead of a service class. *See* performance group workload.

service objective. Workload performance goal, specified in terms of response time for TSO workloads or turnaround time for batch workloads. Performance group workloads can be measured by either objective. Composite workload service objectives consist of user-defined weighting factors assigned to each constituent workload. For compatibility mode, neither OS/390 nor z/OS provides any way to measure service.

service point. Specification, to MAINVIEW, of the services required to enable a specific product. Services can be actions, selectors, or views. Each target (for example, CICS, DB2, or IMS) has its own service point.

The PLEX view lists all the defined service points known to the CAS to which the terminal session is connected.

service request block (SRB). Control block that represents a routine to be dispatched. SRB mode routines generally perform work for the operating system at a high priority. An SRB is similar to a task control block (TCB) in that it identifies a unit of work to the system. *See also* task control block.

service select code. Code entered to invoke analyzers, monitors, and general services. This code is also the name of the individual service.

session. Total period of time an address space has been active. A session begins when monitoring can be performed. If the product address space (PAS) starts after the job, the session starts with the PAS.

SG-Auto. *See* MAINVIEW SRM SG-Auto.

SG-Control. *See* MAINVIEW SRM SG-Control.

single system image (SSI). Feature of the MAINVIEW window environment architecture where you can view and perform actions on multiple OS/390 systems as though they were a single system. The rows of a single tabular view can contain rows from different OS/390 or z/OS images.

Skeleton Tailoring Facility. A facility in MAINVIEW AutoOPERATOR that allows skeleton JCL to be used during job submission. Skeleton JCL can contain variables within the JCL statements to be substituted with data values at job submission time. Directive statements can be used in the skeleton JCL to cause the repetition of a set of skeleton statements. This facility functions similar to the TSO skeleton tailoring facility.

SRB. *See* service request block.

SSI. *See* single system image.

SSI context. Name created to represent one or more targets for a given product. *See* context, target.

started task workload. Address spaces running jobs that were initiated programmatically.

statistics interval. For MAINVIEW for DB2, cumulative count within a predefined interval (30-minute default set by the DB2STATS parameter in the distributed BBPARM member BBIISP00) for an analyzer service DELTA or RATE display. Specifying the DELTA parameter displays the current value as the difference between the value sampled by the current analyzer request and the value sampled at the start of the current interval. Specifying the RATE parameter displays the current value by minute (DELTA divided by the number of elapsed minutes).

stem variables. A REXX facility, supported in MAINVIEW AutoOPERATOR REXX EXECs and the Skeleton Tailoring Facility, where variable names end with a period followed by a

number, such as &POOL.1. This configuration allows each variable to actually represent a table or array of data, with the zero variable containing the number of entries in the array. For example, &POOL.0 = 5 would indicate variables &POOL.1 through &POOL.5 exist.

StopX37/II. *See* MAINVIEW SRM StopX37/II.

StorageGUARD. *See* MAINVIEW SRM StorageGUARD.

summary view. View created from a tabular view using the Summarize option in view customization. A summary view compresses several rows of data into a single row based on the summarize criteria.

SYSPROG services. Component of MAINVIEW for OS/390. Over 100 services that detect, diagnose, and correct OS/390 or z/OS system problems as they occur. Accessible from the OS/390 Performance and Control Main Menu. Note that this component is also available as a stand-alone product MAINVIEW SYSPROG Services.

system resource. *See* object.

T

target. Entity monitored by one or more MAINVIEW products, such as an OS/390 or z/OS image, an IMS or DB2 subsystem, a CICS region, or related workloads across systems. *See* context, scope, SSI context.

target context. Single target/product combination. *See* context.

TASCOSTR. MAINVIEW for IMS Offline program that summarizes detail and summary IMS Resource Utilization Files (IRUFs) to be used as input to the offline components.

task control block (TCB). Address space-specific control block that represents a unit of work that is dispatched in the address space in which it was created. *See also* service request block.

TCB. *See* task control block.

terminal session (TS). Single point of control for MAINVIEW products, allowing data manipulation and data display and providing other terminal user services for MAINVIEW products. The terminal session runs in a user address space (either a TSO address space or a standalone address space for EXCP/VTAM access).

TDIR. *See* trace log directory.

threshold. Specified value used to determine whether the data in a field meets specific criteria.

TLDS. *See* trace log data set.

total mode. Usage mode in CMFMON wherein certain columns of data reflect the cumulative value between collection intervals. Invoked by the DELTA OFF command. *See also* collection interval, delta mode.

trace. (1) Record of a series of events chronologically listed as they occur. (2) Online data collection and display services that track transaction activity through DB2, IMS, or CICS.

trace log data set (TLDS). Single or multiple external VSAM data sets containing summary or detail trace data for later viewing or printing. The trace log(s) can be defined as needed or dynamically allocated by the BBI-SS PAS. Each trace request is assigned its own trace log data set(s).

trace log directory (TDIR). VSAM linear data set containing one entry for each trace log data set. Each entry indicates the date and time of data set creation, the current status of the data set, the trace target, and other related information.

transaction. Specific set of input data that initiates a predefined process or job.

Transaction Accountant. MVIMS Offline component that produces cost accounting and user charge-back records and reports.

TS. *See* terminal session.

TSO workload. Workload that consists of address spaces running TSO sessions.

U

UAS. *See* user address space.

UBBPARM. *See* parameter library.

UBBPROC. *See* procedure library.

UBBSAMP. *See* sample library.

user address space. Runs a MAINVIEW terminal session (TS) in TSO, VTAM, or EXCP mode.

User BBPROF. *See* profile library.

V

view. Formatted data within a MAINVIEW window, acquired from a product as a result of a view command or action. A view consists of two parts: query and form. *See also* form, job activity view, query.

view definition. Meaning of data that appears online, including source of data, selection criteria for data field inclusion and placement, data format, summarization, context, product, view name, hyperlink fields, and threshold conditions.

view command. Name of a view that you type on the COMMAND line to display that view.

view command stack. Internal stack of up to 10 queries. For each command, the stack contains the filter parameters, sort order, context, product, and timeframe that accompany the view.

view help. Online help describing the purpose of a view. To display view help, place the cursor on the view name on the window information line and press PF1 (HELP).

W

window. Area of the MAINVIEW screen in which views and resources are presented. A window has visible boundaries and can be smaller than or equal in size to the MAINVIEW window area. *See* active window, alternate window, current window, MAINVIEW window area.

window information line. Top border of a window. Shows the window identifier, the name of the view displayed in the window, the system, the scope, the product reflected by the window, and the timeframe for which the data in the window is relevant. *See also* window status field.

window number. Sequential number assigned by MAINVIEW to each window when it is opened. The window number is the second character in the window status field. *See also* window status field.

window status. One-character letter in the window status field that indicates when a window is ready to receive commands, is busy processing commands, is not to be updated, or contains no data. It also indicates when an error has occurred in a window. The window status is the first character in the window status field. *See also* window information line, window status field.

window status field. Field on the window information line that shows the current status and assigned number of the window. *See also* window number, window status.

windows mode. Display of one or more MAINVIEW product views on a screen that can be divided into a maximum of 20 windows. A window information line defines the top border of each window. *Contrast with* full-screen mode.

WLM workload. In goal mode in MVS/SP 5.1 and later, a composite of service classes. MAINVIEW for OS/390 creates a workload for each WLM workload defined in the active service policy.

workflow. Measure of system activity that indicates how efficiently system resources are serving the jobs in a workload.

workload. (1) Systematic grouping of units of work (e.g., address spaces, CICS transactions, IMS transactions) according to classification criteria established by a system administrator. (2) In OS/390 or z/OS, a group of service classes within a service definition.

workload activity view. Tracks workload activity as the workload accesses system resources. A workload activity view measures workload activity in terms of resource consumption and how well the workload activity meets its service objectives.

Workload Analyzer. Online data collection and display services used to analyze IMS workloads and determine problem causes.

workload definition. Workload created through the WKLIST view. Contains a unique name, a description, an initial status, a current status, and selection criteria by which address spaces are selected for inclusion in the workload. *See* Workload Definition Facility.

Workload Definition Facility. In MAINVIEW for OS/390, WKLIST view and its associated dialogs through which workloads are defined and service objectives set.

workload delay view. Tracks workload performance as the workload accesses system resources. A workload delay view measures any delay a workload experiences as it contends for those resources.

Workload Monitor. Online data collection services used to monitor IMS workloads and issue warnings when defined thresholds are exceeded.

workload objectives. Performance goals for a workload, defined in WKLIST. Objectives can include measures of performance such as response times and batch turnaround times.

Index

Symbols

\$E command 144
\$HASP 607 145
\$HASP050 11, 12
\$HASP093 11, 12
\$HASP095 11, 12
\$HASP355 12
\$HASP636 145
\$HASP646 11, 12
\$PI command 144
\$TOTCPU 38
\$TSOCPU 38
&plan 82
@DMPCLRO 20
@TIMER 133–143

A

abend of CICS 44
Access Methods Solution 29–30, 97–??, 101–??
Access Methods Solutions
 See VSAM Failures Solution
ACTS636 variable 144
AEW messages 37
ALERT application 37
ALERT profile
 modify 67
AUTHOR field 5
Automation Reporter 111
AutoOPERATOR ALERTs 111
AutoOPERATOR Web 111

B

batch jobs
 file allocation 57–58
 file deallocation 57–58
BBISSP00 21

C

Case Sensitivity 113
CAT2291D 17
CER001C 43
CICS
 abend 44
CICS Abnormal Termination Solution 44
 EXECs 44
 invoking 44
 processing flow 44
 Rules 44
 variables 44
CICS MANAGER
 MONITOR command 61

CICS Performance Management Solutions 53–62
 See File Allocation - Deallocation Solution
 See File Degradation Analysis Solution
 See Transaction Response Time Monitor Solution
 See VSAM Control Area Split Monitor Solution
CICS Solutions 39–62
 Error Recovery Solutions 41–52
 See CICS Abnormal Termination Solution
 See CICS Temporary Storage Suspensions Solution
 See Storage Violation Solution
 See VSAM Subtask Abnormal Termination Solution
 Performance Management Solutions 53–62
 See File Allocation - Deallocation Solution
 See File Degradation Analysis Solution
 See Transaction Response Time Monitor Solution
 See VSAM Control Area Split Monitor Solution
CICS Temporary Storage Suspensions Solution 47–48
EXECs 48
 invoking 47
 processing flow 47
 Rules 48
 variables 47
CMDCON parameter 21
CODE field 5
Commands 111
Continuous State Manager 111
CONTROLM Solution 31
 EXECs 31
 invoking 31
 processing flow 31
 Rules 31
 variables 31
CPU usage 34
customization 65–67
 EXECs 65–67
 initial 65–67

D

D_ALERT_IDS variable 73
D_ALERT_Q variable 73
D_ALERT_SEND variable 73
D_ALERT_SENDOPT variable 73
D_ALERT_TGT variable 73
D_DB2n variable 73
D_DB2NUM variable 73
D_DBM1n variable 73
D_DISTn variable 73
D_IRLMn variable 73
D_MSTRn variable 73
DASD Reserve Analysis Solution 35–36
 EXECs 35
 invoking 35
 processing flow 35
 Rules 36

- variables 35
- Data Set Monitoring 25–26
- DB2 111
- DB2 Environment Set-Up 73–74
 - EXECs 74
 - invoking 74
 - processing flow 74
 - Rules 74
 - variables 73
- DB2 Global Operations Solutions
 - See* DB2ALRT Solution
 - See* DB2INIT Solution
 - See* DMRALRT Solution
 - See* DMRUTIL Solution
- DB2 Major Messages 69–72
 - EXECs 72
 - invoking 69
 - processing flow 71
 - Rules 72
 - variables 69
- DB2 Performance Management Solutions 89–93
 - See* DB2RESP Solution
 - See* DB2RNWY Solution
- DB2 Resource Contention Analysis Solutions
 - See* DB2DLOK Solution
 - See* DB2TFUL Solution
 - See* DB2THRD Solution
- DB2 Response Time 92–93
 - EXECs 93
 - invoking 92
 - processing flow 93
 - Rules 93
 - variables 92
- DB2 Runaway Query Control 89–91
 - EXECs 90
 - invoking 90
 - processing flow 90
 - Rules 91
 - variables 89
- DB2 Solutions 63–93
 - DB2 Global Operations Solutions
 - See* DB2ALRT Solution
 - See* DB2INIT Solution
 - See* DMRALRT Solution
 - See* DMRUTIL Solution
 - DB2 Performance Management Solutions
 - See* DB2RESP Solution
 - See* DB2RNWY Solution
 - DB2 Resource Contention Analysis Solutions
 - See* DB2DLOK Solution
 - See* DB2TFUL Solution
 - See* DB2THRD Solution
- DB2 Subsystems
 - Determine Number of 65
- DB2 Table Space Filling Up 84–85
 - EXECs 85
 - invoking 85
 - processing flow 85
 - Rules 85
- variables 84
- DB2 Thread Control 86–87
 - EXECs 86
 - invoking 86
 - processing flow 86
 - Rules 87
 - variables 86
- DB2ALRT 69
- DB2ALRT Solution 69–72
 - EXECs 72
 - invoking 69
 - processing flow 71
 - Rules 72
 - variables 69
- DB2DLOK 79
- DB2DLOK Solution 79–83
 - EXECs 83
 - invoking 80
 - processing flow 80
 - ALERTs 81
 - data extractor 82
 - history table 80
 - Rules 83
 - variables 79
- DB2INIT 73
- DB2INIT Solution 73–74
 - EXECs 74
 - invoking 74
 - processing flow 74
 - Rules 74
 - variables 73
- DB2RESP 92
- DB2RESP Solution 92–93
 - EXECs 93
 - invoking 92
 - processing flow 93
 - Rules 93
 - variables 92
- DB2RNWY 89
- DB2RNWY Solution 89–91
 - EXECs 90
 - invoking 90
 - processing flow 90
 - Rules 91
 - variables 89
- DB2TFUL 84
- DB2TFUL Solution 84–85
 - EXECs 85
 - invoking 85
 - processing flow 85
 - Rules 85
 - variables 84
- DB2THRD 86
- DB2THRD Solution 86–87
 - EXECs 86
 - invoking 86
 - processing flow 86
 - Rules 87
 - variables 86

Deadlock/Timeout Analysis 79–83
 EXECs 83
 invoking 80
 processing flow 80
 ALERTs 81
 data extractor 82
 history table 80
 Rules 83
 variables 79
 DESC field 5
 DFH0508 42, 43
 DFH0606 44
 DFH0901 45, 46
 DFH1311 49, 50
 DFH3437I 51
 DISP field 5
 DMR Major Messages 75–76
 EXECs 76
 invoking 75
 processing flow 76
 Rules 76
 variables 75
 DMR Utilities 77–78
 EXECs 77
 invoking 77
 Rules 78
 variables 77
 DMRALRT 75
 DMRALRT Solution 75–76
 EXECs 76
 invoking 75
 processing flow 76
 Rules 76
 variables 75
 DMRUTIL 77
 DMRUTIL Solution 77–78
 EXECs 77
 invoking 77
 Rules 78
 variables 77
 documentation boxes 3–5
 DPE002C 77, 93
 DPE005C 93
 DPE007C 93
 DPE010C 83
 DPE011C 83
 DPE012C 83
 DPE013C 83
 DPE014C 77, 83
 DPE015C 83
 DPE019C 83
 DPE020C 90
 DPE021C 90
 DPE022C 90
 DPE023C 90
 DPE024C 90
 DPE025C 90
 DPE026C 90
 DPE027C 90

DPE030C 85
 DPE031C 85
 DPE040C 86
 DPE041C 86
 DPE900C 77
 DPE900D 77
 DPE901C 77
 DPE901D 77
 DPE902C 77
 DPE902D 77
 DPE903C 77
 DPE903D 77
 DPE905C 77
 DPE905D 77
 DPE908C 77
 DPE908D 77
 DPE908DR 77
 DPE909C 77
 DPE917C 77
 DPE919C 77
 DPE920C 77
 DPE930C 77
 DPE931C 78
 DPE932C 78
 DPE933C 78
 DPE935C 78
 DSNT375I 80, 83
 DSNT376I 80, 83
 DSNT501I 80
 DSNT501X 83
 DSU001C 74
 DSU10nC 74
 Dump Data Sets Monitoring Solution 19–20
 EXECs 20
 invoking 19
 processing flow 20
 Rules 20
 variables 19
 DUMPCLER variable 19
 DUMPMF variable 19
 DUMPNDS variable 19

E

ERB102I 27
 Error Recovery Solutions 41–52
 See CICS Abnormal Termination Solution
 See CICS Temporary Storage Suspensions Solution
 See Storage Violation Solution
 See Temporary Storage Data Set Extension Failure Solution
 See Terminal Errors Solution
 See VSAM Subtask Abnormal Termination Solution
 example EXECs explained 146–150
 Exception Monitoring Solution 37–38
 EXECs 38
 invoking 37
 processing flow 38
 Rules 38

- variables 37
- EXEC Management application 3, 145
 - @TIMER EXEC 133
 - documentation box 3
 - JESDOWN 144
- EXECs 66
 - \$HASP050 12
 - \$HASP093 12
 - \$HASP095 12
 - \$HASP355 12
 - \$HASP646 12
 - \$TOTCPU 38
 - \$TSOCPU 38
 - @DMPCLRO 20
 - @TIMER 133–143
 - CER001C 43
 - DFH0508 43
 - DFH0606 44
 - DFH1311 49
 - DPE002C 77, 93
 - DPE005C 93
 - DPE007C 93
 - DPE010C 83
 - DPE011C 83
 - DPE012C 83
 - DPE013C 83
 - DPE014C 77, 83
 - DPE015C 83
 - DPE019C 83
 - DPE020C 90
 - DPE021C 90
 - DPE022C 90
 - DPE023C 90
 - DPE024C 90
 - DPE025C 90
 - DPE026C 90
 - DPE027C 90
 - DPE030C 85
 - DPE031C 85
 - DPE040C 86
 - DPE041C 86
 - DPE900C 77
 - DPE900D 77
 - DPE901C 77
 - DPE901D 77
 - DPE902C 77
 - DPE902D 77
 - DPE903C 77
 - DPE903D 77
 - DPE905C 77
 - DPE905D 77
 - DPE908C 77
 - DPE908D 77
 - DPE908DR 77
 - DPE909C 77
 - DPE917C 77
 - DPE919C 77
 - DPE920C 77
 - DPE930C 77

- DPE931C 78
- DPE932C 78
- DPE933C 78
- DPE935C 78
- DPE994C 76
- DSU001C 74
- DSU101C 65
- DSU10nC 74
- ERB102I 27
- IDC3009I 29
- IDC3351I 29
- IEA404A 21
- IEA405E 21
- IEA911E 20
- IEA994A 20
- IEA994E 20
- IEC701D 18
- IEE249I 21
- IEE361I 24
- IEE362A 24
- IEE364I 24
- IEE366I 24
- IEE949I 24
- IEF238D 18
- IEF433D 18
- IKT001D 13
- IKT010D 13
- IKT012D 13
- IOS071I 35
- JESDOWN 144–150
- LBSTART 34
- MMN001C 24
- MPE001C 35
- MPE002C 35
- MPE003C 34
- MPE004C 34
- MPE005C 34
- MPE006C 34
- national characters 6
- PWSCPU01 38
- PWSCPU02 38
- PWSCSA01 38
- PWSDEV01 38
- SCHEDCOM 31
- TLMS 17

F

- File Allocation - Deallocation Solution 57–58
 - EXECs 58
 - invoking 57
 - JCL 57
 - processing flow 58
 - Rules 58
 - variables 57
- File Degradation Analysis Solution 54–56
 - EXECs 55
 - invoking 54
 - processing flow 55

Rules 56
variables 54
FOCALPT variable 42
FT094W 61
FUNC field 5

G

GOODFOR parameter 133
GROUP field 5
Group ID 113

I

IDC3009I 29, 30
IDC3351I 29, 30
IEA404A 21, 22
IEA405E 21, 22
IEA911E 19, 20
IEA994A 19, 20
IEA994E 19, 20
IEC701D 18
IEE249I 21
IEE361I 23, 24
IEE362A 23, 24
IEE364I 23, 24
IEE366I 23, 24
IEE949I 24
IEF238D 18
IEF433D 18
IFB040I 25, 26
IFB060E 25, 26
IFB070I 25, 26
IKT001D 13, 14
IKT010D 13, 14
IKT012D 13, 14
Initial Customization Steps 65–67
initiators
 ACTIVE 34
 DRAINED 34
 HALTED 34
 inactive 34
INTERVAL parameter 133
Interval Services Application 133–134
IOS071I 35, 36
ISRBPT 134

J

Java 112
JES2 Monitoring Solution 11–12
 EXECs 12
 invoking 11
 processing flow 12
 Rules 12
 variables 11
JES2 Solution 11–12
 See JES2 Monitoring Solution
Job Scheduling Solution 31

See CONTROLM Solution

K

K Q command 21

L

LBGCLAS variable 33, 34
LBGCPUH variable 33
LBGCPUL variable 33
LBGIBEG variable 33
LBGIDEC variable 33, 34
LBGIEND variable 33
LBGIINC variable 33
LBGPAGE variable 33
LBSTART 34
list variables 144
LINES636 variable 144
Load Balancing Solution 33–34
 EXECs 34
 invoking 34
 processing flow 34
 Rules 34
 variables 33
LOGREC Data Set Monitoring Solution 25–26
 variables 25
LOGREC variable 25

M

MAX_AGE variable 79
MAX_EITHER variable 80
MAX_EXTS variable 84
MAX_EXTSn variable 84
MAX_OWNER variable 79
MAX_REQ variable 79
MAX_UTIL variable 84
MAX_UTILn variable 84
MAXACSA variable 37
MAXBCSA variable 37
MAXCCPU variable 37
MAXDEVU variable 37
MAXTCPU variable 37
MMN001C 24
monitor 66
MONITOR command 61
Monitoring Solutions 19–27
 See Dump Data Sets Monitoring Solution
 See LOGREC Data Set Monitoring Solution
 See RMF Monitoring Solution
 See SMF Data Set Monitoring Solution
 See WTO Buffers Monitoring Solution
MPE001C 35
MPE002C 35
MPE003C 34
MPE004C 34
MPE005C 34
MPE006C 34

MPP 111
MVS Performance Management Solutions
 See DASD Reserve Analysis Solution
 See Exception Monitoring Solution
 See Load Balancing Solution

MVS Solutions 7–38
 Access Methods Solution
 See VSAM Failures Solution
 JES2 Solution 11–12
 See JES2 Monitoring Solution
 Job Scheduling Solution 31
 See CONTROLM Solution
 Monitoring Solutions 19–27
 See Dump Data Sets Monitoring Solution
 See LOGREC Data Set Monitoring Solution
 See RMF Monitoring Solution
 See SMF Data Set Monitoring Solution
 See WTO Buffers Monitoring Solution
 MVS Performance Management Solutions
 See DASD Reserve Analysis Solution
 See Exception Monitoring Solution
 See Load Balancing Solution
 Performance Management Solutions 33–38
 Storage Subsystems Solutions 15
 See Shared DASD Control Solution
 See Storage Reply Solution
 See TLMS Solution
 VTAM/NCP Solution 13
 See TCAS Reply Solution
MVSCALL variable 11

N

NAMEn 84
naming convention
 exceptions to xvi
 national characters 6
NEXTTIME parameter 133, 134
NUM 84

O

ONLALMN variable 42
ONLALTN variable 42
ONLCNTN variable 42
ONLSYSN variable 42
ONLTYPN variable 42
operator-initiated EXECs xvi

P

page threshold counters 34
Performance Management Solutions
 CICS 53–62
 MVS 33–38
PJES2 variable 144
PRTS636 variable 144
PWSCPU01 38
PWSCPU02 38

PWSCSA01 38
PWSDEV01 38

R

REPLIES command 17
RESERVE command 35
RESOLVE AEW messages 37
RESOLVE commands
 REPLIES 17
 RESERVE 35
response time 53
RETAIN parameter 133
RLJP variable 144, 145
RLPJ variable 144
RMF Monitoring Solution 27
 EXECs 27
 invoking 27
 processing flow 27
 Rules 27
 variables 27
RMTS636 variable 144
RSTA service 146
Rules
 \$HASP050 12
 \$HASP093 12
 \$HASP095 12
 \$HASP355 12
 \$HASP646 12
 activate 66
 DFH0901 46
 DFH1311 50
 DSNT375I 83
 DSNT376I 83
 DSNT501X 83
 ERB102I 27
 IDC3009I 30
 IDC3351I 30
 IEA404A 22
 IEA405E 22
 IEA911E 20
 IEA994A 20
 IEA994E 20
 IEC701D 18
 IEE361I 24
 IEE362A 24
 IEE364I 24
 IEE366I 24
 IEE949I 24
 IEF238D 18
 IFB040I 26
 IFB060E 26
 IFB070I 26
 IKT001D 14
 IKT010D 14
 IKT012D 14
 IOS071I 36
 PWSCPU02 38
 PWSCSA01 38

PWSDEV01 38
 update 66

S

SCHEDCOM 31

scheduling

- CONTROLM Solution 31
 - EXECs 31
 - invoking 31
 - processing flow 31
 - variables 31
- service level objectives 61
- Shared DASD Control Solution 15
 - EXECs 16
 - invoking 15
 - processing flow 16
 - Rules 16
 - variables 15
- shared device 23, 35
- skeleton tailoring
 - defined 183
- SMF Data Set Monitoring Solution 23–24
 - EXECs 24, 26
 - invoking 23, 25
 - processing flow 23, 25
 - Rules 24, 26
 - variables 23
- SMFALT variable 23
- SMFCLEAR variable 23
- SPROC variable 133
- STIME variable 134
- STOPTIME parameter 133, 134
- Storage Reply Solution
 - EXECs 18
 - invoking 18
 - processing flow 18
 - Rules 18
 - variables 18
- Storage Subsystems Solutions
 - See* Shared DASD Control Solution
 - See* Storage Reply Solution
 - See* TLMS Solution
- Storage Violation Solution 42–43
 - EXECs 43
 - invoking 42
 - processing flow 42
 - Rules 43
 - variables 42
- Superuser authority 113
- SYS1.LOGREC data set 25
- SYS1.MANx 23
- SYSBEEP variable 11
- SYSN variable 15, 35
- SYSPROG variable 11
- system dump data sets 19

T

Tape

- See* Storage Subsystems Solutions

TARGETSS parameter 134

TCAS Reply Solution 13

- EXECs 13
- invoking 13
- processing flow 13
- Rules 14
- variables 13

Temporary Storage Data Set Extension Failure Solution 49–50

- EXECs 49
- invoking 49
- processing flow 49
- Rules 50
- variables 49

Terminal Errors Solution 51–52

- EXECs 51
- invoking 51
- processing flow 51
- Rules 52
- variables 51

text ID-driven EXECs xvi

threshold

- level
 - set 67

timer queue 133

TLMS 17

TLMS Solution 17

- EXECs 17
- invoking 17
- processing flow 17
- Rules 17
- variables 17

TMRCTR 135

TOD parameter 133, 134

TOTCPU monitor 38

Transaction Response Time Monitor Solution 61–62

- EXECs 61
- invoking 61
- JCL 61
- processing flow 61
- Rules 62
- variables 61

TSOCPU monitor 38

V

variables

- CICS Solution 161–162
- DB2 Solution 163–171
- list
- MVS Solution 159–160

VSAM Control Area Split Monitor Solution 59–60

- EXECs 59
- invoking 59
- JCL 59
- processing flow 59

- Rules 60
- variables 59
- VSAM Failures Solution 29–30
 - EXECs 29, 101
 - invoking 29
 - processing flow 29
 - Rules 30
 - variables 29
- VSAM Subtask Abnormal Termination Solution 45–46
 - EXECs 46
 - invoking 45
 - processing flow 45
 - Rules 46
 - variables 45
- VTAM/NCP Solutions
 - See* TCAS Reply Solution

W

- WTO buffer shortages 21
- WTO Buffers Monitoring Solution 21–22
 - EXECs 21
 - invoking 21
 - processing flow 21
 - Rules 22
 - variables 21
- WTORS
 - IKT001D 13
 - IKT010D 13
 - TCAS 13

X

- XSTOP variable 134

STOP!

IMPORTANT INFORMATION - DO NOT INSTALL THIS PRODUCT UNLESS YOU HAVE READ ALL OF THE FOLLOWING MATERIAL

By clicking the YES or ACCEPT button below (when applicable), or by installing and using this Product or by having it installed and used on your behalf, You are taking affirmative action to signify that You are entering into a legal agreement and are agreeing to be bound by its terms, EVEN WITHOUT YOUR SIGNATURE. BMC is willing to license this Product to You ONLY if You are willing to accept all of these terms. CAREFULLY READ THIS AGREEMENT. If You DO NOT AGREE with its terms, DO NOT install or use this Product; press the NO or REJECT button below (when applicable) or promptly contact BMC or your BMC reseller and your money will be refunded if by such time You have already purchased a full-use License.

SOFTWARE LICENSE AGREEMENT FOR BMC PRODUCTS

SCOPE. This is a legally binding Software License Agreement ("**License**") between You (either an individual or an entity) and BMC pertaining to the original computer files (including all computer programs and data stored in such files) contained in the enclosed Media (as defined below) or made accessible to You for electronic delivery, if as a prerequisite to such accessibility You are required to indicate your acceptance of the terms of this License, and all whole or partial copies thereof, including modified copies and portions merged into other programs (collectively, the "**Software**"). "**Documentation**" means the related hard-copy or electronically reproducible technical documents furnished in association with the Software, "**Media**" means the original BMC-supplied physical materials (if any) containing the Software and/or Documentation, "**Product**" means collectively the Media, Software, and Documentation, and all Product updates subsequently provided to You, and "**You**" means the owner or lessee of the hardware on which the Software is installed and/or used. "**BMC**" means BMC Software Distribution, Inc. unless You are located in one of the following regions, in which case "BMC" refers to the following indicated BMC Software, Inc. subsidiary: (i) Europe, Middle East or Africa --BMC Software Distribution, B.V., (ii) Asia/Pacific -- BMC Software Asia Pacific Pte Ltd., (iii) Brazil -- BMC Software do Brazil, or (iv) Japan -- BMC Software K.K. If You enter into a separate, written software license agreement signed by both You and BMC or your authorized BMC reseller granting to you the rights to install and use this Product, then the terms of that separate, signed agreement will apply and this License is void.

FULL-USE LICENSE. Subject to these terms and payment of the applicable license fees, BMC grants You this non-exclusive License to install and use one copy of the Software for your internal use on the number(s) and type(s) of servers or workstations for which You have paid or agreed to pay to BMC or your BMC reseller the appropriate license fee. If your license fee entitles You only to a License having a limited term, then the duration of this License is limited to that term; otherwise this License is perpetual, subject to the termination provisions below.

TRIAL LICENSE. If You have not paid or agreed to pay to BMC or your BMC Reseller the appropriate license fees for a full use license, then, **NOTWITHSTANDING ANYTHING TO THE CONTRARY CONTAINED IN THIS LICENSE:** (i) this License consists of a non-exclusive evaluation license ("Trial License") to use the Product for a limited time ("Trial Period") only for evaluation; (ii) during the Trial Period, You may not use the Software for development, commercial, production, database management or other purposes than those expressly permitted in clause (i) immediately above; and (iii) your use of the Product is on an **AS IS** basis, and **BMC, ITS RESELLERS AND LICENSORS GRANT NO WARRANTIES OR CONDITIONS (INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) TO YOU AND ACCEPT NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE.** If You use this Product for other than evaluation purposes or wish to continue using it after the Trial Period, you must purchase a full-use license. When the Trial Period ends, your right to use this Product automatically expires, though in certain cases You may be able to extend the term of the Trial Period by request. Contact BMC or your BMC reseller for details.

TERM AND TERMINATION. This License takes effect on the first to occur of the date of shipment or accessibility to You for electronic delivery, as applicable (the "**Product Effective Date**"). You may terminate this License at any time for any reason by written notice to BMC or your BMC reseller. This License and your right to use the Product will terminate automatically with or without notice by BMC if You fail to comply with any material term of this License. Upon termination, You must erase or destroy all components of the Product including all copies of the Software, and stop using or accessing the Software. Provisions concerning Title and Copyright, Restrictions (or Restricted Rights, if You are a U.S. Government entity) or limiting BMC's liability or responsibility shall survive any such termination.

TITLE AND COPYRIGHT; RESTRICTIONS. All title and copyrights in and to the Product, including but not limited to all modifications thereto, are owned by BMC and/or its affiliates and licensors, and are protected by both United States copyright law and applicable international copyright treaties. You will not claim or assert title to or ownership of the Product. To the extent expressly permitted by applicable law or treaty notwithstanding this limitation, You may copy the Software only for backup or archival purposes, or as an essential step in utilizing the Software, but for no other purpose. You will not remove or alter any copyright or proprietary notice from copies of the Product. You acknowledge that the Product contains valuable trade secrets of BMC and/or its affiliates and licensors. Except in accordance with the terms of this License, You agree (a) not to decompile, disassemble, reverse engineer or otherwise attempt to derive the Software's source code from object code except to the extent expressly permitted by applicable law or treaty despite this limitation; (b) not to sell, rent, lease, license, sublicense, display, modify, time share, outsource or otherwise transfer the Product to, or permit the use of this Product by, any third party; and (c) to use reasonable care and protection to prevent the unauthorized use, copying, publication or dissemination of the Product and BMC confidential information learned from your use of the Product. **You will not export or re-export any Product without both the written consent of BMC and the appropriate U.S. and/or foreign government license(s) or license exception(s).** Any programs, utilities, modules or other software or documentation created, developed, modified or enhanced by or for You using this Product shall likewise be subject to these restrictions. BMC has the right to obtain injunctive relief against any actual or threatened violation of these restrictions, in addition to any other available remedies. Additional restrictions may apply to certain files, programs or data supplied by third parties and embedded in the Product; consult the Product installation instructions or Release Notes for details.

LIMITED WARRANTY AND CONDITION. If You have purchased a Full-Use License, BMC warrants that (i) the Media will be, under normal use, free from physical defects, and (ii) for a period of ninety (90) days from the Product Effective Date, the Product will perform in substantial accordance with the operating specifications contained in the Documentation that is most current at the Product Effective Date. BMC's entire liability and your exclusive remedy under this provision will be for BMC to use reasonable best efforts to remedy defects covered by this warranty

and condition within a reasonable period of time or, at BMC's option, either to replace the defective Product or to refund the amount paid by You to license the use of the Product. BMC and its suppliers do not warrant that the Product will satisfy your requirements, that the operation of the Product will be uninterrupted or error free, or that all software defects can be corrected. This warranty and condition shall not apply if: (i) the Product is not used in accordance with BMC's instructions, (ii) a Product defect has been caused by any of your or a third party's malfunctioning equipment, (iii) any other cause within your control causes the Product to malfunction, or (iv) You have made modifications to the Product not expressly authorized in writing by BMC. No employee, agent or representative of BMC has authority to bind BMC to any oral representations, warranties or conditions concerning the Product. **THIS WARRANTY AND CONDITION IS IN LIEU OF ALL OTHER WARRANTIES AND CONDITIONS. THERE ARE NO OTHER EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS LICENSE OR ANY PRODUCT LICENSED HEREUNDER. THIS PARAGRAPH SHALL NOT APPLY TO A TRIAL LICENSE.** Additional support and maintenance may be available for an additional charge; contact BMC or your BMC reseller for details.

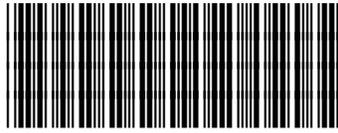
LIMITATION OF LIABILITY. Except as stated in the next succeeding paragraph, BMC's and your BMC reseller's total liability for all damages in connection with this License is limited to the price paid for the License. **IN NO EVENT SHALL BMC BE LIABLE FOR ANY CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE OR INDIRECT DAMAGES OF ANY KIND ARISING OUT OF THE USE OF THIS PRODUCT (SUCH AS LOSS OF PROFITS, GOODWILL, BUSINESS, DATA OR COMPUTER TIME, OR THE COSTS OF RECREATING LOST DATA), EVEN IF BMC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Some jurisdictions do not permit the limitation of consequential damages so the above limitation may not apply.

INDEMNIFICATION FOR INFRINGEMENT. BMC will defend or settle, at its own expense, any claim against You by a third party asserting that your use of the Product within the scope of this License violates such third party's patent, copyright, trademark, trade secret or other proprietary rights, and will indemnify You against any damages finally awarded against You arising out of such claim. However, You must promptly notify BMC in writing after first receiving notice of any such claim, and BMC will have sole control of the defense of any action and all negotiations for its settlement or compromise, with your reasonable assistance. BMC will not be liable for any costs or expenditures incurred by You without BMC's prior written consent. If an order is obtained against your use of the Product by reason of any claimed infringement, or if in BMC's opinion the Product is likely to become the subject of such a claim, BMC will at its option and expense either (i) procure for You the right to continue using the product, or (ii) modify or replace the Product with a compatible, functionally equivalent, non-infringing Product, or (iii) if neither (i) nor (ii) is practicable, issue to You a pro-rata refund of your paid license fee(s) proportionate to the number of months remaining in the 36 month period following the Product Effective Date. This paragraph sets forth your only remedies and the total liability to You of BMC, its resellers and licensors arising out of such claims.

GENERAL. This License is the entire understanding between You and BMC concerning this License and may be modified only in a mutually signed writing between You and BMC. If any part of it is invalid or unenforceable, that part will be construed, limited, modified, or severed so as to eliminate its invalidity or unenforceability. This License will be governed by and interpreted under the laws of the jurisdiction named below, without regard to conflicts of law principles, depending on which BMC Software, Inc. subsidiary is the party to this License: (i) BMC Software Distribution, Inc. - the State of Texas, U.S.A., (ii) BMC Software Distribution, B.V. - The Netherlands, (iii) BMC Software Asia Pacific Pte Ltd. -- Singapore (iv) BMC Software do Brazil -- Brazil, or (v) BMC Software K.K. -- Japan. Any person who accepts or signs changes to the terms of this License promises that they have read and understood these terms, that they have the authority to accept on your behalf and legally obligate You to this License. Under local law and treaties, the restrictions and limitations of this License may not apply to You; You may have other rights and remedies, and be subject to other restrictions and limitations.

U.S. GOVERNMENT RESTRICTED RIGHTS. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in FAR Section 52.227-14 Alt. III (g)(3), FAR Section 52.227-19, DFARS 252.227-7014 (b) or DFARS 227.7202, as amended from time to time. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

Notes



100042325