MAINVIEW® AutoOPERATOR™

Reference Summary

Version 6.3

March 31, 2003

bmcsoftware

Copyright 2003 BMC Software, Inc., as an unpublished work.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. 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 UNPUBLISHED–RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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. Contractor/Manufacturer is BMC Software, Inc., 2101 CityWest Blvd., Houston, TX 77042-2827, USA. Any contract notices should be sent to this address.

Contacting BMC Software

You can access the BMC Software Web site at **http://www.bmc.com**. From this Web site, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States 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 United States 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.”

Support Web Site

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at **http://www.bmc.com/support.html**. From this Web 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 by Telephone or E-mail

In the United States and Canada, if you need technical support and do not have access to the Web, call 800 537 1813. Outside the United States 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 Customer Support 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 other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level

- 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

BMC Software, Inc., Confidential and Proprietary Information

Contents

About This Book ix

Chapter 1 **Default PF Keys and Transfer Commands Summary**
Chart of Default PF Keys 1-1
BBI Command Prefixes 1-2
BBI Control Commands 1-3
Application and Product Transfer Commands 1-9
BBI Transfer Commands 1-9
Application Transfer Commands 1-10

Chapter 2 **CLIST and REXX EXECs Summary**
Passing Information to CLIST EXECs: PROC Statement 2-1
Passing Information to REXX EXECs 2-2
EXEC Parameters 2-2
Documenting EXECs and Rules 2-4

Chapter 3 **TSO Variables for CLIST and REXX EXECs**
MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs 3-1
MAINVIEW AutoOPERATOR-Supplied SHARED Variables for CLIST and REXX EXECs 3-8
ALERT Variables for REXX and CLIST EXECs and MAINVIEW AutoOPERATOR Rules 3-9
OSPI Control Variables in LOCAL Variable Pool—REXX and CLIST EXECs 3-11
RxD2/LINK Control Variables for REXX EXECs Only 3-12
Functions and Commands for REXX EXECs 3-13
Restricted TSO/E Commands and Functions 3-13
RxD2/LINK REXX Function Extensions 3-14

BMC Software, Inc., Confidential and Proprietary Information

RxD2/LINK REXX Common Function EXECs 3-15
Command Restrictions in EXECs 3-15

Chapter 4 **IMFEXEC Command Syntax Summary**
Commands and Examples 4-1

Chapter 5 **ACCESS NV Command Syntax Summary**
NAIEXEC Completion Codes 5-2

Chapter 6 **Rule Statement Usage**
Selection Criteria Fields 6-1
Action Specification Fields 6-4
Basic SYSPROG Services Command Summary 6-5

Chapter 7 **MQSeries Variables**
General Purpose Variables for MQI API Only 7-1
Options and Miscellaneous Variables for MQI API Only 7-2
Message Descriptor Variables 7-3
Object Descriptor Variables 7-5
Common MQseries Variables 7-5
Event Variables 7-6
Dead Letter Header Variables 7-7
Trigger Message Variables. 7-8
CICS Information Header Variables 7-9
IMS Information Header Variables 7-11
Transmission Queue Header Variables 7-12
PCF Variables 7-14
Message Descriptor Extension Variables 7-14
Work Information Header Variables 7-15
Reference Message Header Variables 7-16
Trigger Message 2 (Character format) Variables 7-16
Rules and Formatting Header Variables 7-17
Rules and Formatting Header Variables Version 2 7-18

Chapter 8 **MQI Command Syntax Summary**
MQI Commands and Examples 8-1
Completion Codes for MQI Commands 8-6

Chapter 9 **Continuous State Management (CSM)**
CSMACT Commands 9-1

BMC Software, Inc., Confidential and Proprietary Information

CSM Command Line Interface 9-3

Chapter 10 **AOAnywhere Command Summary**
Commands and Examples 10-1
AOAnywhere Return Codes 10-5

Chapter 11 **IMFEXEC ARRAY Commands**
IMFEXEC ARRAY Commands and Examples 11-1

Chapter 12 **MAINVIEW API Command Summary**
MAINVIEW API Commands and Examples 12-1
MAINVIEW API Return Codes 12-3

BMC Software, Inc., Confidential and Proprietary Information

BMC Software, Inc., Confidential and Proprietary Information

About This Book

This book should be used as a quick reference only; complete user information about MAINVIEW AutoOPERATOR can be found in the following manuals:

- MAINVIEW AutoOPERATOR Customization Guide
- *MAINVIEW AutoOPERATOR Basic Automation Guide*
- *MAINVIEW AutoOPERATOR Advanced Automation Guide*
- MAINVIEW AutoOPERATOR *Options User Guide*
- MAINVIEW AutoOPERATOR *for MQSeries Installation and User Guide*

Note

This book assumes that you are familiar with your host operating system. Additionally, you should be familiar with MAINVIEW AutoOPERATOR advanced automation techniques.

BMC Software, Inc., Confidential and Proprietary Information

Related Documentation

BMC Software products offer several types of documentation:

- online and printed books
- release notes

In addition to the above list of books and the online Help, you can find useful information in the publications listed in the following table. As "Online and Printed Books" on page -xi explains, these publications are available on request from BMC Software.

Category	Document	Description
Installation Documents	*OS/390 and z/OS Installer Guide*, *MAINVIEW Installation Requirements Guide* *Using MAINVIEW* *MAINVIEW Administration Guide*	contains instructions for setting up your installation environment and installing your MAINVIEW products
Core Documents	MAINVIEW Common Customization Guide	contains instructions for setting up the operational environment of all MAINVIEW products to your site's requirements

The following IBM manuals should also be referenced if necessary:

- *TSO/E Administration*
- *TSO/E CLISTs*
- *TSO/E Primer*
- *TSO/E REXX User's Guide*
- *TSO/E REXX/MVS Reference*

BMC Software, Inc., Confidential and Proprietary Information

Online and Printed Books

The books that accompany BMC Software products are available in online format and printed format. You can view online books with Acrobat Reader from Adobe Systems. The reader is provided at no cost, as explained in “To Access Online Books.” You can also obtain additional printed books from BMC Software, as explained in “To Request Additional Printed Books.”

To Access Online Books

Online books are formatted as Portable Document Format (PDF) files. You can view them, print them, or copy them to your computer by using Acrobat Reader 3.0 or later. You can access online books from the documentation compact disc (CD) that accompanies your product or from the World Wide Web.

In some cases, installation of Acrobat Reader and downloading the online books is an optional part of the product-installation process. For information about downloading the free reader from the Web, go to the Adobe Systems site at **http://www.adobe.com**.

To view any online book that BMC Software offers, visit the support page of the BMC Software Web site at **http://www.bmc.com/support.html**. Log on and select a product to access the related documentation. (To log on, first-time users can request a user name and password by registering at the support page or by contacting a BMC Software sales representative.)

To Request Additional Printed Books

BMC Software provides a core set of printed books with your product order. To request additional books, go to **http://www.bmc.com/support.html**.

BMC Software, Inc., Confidential and Proprietary Information

Release Notes

Printed release notes accompany each BMC Software product. Release notes provide up-to-date information such as

- updates to the installation instructions
- last-minute product information

The latest versions of the release notes are also available on the Web at **http://www.bmc.com/support.html**.

Conventions

The following conventions are used in this book:

- This book includes special elements called *notes*, *warnings*, *examples*, and *tips*:

Note

Notes provide additional information about the current subject.

Warning

Warnings alert you to situations that can cause problems, such as loss of data, if you do not follow instructions carefully.

Example

An example clarifies a concept discussed in text.

Tip

A tip provides useful information that may improve product performance or make procedures easier to follow.

- `All syntax, operating system terms, and literal examples are presented in this typeface.`

- In instructions, **boldface** type highlights information that you enter. File names, directories, and Web addresses also appear in boldface type.

- The symbol » denotes one-step instructions.

- In syntax, path names, or system messages, *italic* text represents a variable, as shown in the following examples:

  `The table` *`table_name`* `is not available.`

  system/*instance/file_name*

- In syntax, the following additional conventions apply:

  — A vertical bar (|) separating items indicates that you must choose one item. In the following example, you would choose *a*, *b*, or *c*:

  `a | b | c`

  — An ellipsis (. . .) indicates that you can repeat the preceding item or items as many times as necessary.

  — Square brackets ([ ]) around an item indicate that the item is optional.

BMC Software, Inc., Confidential and Proprietary Information

BMC Software, Inc., Confidential and Proprietary Information

Default PF Keys and Transfer Commands Summary

This chapter describes default PF keys and various BBI commands.

Chart of Default PF Keys

Table 1-1 Default assignments for PF Keys

Key	Description	Key	Description
PF1/13	HELP	PF7/19	UP
PF2/14	SPLIT	PF8/20	DOWN
PF3/15	END	PF9/21	SWAP
PF4/16	TRANSFER/PRINT	PF10/22	LEFT
PF5/17	EXPAND/LOG DISPLAY	PF11/23	RIGHT
PF6/18	GO*	PF12/24	RETRIEVE
*Terminate with the ATTN (SNA) or PA1 (non-SNA) key			

BMC Software, Inc., Confidential and Proprietary Information

BBI Command Prefixes

This section contains a matrix showing

- subsystem where a command is processed
- MAINVIEW AutoOPERATOR prefix for that particular subsystem
- examples of commands
- related security keyword in the UBBPARM user profile entry
- products required for this command to be executed
- additional comments where applicable

Issue these commands from the BBI Log display, master console, or from an EXEC (see the IMFEXEC CMD command).

Command/ Subsys	Pre-fix	Example	UBBPARM/ USERID	Change Prefix (BBISSP00)	Product Re-quired	Comments
DB2	- #-	-DIS THREAD(*) #-DIS THREAD(*)	DB2CMD MVSCMD		DMR AAO	
MVS	#	#D A,L	MVSCMD		AAO	
IMS	/	/DIS A	IMSCMD		IAO	
CICS	#	#F CICS,CEMT	MVSCMD		AAO	
BBI	.	.D A	BBICMD	CMDCHAR	BBI	MVS console only
JES2	#$	#$DI	MVSCMD	JESCHAR	AAO	
JES3	#*	#*I,J=PROD1234	MVSCMD	JES3CHAR	AAO	
JRNL	*	*THIS IS A TEST	JRNLMSG		BBI	
IMS Tran	!	!PART A12345	IMSTRAN		IAO	
SYSPROG service	?	?STATUS	RESAUTH		MAO	SYSPROG services
IMS Msg	$	$'This is a test' LTERM(AB123)	IMSMSG		IAO	
EXEC	% | ¢	%STARTUP STARTUP	EXEC EXEC		AAO AAO	

BMC Software, Inc., Confidential and Proprietary Information

BBI Control Commands

BBI control commands are used to control resources and functions for these products:

- MAINVIEW AutoOPERATOR
- MAINVIEW for CICS
- MAINVIEW for DB2
- MAINVIEW for DBCTL
- MAINVIEW for IMS

To use the BBI control commands:

For command syntax:

- Use a blank between commands and keywords.
- Use an equal sign, blank, or comma between keywords and parameters.

For command notation shown in the table that follows:

- Command, keyword and parameters are shown as an alternate choice as indicated by a vertical line between them.
- The symbolic word, ssid, represents a one- to four-character BBI subsystem ID.
- The symbolic word, userid, represents a one- to seven-character TSO user logon ID.
- The symbolic word, DESTID, represents a one- to eight-character VTAM destination ID

Table 1-2 BBI Control Commands (Part 1 of 6)

Command	Keywords	Parameters	Notes
.CANCEL | C	EXEC | E	eid | ALL	**Warning:** Use extreme caution when using the ALL keyword because all EXECs running or queued at that point in time will be canceled.

BMC Software, Inc., Confidential and Proprietary Information

Table 1-2 BBI Control Commands (Part 2 of 6)

Command	Keywords	Parameters	Notes
.DISPLAY | D	ACTIVE | A	N/A	Logs the status of the BBI-SS PAS.
	EXEC | E	[ALL | HIGH | NORMAL | STATS]	Applies to MAINVIEW AutoOPERATOR only.
	GME | G	[ALL | CONN | node | RECEIVE | PUBLISH | TRACE]	
	JOURNAL | J	N/A	Logs the status of the BBI-SS PAS Journal log.
	LINK | L	subsys | ALL	
	XCF	N/A	
	KEYS | K	N/A	Displays current KEYS used. Also displays information about whether the keys are valid, expired, or invalid along with the CPU ID.
	NETCALRT | NETC	[ALL | destid]	Applies to MAINVIEW AutoOPERATOR to PATROL Enterprise Manager PATROL ENTERPRISE MANAGER ONLY.
	PARTNERS	N/A	Applies to MAINVIEW AutoOPERATOR only. Shows the status of all TapeSHARE partners in the TapeSHARE PLEX.
	PRODUCTS | P	N/A	
	REMOTE	N/A	Display users that are connected remotely.
	RULES	N/A	
	SOF	N/A	Applies to MAINVIEW AutoOPERATOR only.
	SSR	N/A	Displays CPU usage for each subsystem exit used by the BBI-SS PAS.
	USER | U	[ALL | userid]	Logs status of all TS users assigned to this BBI-SS PAS.
	VPOOL | V	[ALL | name] [SHARED | PROFILE]	Applies to MAINVIEW AutoOPERATOR only. Displays the SHARED (default) or PROFILE variable pool statistics for a particular variable (name) or for all variables (ALL).

BMC Software, Inc., Confidential and Proprietary Information

Table 1-2 BBI Control Commands (Part 3 of 6)

Command	Keywords	Parameters	Notes
.GET|G	dddd	ONLINE GIVE|NOGIVE TAKE|NOTAKE	
.HELP|H	[ALL|bbicmd]	N/A	A specific command would be requested as: .HELP | .H command
.LOCATE|L	U,*dddd*	no parameter	Applies to MAINVIEW AutoOPERATOR only. Locates and displays the status of a device, where *dddd* is the device address. Requires MAINVIEW AutoOPERATOR TapeSHARE.
.RESET|E	AUTH|A	[ALL|userid]	If ALL is specified, all authorization is re-created for local and remote users as well as the generic authorization defined by $USERID and $RMTID.
	BLDL|B	SYSPROC	The RESET command is not needed if members are simply updated.
	DUMP|D	N/A	Dump data sets are cleared only when specified in the SSJCL.
	EXEC|E	N/A	
	GME|G	gme-node	
	MQLIST|MQ	[SUFFIX ID]	Resets AAOMQL*xx*.
	NETCALRT| NETC	[ALL | destid] [ON | OFF | RELOAD]	Applies to MAINVIEW AutoOPERATOR to PATROL Enterprise Manager ONLY.
	PARM|P	[AAOALS*xx*]	Applies to MAINVIEW AutoOPERATOR only. Resets parameters pertaining to MAINVIEW AutoOPERATOR ALERTs.

BMC Software, Inc., Confidential and Proprietary Information

Table 1-2 BBI Control Commands (Part 4 of 6)

Command	Keywords	Parameters	Notes
RESET | E (cont.)	PARM | P (cont.)	AAOALT*xx*	MAINVIEW AutoOPERATOR to PATROL Enterprise Manager ONLY. After processing of AAOALT*xx*, where *xx* can be any suffix, the VTAM connection and all remote connections are established.
		AAOARP*xx*	Applies to MAINVIEW AutoOPERATOR only.
		AAOEXP*xx*	Applies to MAINVIEW AutoOPERATOR only.
		AAORULxx	Used to reset a Rule Set. Causes MAINVIEW AutoOPERATOR to read in a new version of the entire Rule Set from DASD.
		AAOTRN*xx*	Applies to MAINVIEW AutoOPERATOR for IMS only.
		AAOTSP*xx*	
		AAOGME*xx*	
		CMRSECU	Applies to MAINVIEW for CICS and MAINVIEW AutoOPERATOR for CICS option only.
		RULxxxx	Used to reset a Rule Set.
	QManager	mq-ssid	
	RULES	N/A	Applies to MAINVIEW AutoOPERATOR only. Reinitializes all currently active Rule Sets.
	STATS | S	N/A	Applies to MAINVIEW AutoOPERATOR only. Starts recording automation statistics.

BMC Software, Inc., Confidential and Proprietary Information

Table 1-2 BBI Control Commands (Part 5 of 6)

Command	Keywords	Parameters	Notes
.SET | T	DUMPS	[YES | NO | SDUMP]	Creates BBI formatted dumps (BBI-SS PAS only) if YES is specified. If SDUMP is specified, SVC dumps are created for the BBI-SS PAS only.
	RULE	[EN | DI,id]	Applies to MAINVIEW AutoOPERATOR only. To enable or disable a Rule, use the format: .T RULE,ENA | DIS,*xxxxxxxx* where *xxxxxxxx* is the Rule ID.
	RULESET	[ENA | DIS | RES | SAV] rule-set-name	Applies to MAINVIEW AutoOPERATOR only. Use the format: .T RULESET,ENA, *xxxxxxxx* where *xxxxxxxx* is the Rule Set name.
	JESFLTR	CMD	
	DAE	ON | OFF	
.START | S	COMP	NETCALRT | NETC	Applies to MAINVIEW AutoOPERATOR to PATROL Enterprise Manager ONLY.
	EXEC | E	pattern-name	Applies to MAINVIEW AutoOPERATOR only.
	GME | G	gme-node	
	GTS	[BBTTCP*xx* suffix] [Default]	
	JOURNAL | J	N/A	Startup logic is the same as when a BBI-SS PAS is initially started.
	LINK | L	[ssid | ALL]	
	LOGON | LO	N/A	Allows users to log on to the BBI-SS PAS through VTAM.
	RULES	N/A	Applies to MAINVIEW AutoOPERATOR only.

BMC Software, Inc., Confidential and Proprietary Information

Table 1-2 BBI Control Commands (Part 6 of 6)

Command	Keywords	Parameters	Notes
.STOP | P	EXEC | E	pattern-name	Applies to MAINVIEW AutoOPERATOR only.
	COMP	NETCALRT | NETC	Applies to MAINVIEW AutoOPERATOR to PATROL Enterprise Manager ONLY.
	IMAGE | I	N/A	Applies to MAINVIEW for CICS, MAINVIEW for DB2, MAINVIEW for DBCTL, and MAINVIEW for IMF only.
	GME | G	gme-node	
	GTS	N/A	
	JOURNAL | J	N/A	Stops journal logging for the current BBI-SS PAS.
	LINK | L	[ssid | ALL]	
	LOGON | LO	N/A	Prevents users from starting any NEW BBI-TS sessions through VTAM.
	.RULES	N/A	Applies to MAINVIEW AutoOPERATOR only.
.SWITCH | I	JOURNAL | J	N/A	

BMC Software, Inc., Confidential and Proprietary Information

Application and Product Transfer Commands

Use these product line names on the COMMAND line with the TRANSFER | TRAN command to transfer to another installed BMC Software product; for example: `TRAN SYSA MAO`.

Command	Product	Command	Product
AO	MAINVIEW AutoOPERATOR Primary Option Menu	CAO	CICS Operator Workstation (MAINVIEW AutoOPERATOR)
IAO	IMS Operator Workstation (MAINVIEW AutoOPERATOR)	MAO	MVS Operator Workstation (MAINVIEW AutoOPERATOR)
CICS	MAINVIEW for CICS	DB2	MAINVIEW for DB2
IMS	MAINVIEW for IMS		

BBI Transfer Commands

Use the following commands within any BBI-2 product to transfer to a BBI application. These applications are common throughout all BBI-2 products.

Command	Application	Command	Application
CODES | COD	Messages and Codes	LOG	LOG Display (same as JOURNAL)
CYCLE | CYC	Service Refresh Cycle	MSG	Messages and Codes List
FOCAL | FOC	FOCAL POINT Overview Display	REFRESH | REF	Service Refresh Cycle (same as CYCLE)
JOURNAL | JOU	LOG Display	TI	Time-Initiated EXECs Requests
KEYS | KEY	Program Function Keys		

BMC Software, Inc., Confidential and Proprietary Information

Application Transfer Commands

Using the information in Table 1-3, enter the Product Line Transfer command listed in the first column, a semicolon, and an Application Transfer Command from the second column on any COMMAND line. You will get the panel listed in the third column if the product listed in the fourth column is installed.

Table 1-3 Application Transfer Commands (Part 1 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
MAINVIEW AutoOPERATOR Base			
AO	ALERTS | ALE	ALERT Overview	MAINVIEW AutoOPERATOR
AO	DPM	Dynamic Parameter Manager	MAINVIEW AutoOPERATOR
AO	EMA	EXEC Management	MAINVIEW AutoOPERATOR
AO	EVENTS | EVE	Event Activity Statistics	MAINVIEW AutoOPERATOR
AO	EXEC	EXEC Management	MAINVIEW AutoOPERATOR
AO	MAS	Event Activity Statistics	MAINVIEW AutoOPERATOR
AO	MSGSTATS | MSGS	Event Activity Statistics	MAINVIEW AutoOPERATOR
AO	OSPI	OSPI Script Development	MAINVIEW AutoOPERATOR
AO	NV	NetView OPERATOR WORKSTATION	MAINVIEW AutoOPERATOR
AO	RULES | RUL	Automation Control	MAINVIEW AutoOPERATOR
AO	SOF	Shared Object Facility	MAINVIEW AutoOPERATOR

BMC Software, Inc., Confidential and Proprietary Information

Table 1-3 Application Transfer Commands (Part 2 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
AO	TI	Time-Initiated EXECs	MAINVIEW AutoOPERATOR
AO	XALRTS | XAL	ALERT Detail	MAINVIEW AutoOPERATOR
CICS Operator Workstation or MAINVIEW for CICS			
CAO | CICS	ALERTS | ALE	ALERT Overview	MAINVIEW AutoOPERATOR
CAO | CICS	AT	Active Timer Requests	MAINVIEW for CICS
CAO | CICS	BROADCAST | BROA	CICS Broadcast	MAINVIEW AutoOPERATOR
CAO | CICS	CMRTOOLS	MAINVIEW for CICS Tools Menu	MAINVIEW for CICS
CAO | CICS	CT	Current Traces	MAINVIEW for CICS
CAO | CICS	DM	Display Monitors	MAINVIEW for CICS
CAO | CICS	DW	Display Warnings	MAINVIEW for CICS
CAO | CICS	EXEC | VIEW | EX svc parm1, parm2	Execute a MAINVIEW for CICS service with defaults or passed parameters	MAINVIEW for CICS
CAO | CICS	HISTORY | HIST	MAINVIEW for CICS History Selection	MAINVIEW for CICS
CAO | CICS	HT	History Traces	MAINVIEW for CICS
CAO | CICS	PUT	MAINVIEW for CICS PUT Level	MAINVIEW for CICS
CAO | CICS	SD	Statistics and Defaults	MAINVIEW for CICS
CAO | CICS	SM	Start Monitor	MAINVIEW for CICS
CAO | CICS	ST	Start Trace	MAINVIEW for CICS
CAO | CICS	STATUS | STA	CICS System Status	MAINVIEW AutoOPERATOR
CAO | CICS	UGRAPH | UGR	User Defined Graph Selection	MAINVIEW for CICS

BMC Software, Inc., Confidential and Proprietary Information

Table 1-3 Application Transfer Commands (Part 3 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
CAO | CICS	XALRTS | XAL	Alert Detail	MAINVIEW AutoOPERATOR
MAINVIEW for DB2			
DB2	AN	Analyzer Display Services	MAINVIEW for DB2
DB2	AT	Active Timer Requests	MAINVIEW for DB2
DB2	CT	Current Traces	MAINVIEW for DB2
DB2	DM	Display Monitors	MAINVIEW for DB2
DB2	DW	Display Warnings	MAINVIEW for DB2
DB2	GC	General Commands	MAINVIEW for DB2
DB2	GT	Graph Thread History	MAINVIEW for DB2
DB2	EXEC | EX svc parm1, parm2	Execute a MAINVIEW for DB2 service with defaults or passed parameters	MAINVIEW for DB2
DB2	HT	HISTORY Traces	MAINVIEW for DB2
DB2	IO	I/O Analysis Options	MAINVIEW for DB2 (Release 3.1 and later)
DB2	CTIO	Current I/O Traces	MAINVIEW for DB2 (Release 3.1 and later)
DB2	HTIO	History I/O Traces	MAINVIEW for DB2 (Release 3.1 and later)
DB2	MN	Data Collection Monitors	MAINVIEW for DB2
DB2	PM	DB2 System Status	MAINVIEW for DB2
DB2	SD	Statistics and Defaults	MAINVIEW for DB2
DB2	SM	Start Monitors	MAINVIEW for DB2
DB2	ST	Start Application Trace	MAINVIEW for DB2
DB2	VT	Current Traces	MAINVIEW for DB2
IMS Operator Workstation or MAINVIEW for IMS			
MAINVIEW for IMS applies to both MAINVIEW for IMS and MAINVIEW for DBCTL.			

BMC Software, Inc., Confidential and Proprietary Information

Table 1-3 Application Transfer Commands (Part 4 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
IAO	ALERTS | ALE	ALERTS Overview	MAINVIEW AutoOPERATOR
IAO | IMS	AN	Analyzer Display Services	MAINVIEW for IMS
IAO | IMS	AR	Data Entry Database Areas	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	AT	Active Timer Requests	MAINVIEW for IMS
IAO | IMS	CT	View Current Traces	MAINVIEW for IMS
IAO | IMS	DATABASE | DAT	Database	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	DB	Database	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	DE	Data Entry Databases	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	DM	Display Monitor Requests	MAINVIEW for IMS
IAO | IMS	DW	Display Warnings	MAINVIEW for IMS
IAO | IMS	EXEC svc parm1, parm2	Execute a service with passed parameters	MAINVIEW for IMS
IAO | IMS	EX	Status/Exception	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	GC	General Commands	MAINVIEW for IMS
IAO | IMS	HT	HISTORY Traces	MAINVIEW for IMS
IAO | IMS	MAINVIEW for IMS	MAINVIEW for IMS Performance Management	MAINVIEW for IMS
IAO | IMS	ISC	ISC Links	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	LINE	BTAM Lines	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	LTERM | LT	LTERMS	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	MN	Data Collection Monitors	MAINVIEW for IMS

BMC Software, Inc., Confidential and Proprietary Information

Table 1-3 Application Transfer Commands (Part 5 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
IAO | IMS	MS	Main Storage Databases	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	NODE | NO	VTAM nodes	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	PD	MAINVIEW for IMS Performance Management	MAINVIEW for IMS
IAO | IMS	PM	MAINVIEW for IMS Performance Management	MAINVIEW for IMS
IAO | IMS	PROGRAM | PR	Program	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	RC	Fast Path Routing Codes	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	REGION | REG	IMS Regions	MAINVIEW AutoOPERATOR for IMS
IAO | IMS	SD	Statistics and Defaults	MAINVIEW for IMS
IAO | IMS	SM	Start Monitors	MAINVIEW for IMS
IAO | IMS	ST	Start Trace	MAINVIEW for IMS
IAO	STATUS | STA	Status/Exception	MAINVIEW AutoOPERATOR
IMS	STAT | STA	IMS SYSTEM STATUS	MAINVIEW for IMS
IAO | IMS	TRANSACTION | TR	Transaction	MAINVIEW AutoOPERATOR
IAO | IMS	VT	View Current Traces	MAINVIEW for IMS
IAO	XALRTS | XAL	ALERTS Detail	MAINVIEW AutoOPERATOR
MVS Operator Workstation			
MAO	ALERTS | ALE	ALERTS Overview	MAINVIEW AutoOPERATOR
MAO	DA	Address Spaces	MAINVIEW AutoOPERATOR for OS/390

BMC Software, Inc., Confidential and Proprietary Information

Table 1-3 Application Transfer Commands (Part 6 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
MAO	DASD	DASD Status/Control	MAINVIEW AutoOPERATOR for OS/390
MAO	DISPLAY | DISP	Address Spaces	MAINVIEW AutoOPERATOR for OS/390
MAO	ENQUEUES | ENQ	Enqueue/Reserve	MAINVIEW AutoOPERATOR for OS/390
MAO	OPERATOR | OPE	Operator Requests	MAINVIEW AutoOPERATOR for OS/390
MAO	OR	Operator Requests	MAINVIEW AutoOPERATOR for OS/390
MAO	REQUESTS | REQ	Operator Requests	MAINVIEW AutoOPERATOR
MAO	RESERVES | RES	Enqueue/Reserve	MAINVIEW AutoOPERATOR for OS/390
MAO	STATUS | STA	System Status	MAINVIEW AutoOPERATOR for OS/390
MAO	TAPE | TAP	Tape Status/Control	MAINVIEW AutoOPERATOR for OS/390
MAO	MAJNODE | MAJ	VTAM Major Nodes	MAINVIEW AutoOPERATOR for OS/390
MAO	APPL	VTAM Applications	MAINVIEW AutoOPERATOR for OS/390
MAO	CDRM	VTAM CDRMs	MAINVIEW AutoOPERATOR for OS/390

BMC Software, Inc., Confidential and Proprietary Information

Table 1-3 Application Transfer Commands (Part 7 of 7)

Product Line Transfer Command	Application Transfer Command	Description	Product
MAO	CDRSC | CDRS	VTAM CDRSCs	MAINVIEW AutoOPERATOR for OS/390
MAO	LINE	VTAM Lines	MAINVIEW AutoOPERATOR for OS/390
MAO	CLSTR | CLS	VTAM Clusters	MAINVIEW AutoOPERATOR for OS/390
MAO	TERMINAL | TERM	VTAM Terminals	MAINVIEW AutoOPERATOR for OS/390
MAO	XALRTS | XAL	ALERTS Detail	MAINVIEW AutoOPERATOR for OS/390

BMC Software, Inc., Confidential and Proprietary Information

CLIST and REXX EXECs Summary

This chapter briefly discusses how to pass parameters to MAINVIEW AutoOPERATOR CLIST and REXX EXECs and how to document EXECs and Rules.

Passing Information to CLIST EXECs: PROC Statement

The PROC command must be the first statement in the EXEC, as described in the IBM publication *TSO Extensions Version 2: CLISTs*:

```
PROC parmnum p1 ... px
```

where

`parmnum` is the maximum number of parameters passed to the EXEC.

`p1 ... px` are the variable names of the parameters being passed.

The actual parameter names need not be p1 through px. If fewer parameters are passed than parameters listed on the PROC statements, the trailing variables each contain a single period.

The number and types of parameters being passed depend on how the EXEC was initiated.

BMC Software, Inc., Confidential and Proprietary Information

Passing Information to REXX EXECs

EXECs written in REXX receive the same set of parameters as CLISTs. Include the following command near the beginning of the EXEC to obtain the parameters passed:

```
PARSE UPPER ARG p1 ... px
```

where `p1` through `px` are positional parameters passed. The actual parameter names need not be p1 through px. If fewer parameters are passed than parameters listed on the PROC statements, the trailing variables each contain a single period.

MAINVIEW AutoOPERATOR passes all variables required by the type of EXEC **plus** a character string of ". . . . . ". The sum of the number of characters in this string and the number of characters in the variables passed to the EXEC is 255. This string of periods is concatenated to the value of the last positional parameter passed to the EXEC. To avoid this string of periods, code a single period (.) or any valid variable name after the last variable name in the template.

EXEC Parameters

The following tables list the parameters passed for each method of EXEC initiation. Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide* for complete information.

BMC Software, Inc., Confidential and Proprietary Information

Table 2-1 Positional Parameters for the PROC/ARG Statement: Part 1

Positional Parameter	Rule-initiated EXEC	ALERT-initiated EXEC	User-initiated EXEC	EXEC-initiated EXEC
1	Message ID or name of first parameter	EXEC name or first parameter	EXEC name or first parameter	EXEC name or first parameter
2	1st word of msg	*	1st optional parameter	1st optional parameter
3	2nd word of msg	*	2nd optional parameter	2nd optional parameter
4	3rd word of msg	*	3rd optional parameter	3rd optional parameter
5	4th word of msg	*	4th optional parameter	4th optional parameter
6	5th word of msg	*	5th optional parameter	5th optional parameter
7	6th word of msg	*	6th optional parameter	6th optional parameter
8	7th word of msg	*	7th optional parameter	7th optional parameter
9	8th word of msg	*	8th optional parameter	8th optional parameter
10	9th word of msg	*	9th optional parameter	9th optional parameter
11	10th word of msg	*	10th optional parameter	10th optional parameter

Warning: For both Rule- and ALERT-initiated EXECs, the values passed to the EXECs vary depending on whether optional parameters are specified.

Note: If parameters are specified where the EXEC is called, parameter 1 is the first parameter specified, instead of the Message ID or the EXEC name.

Refer to the *MAINVIEW AutoOPERATOR Advanced Automation Guide* for more information.

BMC Software, Inc., Confidential and Proprietary Information

Table 2-2 Positional Parameters for the PROC/ARG Statement: Part 2

Positional Parameter	Time-initiated EXEC	Externally initiated EXEC	End-of-Memory EXEC or IMFEOM
1	EXEC name	EXEC name	*EOM*
2	Target name	1st optional parameter	NORMAL or ABNORMAL
3	IMS ID (IAO only)	2nd optional parameter	N/A
4	BBI-SS subsystem identifier	3rd optional parameter	N/A
5	Current Gartering date	4th optional parameter	N/A
6	Time the EXEC is scheduled	5th optional parameter	N/A
7	Day of the week	6th optional parameter	N/A
8	Current Julian date	7th optional parameter	N/A
9	Elapsed time of the active IMS/VS. (IAO only)	8th optional parameter	N/A
10	The IMS/VS restart type (IAO only)	9th optional parameter	N/A
11	Number of times the EXEC has been invoked (IAO only)	10th optional parameter	N/A
Note: Each EXEC type is discussed separately in the *MAINVIEW AutoOPERATOR Advanced Automation Guide*.			

Documenting EXECs and Rules

This section describes how and why you should document Rules and EXECs (documentation is not required).

BMC Software, Inc., Confidential and Proprietary Information

Documenting EXECs

By using documentation boxes in EXECs, you can use the MAINVIEW AutoOPERATOR EXEC Manager application to locate specific or related EXECs quickly and efficiently. Refer to the *MAINVIEW AutoOPERATOR Basic Automation Guide* for information about using this application. If you do use a documentation box, include near the beginning of each EXEC.

Figure 2-1 Example of REXX Documentation Box

```
/* REXX                                          */
/*-----------------------------------------------*/
/*   DOC GROUP(                )      8 CHAR     */
/*   DOC  FUNC(                )      8 CHAR     */
/*   DOC  CODE(     )                 2 CHAR     */
/*   DOC  DESC(     )                25 CHAR     */
/*   DOCAUTHOR(                )      8 CHAR     */
/*-----------------------------------------------*/
```

Figure 2-2 Example of CLIST Documentation Box

```
PROC 1 MSGID
/*-----------------------------------------------*/
/*   DOC GROUP(                )      8 CHAR     */
/*   DOC  FUNC(                )      8 CHAR     */
/*   DOC  CODE(     )                 2 CHAR     */
/*   DOC  DESC(     )                25 CHAR     */
/*   DOCAUTHOR(                )      8 CHAR     */
/*-----------------------------------------------*/
```

BMC Software, Inc., Confidential and Proprietary Information

Figure 2-3 Example of Completed CLIST Documentation Box

```
*-----------------------------------------------*/
/*   DOC    GROUP(MVS)                           */
/*   DOC    FUNC(MONITOR)                        */
/*   DOC    CODE(PR)                             */
/*   DOC    DESC(Create ALERT if CPU high)       */
/*   DOC    AUTHOR(TAT2)                         */
/*-----------------------------------------------*/
```

Documenting Rules

On the Rule Processor Detail Control panel, you can include similar documentation for a Rule as you do for an EXEC; see Figure 2-4.

Figure 2-4 Rule Processor Detail Control Panel

```
⋮
Application information:
Group       ===> MVS      Function     ===> SMF      Code ===> CD
Author      ===> SMF      Description  ===> DUMP SMF DATASET
Last modified by           on         at

  Press ENTER to continue, END to apply changes, CANCEL to cancel changes
```

BMC Software recommends that, if you choose to document Rules or EXECs, you apply similar coding standards specific to your site's needs.

BMC Software, Inc., Confidential and Proprietary Information

TSO Variables for CLIST and REXX EXECs

This chapter describes MAINVIEW AutoOPERATOR-supplied variables for MAINVIEW AutoOPERATOR CLIST and REXX EXECs.

MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs

Table 3-1 lists the MAINVIEW AutoOPERATOR-supplied TSO variables that you can use in CLIST or REXX EXECs. To use these variables with CLIST EXECs, place an ampersand sign (&) in front of the variable name.

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 1 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFACCTG	All accounting fields for a particular event. The accounting field values are separated by blanks. Maximum length is 142.	Rule-initiated EXECs only
IMFALID	Alarm ID associated with an alarm created by MAINVIEW Alarm Manager	Rule-initiated EXECs only

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 2 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFALPRI	User-assigned priority of the alarm. Possible values are 1. Critical 2. Major 3. Minor 4. Warning 5. Informational 6. Clearing	Rule-initiated EXECs only
IMFALQID	Name of the queue to which the alarm was assigned	Rule-initiated EXECs only
IMFALRM	Either Y (sound an alarm) or N (do not sound an alarm)	Rule-initiated EXECs only
IMFCC	Condition code set for each IMFEXEC statement: IMFCC = 00 Normal completion IMFCC = 04 Warning condition; not necessarily an error IMFCC = 08 Exception condition or command not found IMFCC = 12 Error condition; did not complete operation IMFCC = 16 Error condition IMFCC = 20 Severe error condition Refer to the specific IMFEXEC statement for the exact codes.	All EXEC types
IMFCLOCK	A value that represents the time difference (in seconds) between when two Rules can fire. For example, a Rule can set a shared variable to IMFCLOCK plus 15 seconds. A second Rule can check the value of IMFCLOCK to ensure that 15 seconds have passed and then the second Rule can fire.	Rule-initiated EXECs only
IMFCNTXT	Name of the context of the alarm	Rule-initiated EXECs only
IMFCONID	Console ID of the message, if message was issued for a specific console. Valid only for messages captured through the Rule Processor application.	Rule-initiated EXECs only
IMFCONNM	Console name to which the WTO was issued. Valid only for MVS SP4 and later.	Rule-initiated EXECs only

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 3 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFDAY	Three-character day of the week: MON, TUE, WED, THU, FRI, SAT, SUN.	All EXEC types
IMFDDNAM	The DDNAME specified by the user to generate an external events (EXT event type). EXT events are generated by using the SUBSYS= parameter on a DD statement in JCL. Refer to “EXT Events” in the *MAINVIEW AutoOPERATOR Basic Automation Guide* for more information about EXT events.	Rule-initiated EXECs only
IMFDOMID	DOM ID associated with a WTO that initiated an EXEC	Rule-initiated EXECs only
IMFEID	EXEC identification number, 1 to 99999, assigned to each execution by the EXEC manager. The EXEC MANAGER will not assign the same number to two EXECs in the running or deferred queues, except when an EXEC selected with WAIT=(YES) has the same IMFEID as the calling EXEC.	All EXEC types
IMFENAME	Name of EXEC.	All EXEC types
IMFEROUT	List of routing codes that were assigned to the WTO that triggered the EXEC, such as 1 2 5 9. This variable is defined only for EXECs initiated as a result of a WTO. IMFEROUT supports route codes up to 128.	Rule-initiated EXECs only
IMFETYPE	The event type that caused the Rule to fire. Possible values for IMFETYPE are as follows: • MSG • CICS • CMD • JRNL • IMS • ALRT • DB2 • TIME • ALRM • EXT • VAR • MQS • JES3	Rule-initiated EXECs only
IMFEVFRD	Number of Rules that have fired for a specific event.	Rule-initiated EXECs only

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 4 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFGROUP	RACF group ID for the address space that issued the message. The group ID is taken from the GROUP= parameter on the job card.	Rule-initiated EXECs only
IMFJCLAS	Job class name from the job card of the batch job that has generated the message	Rule-initiated EXECs only
IMFJNUM	JES Job number of JOB/TSU/STC that issued the message; the job number is five digits, right-justified with preceding zeros.	Rule-initiated EXECs only
IMFJTYPE	Type of job issuing the message: **J** -- Batch Job **T** -- TSO User **S --** Started Task	Rule-initiated EXECs only
IMFLPROD	Name of the product associated with the alarm	Rule-initiated EXECs only
IMFLTYPE	A literal value associated with the alarm; possible values can be START or STOP.	Rule-initiated EXECs only
IMFLUSER	User-specified user ID associated with the alarm	Rule-initiated EXECs only
IMFMPFAU	Value of a message from the MPF AUTO keyword Use this variable to determine the value of the MPF AUTO keyword for a message.	Rule-initiated EXECs only
IMFMPFSP	Value of a message from the MPF SUP keyword Use this variable to determine the value of the MPF SUP keyword for a message.	Rule-initiated EXECs only
IMFMSTYP	Two-character variable for the message type. This variable is only for the CMD and MSG event types. Valid values for the first character are **N --** A regular WTO **W --** A regular WTOR **M** -- A major line of a multi-line WTO (MLWTO) Valid values for the second character are: **C --** Command **R --** Command response	Rule-initiated EXECs only

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 5 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFNOL	Number of lines in WTOR that caused the EXEC to be invoked or the number of lines returned from a service. This value is limited to 9999 lines for Rule-initiated EXECs and for data returned by IMFEXEC service with response.	All EXEC types
IMFOASID	Originating Address Space ID (ASID) of the message. For IMFEOM, it is set to the ASID that is being terminated. For ORIGIN=JRNL, it is set to the subsystem ASID name.	Rule-initiated EXECs only
IMFODATE	Date when the message or alarm was issued. Valid only for messages captured through the Rule Processor. The date format is in Julian calendar format; for example: 01.100, where: **01** are the last two digits of the year 2001. **100** is the 100th day of the year. In a non-leap-year, this number is equal to March 10.	Rule-initiated EXECs only
IMFODESC	List of descriptor codes assigned to the WTO that triggered the EXEC, such as 2 11. This variable is defined only for EXECs initiated as a result of a WTO.	Rule-initiated EXECs only
IMFOJOB	For WTOs, IMFOJOB contains the job or started task that issued the WTO. For CICS messages, IMFOJOB contains the CICS region name that the subsystem issued the message for, which is useful when monitoring multiple CICS regions with one BBI-SS PAS. For DB2 messages, IMFOJOB contains the DB2 region name that the subsystem issued the message for, which is useful when monitoring multiple DB2 regions with one BBI-SS PAS. For IMS messages, IMFOJOB contains the • IMS jobname for IMS MTO messages • IMS jobname for commands (and their responses) entered from MAINVIEW AutoOPERATOR • originating LTERM for commands (and their responses) entered from an IMS LTERM For BBI-SS PAS Journal messages issued by an EXEC, IMFOJOB contains the user ID of the person who invoked the EXEC.	All EXEC types
	For time-initiated EXECs, IMFOJOB contains the user ID associated with that EXEC. This ID may be the user ID passed on the command or it may default to the value of the AUTOID keyword specified in BBPARM member BBIISP00.	

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 6 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFXOJOB	Contains the name of the original job or started task that requested the WTO to be issued by another address space. The contents of IMFXOJOB are only meaningful if the WTO is issued by another address space, otherwise its contents are identical to IMFOJOB.	All EXEC types
IMFOQID	CICS transient data queue name if the source of the message is CICSTD.	Rule-initiated EXECs only
IMFORGN	Origin of EXEC-Jobname/USERID causing the EXEC to be invoked. For EXECs triggered by the Rule Processor, IMFORGN contains the BBI-SS PAS ID, so that EXECs invoked on remote systems that are triggered by message filters on the local system can use authorized services, such as SYSPROG services. Security checking is done against a BBPARM member in the remote system with the name of the BBI-SS PAS ID.	All EXEC types
IMFORGSS	BBI subsystem ID of the BBI-SS PAS that originated an EXEC. If originated locally, IMFORGSS is the same as QIMFID.	All EXEC types
IMFOROUT	List of routing codes that were assigned to the WTO that triggered the EXEC, such as 1 2 5 9. This variable is defined only for EXECs initiated as a result of a WTO.	Rule-initiated EXECs only
IMFOTIME	Time when the message was issued. Valid only for messages (also known as events) captured through the Rule Processor. The valid form of the variable is hh:mm:ss for all Rule event types except the MSG event type. For MSG events, the valid form of the variable is hh.mm.ss. For ALRM events, the time represents the time the exception occurred.	Rule-initiated EXECs only
IMFPCMD	The PCMD associated with the alarm.	Rule-initiated EXECs only
IMFPOST	A 1-to 255-character code received from an EXEC that issues the IMFEXEC POST command against an ECB with the same name that the current EXEC is waiting on.	All EXECs
IMFPRIO	Dispatching priority of the currently running EXEC after the IMFEXEC CHAP command has been issued.	All EXECs

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 7 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFRC	The return code set by a called EXEC with WAIT(YES) or the return code set by a non-MAINVIEW AutoOPERATOR command or program.	EXEC-initiated EXECs only
IMFREPLY	Reply ID of the WTOR message. Valid only for messages captured through the Rule Processor.	Rule-initiated EXECs only
IMFRLFRD	The number of times a Rule was fired.	Rule-initiated EXECs only
IMFRLID	The Rule identifier that fired an EXEC.	Rule-initiated EXECs only
IMFRLMAT	The number of times the Rules search criteria were matched.	Rule-initiated EXECs only
IMFRLSET	The name of the Rule Set the Rule belongs to.	Rule-initiated EXECs only
IMFRLSTA	The Rule status: • **TEST** indicates that the status of the Rule that invoked the EXEC is in a TEST state. • **ACTIVE** indicates that the status of the Rule that invoked the EXEC is in ACTIVE state.	Rule-initiated EXECs only
IMFRUSER	RACF user ID for the address space that issued the message. The user ID is taken from the USER= parameter on the job card.	Rule-initiated EXECs only
IMFSCOPE	Name of the scope associated with the alarm	Rule-initiated EXECs only
IMFSTOKN	Address Space STOKEN. This name is unique for the life of the IPL.	Rule-initiated EXECs or End-of-Memory initiated EXECs
IMFSYSID	Originating jobname For CICS messages, IMFSYSID contains the BBI started task name.	Rule-initiated EXECs only
IMFTEXT	Character text that caused the EXEC to be scheduled.	All EXEC types
IMFTOKEN	Token ID of the message. Same as hardcopy ID. Used to attach MLWTO Minor/Major Lines. Valid only for messages captured through the Rule Processor.	Rule-initiated EXECs only

BMC Software, Inc., Confidential and Proprietary Information

Table 3-1 MAINVIEW AutoOPERATOR-Supplied TSO Variables for CLIST and REXX EXECs (Part 8 of 8)

Variable Name	Description	Applicable for which EXEC Type
IMFVIEW	Name of the view associated with the alarm	Rule-initiated EXECs only
IMFWTDOM	DOM ID associated with a WTO issued by IMFEXEC WTO command	All EXEC types
IMFWTCON	Created when a reply is successfully received. The eight-character name of the console from which the reply to the WTOR was entered.	
Note that these variables are carried over to the TSO pool created for an EXEC called using the IMFEXEC SELECT command with parameter WAIT(YES) specified.		

MAINVIEW AutoOPERATOR-Supplied SHARED Variables for CLIST and REXX EXECs

Table 3-2 MAINVIEW AutoOPERATOR-Supplied SHARED Variables for CLIST and REXX EXECs (Part 1 of 2)

Variable	Description
QAOREL	five-character string indicating the release of MAINVIEW AutoOPERATOR The string takes the format **v.r.m**, where **v** is the version level **r** is the release level **m** is the modification level
QIMFID	BBI Subsystem ID of this BBI-SS
QIMGSTA	(IMS & DB2 Performance Products only) status of BBI-SS PAS Image logging as ACTIVE or INACTIVE
QIMGSUF	(IMS & DB2 Performance Products only) suffix of the current or last active BBI-SS Image data set If logging has never been initialized, the value is null.

BMC Software, Inc., Confidential and Proprietary Information

Table 3-2 MAINVIEW AutoOPERATOR-Supplied SHARED Variables for CLIST and REXX EXECs (Part 2 of 2)

Variable	Description
QJESCHAR	JES2 command character
QJNLSTA	status of BBI-SS Journal logging as ACTIVE or INACTIVE
QJNLSUF	suffix of the current or last active BBI-SS Journal data set If logging has never been initialized, the value is null.
QSMFID	SMF system ID of the system where the EXEC is running
QSSNAME	jobname of the SS address space
QIMSREL	contains the IMS release number
QIMSNAME	jobname of the IMS/VS being monitored by this BBI-SS
QIMSID	IMSID of the IMS/VS being monitored This IMSID is available only when IMS/VS is active. This IMSID is the same as the IMS/VS identified by QIMSNAME.
QIMSSTA	status of IMS/VS (ERE, WARM, COLD, or INACT)

ALERT Variables for REXX and CLIST EXECs and MAINVIEW AutoOPERATOR Rules

The following are variables that are created when ALERTs are created.

Table 3-3 ALERT Variables for REXX and CLIST EXECs and MAINVIEW AutoOPERATOR Rules (Part 1 of 2)

Name	Contents	Length/Format
AMFKEY	key of the ALERT	1–64 / Character
AMFTEXT	text of the ALERT	0–255 / Character
AMFALARM	alarm value of the ALERT	1 / Y (YES) or N (NO)
AMFCOLOR	color of the ALERT	6 / As specified by COLOR parameter
AMFEDISP	keep or delete the ALERT at the final escalation level	1 / Character (K or D)

BMC Software, Inc., Confidential and Proprietary Information

Table 3-3 ALERT Variables for REXX and CLIST EXECs and MAINVIEW AutoOPERATOR Rules (Part 2 of 2)

Name	Contents	Length/Format
AMFEDIR	increase or decrease the priority of the ALERT when it is escalated	1 / Character (U or D)
AMFEEXEC	name of EXEC and EXEC parameters scheduled at final escalation priority	0 to 255 / Character
AMFEXEC	EXEC and EXEC parameters associated with the ALERT	0–256 / Character
AMFHELP	Extended ALERT member name	8 / Character
AMFEINT1 AMFEINT2 AMFEINT3 AMFEINT4 AMFEINT5 AMFEINT6	number (in minutes) from 0 to 9999	4 / Numeric (or null)
AMFIDATE	date the ALERT was issued	9 / DD-MMM-YY
AMFITIME	time the ALERT was issued	8 / hh:mm:ss
AMFPCMD	primary command specified in the ALERT	0–256 / Character
AMFQUEUE	name of queue for the ALERT	8 / Character
AMFPRIOR	priority of the ALERT	13 / As specified in the PRIORITY parameter
AMFSSID	system from which the ALERT was issued	8 / Character
AMFTGT	target to which the ALERT was issued	1–8 / Character
AMFORGN	origin of the ALERT	1–8 / Character
AMFPUB	value of PUBLISH when the ALERT was created	NO | ADD | REPLACE
AMFPUBS	setting of PUBLISH in either the Rule or the EXEC that generated the ALERT	NO | ADD | REPLACE | null
AMFRTAIN	value specifies whether to retain an ALERT across BBI-SS PAS warm and cold starts	1 / Character (Y or N)
AMFUDATA	user data string	0–256 / Character
AMFUSER	name of the user ID that the ALERT is addressed to	8 / Character
AMFPSYS	value for SYSTEM keyword (could be either YES or NO)	1 / Character (Y or null)

BMC Software, Inc., Confidential and Proprietary Information

OSPI Control Variables in LOCAL Variable Pool—REXX and CLIST EXECs

Note

The OSI prefix can be changed to any other characters on the OSPI LOGON command to help differentiate variables between sessions. The default is OSI.

Table 3-4 OSPI Control Variables in LOCAL Variable Pool—REXX and CLIST EXECs

Variable	Description
OSIAPPL	name of the VTAM application connected to this OSPI session
OSICOL	current column position of cursor. Can be from 1 to 80
OSIKSTAT	current keyboard status. Can contain either LOCKED or UNLOCKED.
OSILNCNT	number of rows for the terminal type to be emulated. Can be from 24 to 43
OSILNxx	contents of each line of the virtual screen buffer image These variables also contain field attributes and must be taken into account when altering the contents.
OSIROW	current row position of the cursor (can be from 1 to 43)
OSISESS	value used in conjunction with the SESSION keyword to identify the OSPI session

BMC Software, Inc., Confidential and Proprietary Information

RxD2/LINK Control Variables for REXX EXECs Only

The following variables are included and available if you have RxD2/LINK installed and are using RxD2/LINK within an MAINVIEW AutoOPERATOR EXEC.

Table 3-5 RxD2/LINK Control Variables for REXX EXECs Only

Variable	Description
$DB2RC	return code from DB2 CAF (SIGNON and SIGNOFF) or IFI (COMMAND and READS)
$DB2RESP	Stem variable for DB2 command responses $DB2RESP.0 contains the number of $DB2RESP.*n* variables that were set to contain responses. $DB2RESP.1 through $DB2RESP.*n* contain the actual command responses.
$DB2RSN	reason code from DB2 CAF (SIGNON and SIGNOFF) or IFI (COMMAND and READS)
SQLCODE	SQLCODE from SQLCA
SQLEMSG	Stem variable for SQL error messages These error messages, like the ones from SPUFI, describe the SQL error in detail.
SQLERRD3	SQLERRD3 from SQLCA
SQLERRM	SQLERRM from SQLCA
SQLWARN	SQLWARN from SQLCA
RCTL	RESERVED for RxD2/LINK Unpredictable results may occur if this variable is used or altered.
$SQLCA	RESERVED for RxD2/LINK Unpredictable results may occur if this variable is used or altered.
$WQAL	RESERVED for RxD2/LINK Unpredictable results may occur if this variable is used or altered.
$WBUF	RESERVED for RxD2/LINK Unpredictable results may occur if this variable is used or altered.
$IFCA	RESERVED for RxD2/LINK Unpredictable results may occur if this variable is used or altered.

BMC Software, Inc., Confidential and Proprietary Information

Functions and Commands for REXX EXECs

The section lists

- restrictions of TSO/E functions and commands for REXX EXECs[1]
- RxD2/LINK REXX functions and extensions

Restricted TSO/E Commands and Functions

MAINVIEW AutoOPERATOR supports the following commands and functions if you specify ADDRESS MVS prior to issuing the command.

Table 3-6 MAINVIEW Supported TSO/E Commands and Functions

Command	Function
DELSTACK	Deletes the most recently created data stack that was created by the NEWSTACK command, and all elements on it. If a new data stack was not created, DELSTACK removes all the elements from the original data stack.
DROPBUF	Deletes the most recently created data stack buffer that was created by the MAKEBUF command. To remove a specific data stack buffer and all buffers created after it, issue the DROPBUF command with the number of the buffer.
MAKEBUF	Creates a new buffer on the data stack.
NEWSTACK	Creates a new data stack and hides or isolates the current data stack. Elements on the previous data stack cannot be accessed until a DELSTACK command is issued to delete the new data stack.
PROMPT( )	Returns to the previous setting of prompting for the EXEC, which is always OFF when running under MAINVIEW AutoOPERATOR.
QBUF	Queries the number of buffers that were created on the data stack with the MAKEBUF command.
QELEM	Queries the number of data stack elements that are in the most recently created data stack buffer.
QSTACK	Queries the number of data stacks in existence for an EXEC that is executing.
SUBCOM	Queries the existence of a specified host command environment.

1. The description of these TSO/E REXX commands and functions are from the IBM publication, *OS/390 TSO/E REXX Reference*.

BMC Software, Inc., Confidential and Proprietary Information

RxD2/LINK REXX Function Extensions

Table 3-7 RxD2/LINK REXX Function Extensions

Functions	Description
CONVSTCK(tod)	Converts the 8-byte TOD clock (from the STCK instruction) into character format of YYYYDDDHHMMSSTH. Valid from 1/1/1988 onward.
CTOD(tod)	Converts the 8-byte TOD clock (from the STCK instruction) into the character format of HHMMSSTH.
F2C(f)	Performs a floating point conversion on f, returning f in character format.
GBLVAR({GETV | SETV | DROP | UPDV},varname)	Creates and manages the global variable environment. The global variable is created automatically the first time this function is used. Subsequent environments share the same environment. The environment is destroyed at the EOT of the task, that initially created the environment: • GETV obtains the global variable name and places it into the local variable *varname*. • SETV creates the global variable *varname* using the local variable. DROP deletes the global variable *varname*. • UPDV updates the global variable *varname* from the local variable *varname*.
P2C(p)	Reformats a packed-decimal variable *p* into character format.
UENV (hcename,pgm)	Identifies to REXX Host Command Environment (HCE) called hcename, such that pgm will receive control for ADDRESS hcename. The hcename currently is required to be DB2.
VARSPF(varname)	Converts compound REXX variables (AA.1) into a CLIST or ISPF variable (AA1).
WAITSEC(n)	Suspends EXEC processing for *n* seconds.

BMC Software, Inc., Confidential and Proprietary Information

RxD2/LINK REXX Common Function EXECs

Table 3-8 RxD2/LINK REXX Common Function EXECs

Common Function EXECs	Description
RXBKLINE(mxline,iline)	Truncates the character text in ILINE at a word boundary to a length no greater than MXLEN. It is useful in displaying a long SQL statement. If either argument is null, a null string is returned.
RXQCHAR(wname,wdata)	Builds a predicate for the character-type column WNAME from the string entered as a qualifier in WDATA. It is used to generate SQL predicates from user input specifying a selection qualifier for a column of a table.
RXSAMPEX( )	Is a sample EXEC to process SQL statements or DB2 commands and display the results in line mode. It does not require ISPF and therefore is usable in any address space; for example, batch jobs, NetView, or AutoOPERATOR EXECs.
RXSETSQL( )	Constructs an SQL statement from the text pointed to by a cursor in an ISPF/PDF edit panel.
RXVODS(wdsn)	Verifies that the data set name in wdsn is valid. It checks that the data set is cataloged and what type of data set it is (sequential or a PDS). These commands and functions are available only if RxD2/LINK is installed and made available to MAINVIEW AutoOPERATOR.
RXQNUM(wname,wdata)	Builds a predicate for the numeric-type column WNAME from the string entered as a qualifier in WDATA. It is used to generate SQL predicates from user input specifying a selection qualifier for a column of a table.

Command Restrictions in EXECs

You cannot use the following commands in AutoOPERATOR EXECs.

ACCOUNT	LISTLDX	RUN
ALTFILE option of ALLOCATE	LOADGO	SLIP
CANCEL	LOGOFF	STATUS
DISPLAY	LOGON	STOPMN

BMC Software, Inc., Confidential and Proprietary Information

EDIT	MONITOR	SUBMIT
HELP	OUTPUT	TERMINAL
LISTALC	OPERATOR	TEST
LISTBC	PROFILE	
LISTCT	PROTECT	
	RENAME	

MAINVIEW AutoOPERATOR EXECs do not support the following TSO CLIST statements:

READ	WRITENR	TERMIN

MAINVIEW AutoOPERATOR does not support the REXX language facilities Interactive Trace, such as `TRACE R?.` MAINVIEW AutoOPERATOR also does not support the following immediate commands:

Halt Interpretation (HI)	Trace Start (TS)	Resume Typing (RT)
Halt Typing (HT)	Trace End (TE)	

BMC Software, Inc., Confidential and Proprietary Information

IMFEXEC Command Syntax Summary

Commands and Examples

This section lists all the MAINVIEW AutoOPERATOR IMFEXEC command statements, their syntax, function, parameters available, condition codes (where applicable), and a short example. Examples are provided for CLIST EXECs and then REXX EXECs.

Note that in Table 4-1, the syntax of the command statement has been condensed to show only the command and its parameters and keywords. However, IMFEXEC is required before each command, for example:

Example

IMFEXEC ‘command’ ‘parms and keywords’

Variable names are limited to 32 characters in length. The first character of the variable must be alphanumeric or one of the following special characters:

- $
- @
- #

Every command returns a condition code in the variable IMFCC in the TSO pool.

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 1 of 26)

Syntax:	**ALERT** alert-key 'alert text' [FUNCTION(ADD | CREATEQ | DELETE | DELETEQ | LISTQ | READQ | COUNT)] [ALARM(NO | YES) [COLOR(RED | PINK | YELLOW | LTBLUE | DKBLUE | GREEN | WHITE)] [DISPOSE(KEEP | DELETE)] [ESCALATE(UP | DOWN)] [ESCEXEC('execname p1 p2 p3 ... pn')] [EXEC('execname p1 p2 ... pn')] [HELP(panelname)] [INTERVAL(nnnn,nnnn,nnnn,nnnn,nnnn,nnnn)] [PCMD('command string')] [POSITION(n)] [PRI(CRITICAL | MAJOR | MINOR | WARN | INFO | CLEARING)] [PUBLISH(REPLACE | ADD | NO] [QUEUE(MAIN | queue name)] [RETAIN(YES | NO)] [SS | SSID(subsystem identifier)] [SYSTEM(YES | NO)] [TARGET(target name)] [TEXT('text string')] [ORIGIN(origin)] [UDATA('user data')] [USER(user name)]	
Function:	Manipulates ALERT messages that notify operators of exception situations within the BBI-SS network.	
Note:	Multi-line alerts can be created by embedding a /N within the alert text to force a new line.	
Cond Codes Without TARGET coded:	**0**	Command was executed successfully.
	4	End of Queue or trying to read beyond queue end (READQ)
	8	Queue not found (READQ, COUNT, or DELETE)
	16	Syntax error
Cond Codes With TARGET coded:	**0**	Alert was issued on remote system.
	8	Node was not found.
	12	Target was not found.
	16	Node was not available.
Example:	IMFEXEC ALERT &NODE DOWN 'TERMINAL, &NODE IS DOWN' PRI(WARNING) QUEUE(NET) "IMFEXEC ALERT "node" DOWN 'TERMINAL, "node" is down.' PRI(WARNING) QUEUE(NET)"	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 2 of 26)

Syntax:	**BKPT**	
Function:	Use this command to halt execution of an EXEC while it is being tested by the EXEC Testing Facility.	
Note:	If you use this command within an EXEC and execute the EXEC outside of the EXEC Testing Facility, the statement has no effect.	
Syntax:	**CHAP** (n)	
Function:	Use this command with a numeric parameter (n) to change the dispatching priority of the EXEC either up or down.	
Note:	Specifying zero (0) returns the current priority.	
Cond Codes:	**16**	Syntax error.
Example:	IMFEXEC CHAP (-10) "IMFEXEC CHAP (-10)"	
Syntax:	**CICS** common information	
Function:	Documents information common to the IMFEXEC CICS commands listed below.	
Note:	If IMFCC contains a value other than 0, variable IMFNOL contains the number of lines returned and variables LINE1 through LINExx contain the response from the service. The condition codes listed below are for all IMFEXEC CICS services.	
Cond Codes:	**0**	Normal completion
	4	Warning condition; not necessarily an error
	8	Exception condition
	12	Error condition; operation not completed. For independent actions, the region was not available. For dependent actions, the region was not connected to the BBI-SS PAS
	16	Error condition
	20	Severe error condition
Syntax:	**CICS ACQUIRE TERMINAL** name	
Function:	Issues a VTAM acquire request for a terminal.	
Cond Codes:	See “CICS common information” for condition codes.	
Example:	IMFEXEC CICS ACQUIRE TERMINAL &TERMID "IMFEXEC CICS ACQUIRE TERMINAL" termid	
Syntax:	**CICS ALLOC** filename [TO] dsname [LOCAL]	
Function:	Allocates a data set to either the CICS/VS region or the BBI-SS.	
Cond Codes:	See “CICS common information” for condition codes.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 3 of 26)

Example:	IMFEXEC CICS ALLOC MASTER TO USER.VSAM.MASTER "IMFEXEC CICS ALLOC MASTER TO USER.VSAM.MASTER"
Syntax:	**CICS ALTER** MAXTASK | ICV | ICVR | CLASSn | TCLASS | SYSTEMP | DUMPDS | DUMPCODE | TCPIPSERVICE | JVMPOOL values
Function:	Changes the value of the CICS request class maximum settings and statistics for each class. Also can be used to set other types of operating values.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS ALTER MAXTASK 35 IMFEXEC CICS ALTER CLASS1 10 IMFEXEC CICS ALTER ICV 1000 IMFEXEC CICS ALTER ICVR 5000 IMFEXEC CICS ALTER TCLASS DFHTCL0 MAXACTIVE 200 IMFEXEC CICS ALTER SYSTEM NO IMFEXEC CICS ALTER DUMPS OPENSTATUS OPEN IMFEXEC CICS ALTER TCPIPSERVICE PRINTER STATUS CLOSE IMFEXEC CICS ALTER DUMPCODE SYSTEM AP0001 DAE YES IMFEXEC CICS ALTER JVMPOOL STATUS DISABLED "IMFEXEC CICS ALTER MAXTASK 35" "IMFEXEC CICS ALTER CLASS1 10" “IMFEXEC CICS ALTER ICV 1000” “IMFEXEC CICS ALTER ICVR 5000” “IMFEXEC CICS ALTER TCLASS DFHTCL0 MAXACTIVE 200” “IMFEXEC CICS ALTER SYSTEM NO” “IMFEXEC CICS ALTER DUMPS OPENSTATUS OPEN” “IMFEXEC CICS ALTER TCPIPSERVICE PRINTER STATUS CLOSE” “IMFEXEC CICS ALTER DUMPCODE SYSTEM AP0001 DAE YES” “IMFEXEC CICS ALTER JVMPOOL STATUS DISABLED”
Syntax:	**CICS ALTERVS** address [FROM] value [TO] value
Function:	Changes contents of virtual memory within CICS.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS ALTERVS 00031F14 FROM 01080000 TO 00000000 "IMFEXEC CICS ALTERVS 00031F14 FROM 01080000 TO 00000000"
Syntax:	**CICS CEMT** command
Function:	Allows a CEMT command to be issued within the CICS region.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS CEMT SET DUMP SWI "IMFEXEC CICS CEMT SET DUMP SWI"

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 4 of 26)

Syntax:	**CICS CHAP** taskno new-priority
Function:	Alters the dispatching priority of the task specified. The task number can be obtained using MAINVIEW for CICS inquiries.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS CHAP &TASKID 232 "IMFEXEC CICS CHAP" taskid" 232"
Syntax:	**CICS CICSKEY** TranID [YES | NO]
Function:	Changes CICSKEY settings for CICS transactions.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS CICSKEY CEMT YES "IMFEXEC CICS CICSKEY CEMT YES"
Syntax:	**CICS CLOSE** filename1 filename2 ...
Function	Closes one or more CICS files. The CICS files must be defined within the CICS File Control Table (FCT).
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS CLOSE MASTER "IMFEXEC CICS CLOSE MASTER"
Syntax:	**CICS CONN** SYSID IN OUT ACQ REL NOTPEND PURGE
Function:	Is the CICS sysid for the MRO/ISC connections.
Note:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS CONN sysid IN "IMFEXEC CICS CONN sysid IN"
Syntax:	**CICS DISABLE** [FILE | TRAN | PROGRAM | DEST] ID
Function:	Marks a CICS resource unavailable to applications or users except those who are currently using them.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS DISABLE TRAN ABWR IMFEXEC CICS DISABLE PROGRAM PAYMASTR "IMFEXEC CICS DISABLE TRAN ABWR" "IMFEXEC CICS DISABLE PROGRAM PAYMASTR"
Syntax:	**CICS DROP** program
Function:	Causes the use count of the specified program to be decreased by one. Once the use count reaches zero, it is eligible to be removed from storage by CICS.

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 5 of 26)

Cond Codes:	See “CICS common information” for condition codes.
Note:	The above is not true for permanently loaded programs.
Example:	IMFEXEC CICS DROP PAYMAST "IMFEXEC CICS DROP PAYMAST"
Syntax:	**CICS DUMPDB** database
Function:	Prepares a database for dumping by preventing updates so backup jobs can be run in another region. See “CICS common information” for condition codes.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS DUMPDB STDCX2P "IMFEXEC CICS DUMPDB STDCX2P"
Syntax:	**CICS ENABLE** [FILE | TRAN | PROGRAM | DEST] ID
Function:	Marks a CICS resource available to applications or users.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS ENABLE TRAN ABWR IMFEXEC CICS ENABLE PROGRAM PAYMASTR "IMFEXEC CICS ENABLE TRAN ABWR" "IMFEXEC CICS ENABLE PROGRAM PAYMASTR"
Syntax:	**CICS FREE** filename [LOCAL]
Function:	Deallocates a file from the CICS or BBI-SS region. The file must already be closed and disabled before a FREE can be issued.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS FREE MASTER "IMFEXEC CICS FREE MASTER"
Syntax:	**CICS INSERVE** [TERIMINAL | LINE | CONTROLLER]ID
Function:	Places a terminal, line, or controller back in service. Wildcard characters may also appear in termid.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS INSERV TERMINAL &TERMID "IMFEXEC CICS INSERV TERMINAL" termid
Syntax:	**CICS ISOLATE** TranID [YES | NO]
Function:	Changes ISOLATE settings for CICS transactions.
Cond Codes:	See “CICS common information” for condition codes.

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 6 of 26)

Example:	IMFEXEC CICS ISOLATE CEMT YES "IMFEXEC CICS ISOLATE CEMT YES"
Syntax:	**CICS KILL TASK** taskno [DUMP | NODUMP | FORCE | PURGE | FORCEPURGE]
Function:	Terminates a CICS task using the task number.
Cond Codes:	KILL TASK has limited function under CICS Version See “CICS common information” for condition codes.
Example:	IMFEXEC CICS KILL TASK &TASKNO "IMFEXEC CICS KILL TASK" taskno
Syntax:	**CICS KILL TERM** terminal-id Type
Function:	Terminates a CICS task associated with a terminal.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS KILL TERM BSA4 "IMFEXEC CICS KILL TERM BSA4"
Syntax:	**CICS LOAD** program
Function:	Increases the use-count for the specified program by one. If the program is not in storage, it is loaded first. Use DROP to decrease the use-count value.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS LOAD PAYMASTR "IMFEXEC CICS LOAD PAYMASTR"
Syntax:	**CICS NEWCOPY** program
Function:	Marks the program specified as nonresident and refreshes its disk address. Next time the program is loaded, it will use the new disk address, thereby using the new copy.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS NEWCOPY PGM1 "IMFEXEC CICS NEWCOPY PGM1"
Syntax:	**CICS OPEN** filename
Function:	Opens a file in the CICS region. The file must first be allocated.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS OPEN MAIN1 "IMFEXEC CICS OPEN MAIN1"
Syntax:	**CICS OUTSERVE** [TERMINAL | LINE | CONTROLLER]ID

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 7 of 26)

Function:	Takes the terminal, line, or control unit out of CICS service. Wildcard characters may also appear in termid.
Cond Codes:	See “CICS common information” for condition codes.
.Example:	IMFEXEC CICS OUTSERVE TERMINAL &TERMID "IMFEXEC CICS OUTSERVE TERMINAL" termid
Syntax:	**CICS PURGE** [TSUT | ICE | DEST | AID]ID
Function:	Terminates a CICS resource.
Cond Codes:	See “CICS common information” for condition codes.
Note:	If an ICE or TSUT contains special characters, use the full (up to 32 bytes) hexadecimal identifier.
Example:	IMFEXEC CICS PURGE TSUT &TSUTID IMFEXEC CICS PURGE TSUT 1C3A773B HEX IMFEXEC CICS PURGE ICE 22222222 HEX IMFEXEC CICS PURGE DEST DEVL IMFEXEC CICS PURGE AID 0000000000000000 L287 TRN1 HEX "IMFEXEC CICS PURGE TSUT" tsutid “IMFEXEC CICS PURGE TSUT 1C3A773B HEX” “IMFEXEC CICS PURGE ICE 22222222 HEX” “IMFEXEC CICS PURGE DEST DEVL” “IMFEXEC CICS PURGE AID 0000000000000000 L287 TRN1 HEX”
Syntax:	**CICS QUERY** command
Function:	Invokes MAINVIEW for CICS interactive services. Information obtained can be used to control the flow of the EXEC by subjecting it to comparison statements. Up to 24 lines of data can be returned in variables LINE1 through LINE24. The variable IMFNOL is set to the number of lines returned.
Cond Codes:	See “CICS common information” for condition codes.

Example:	IMFEXEC CICS QUERY SUBPOOL
	/* DSA PERCENTAGE IS NOW IN MSG #2, COLUMNS 14-16 */
	MFEXEC VGET LINE2 LOCAL
	  SET &DSAPERC = &SUBSTR(14:16,&LINE2)
	  IF &DSAPERC < 80 THEN GOTO LOKAY
	
	"IMFEXEC CICS QUERY SUBPOOL"
	/* DSA usage is in 2nd line of message.*/
	"IMFEXEC VGET LINE2 LOCAL"
	IF SUBSTR(line2,14,2) < '80' THEN NOP
	  ELSE error routine.

Syntax:	**CICS RECOVERDB** database

Table 4-1 IMFEXEC Command Syntax Summary (Part 8 of 26)

Function:	Prepares a database for recovery by preventing reads and updates, allowing a recovery utility to execute in another region.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS RECOVERDB &DBID "IMFEXEC CICS RECOVERDB" dbid
Syntax:	**CICS RELEASE TERMINAL** terminal-id
Function:	Releases control of a CICS terminal back to VTAM. The terminal identifier may be masked.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS RELEASE TERMINAL &TERMID "IMFEXEC CICS RELEASE TERMINAL" termid
Syntax:	**CICS SPURGE** tranid [YES | NO]
Function:	Alters the SPURGE value for a CICS transaction
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS SPURGE PAY1 YES "IMFEXEC CICS SPURGE PAY1 YES"
Syntax:	**CICS STARTDB** database
Function:	Activates a database, making it available for processing.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS STARTDB &DBID "IMFEXEC CICS STARTDB" dbid
Syntax:	**CICS STOPDB** database
Function:	Deactivates a database, making it unavailable for processing.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICS STOPDB &DBID "IMFEXEC CICS STOPDB" dbid
Syntax:	**CICSTRAN** transaction-id ‘parameters...’
Function:	Initiates a CICS transaction on the current CICS TARGET system.
Cond Codes:	See “CICS common information” for condition codes.
Example:	IMFEXEC CICSTRAN FST2 ‘QOFF’ “IMFEXEC CICSTRAN FST2 ‘QOFF’”
Syntax:	**CMD** .commandname,parameter

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 9 of 26)

Function:	Issues a BBI-SS command without response.	
Cond Codes:	**0**	Always returns this condition code.
Example:	IMFEXEC CMD .START LINK,&LINKID “IMFEXEC CMD .START LINK,” linkid	
Syntax:	**CMD** '.commandname,parameter' TYPE(BBI) [ALL] [ALLWAIT(1 - 9999)]	
Function:	Issues a BBI Control Command with response; response is returned to the issuing EXEC.	
Cond Codes:	**0**	Command response returned before WAIT time expired
	4	Command response partially returned before WAIT time expired
	8	No reply received
Example:	IMFEXEC CMD '.D A' TYPE(BBI) ALL "IMFEXEC CMD '.D A' TYPE(BBI) ALL"	
Syntax:	**CMD** '#command' [RESPONSE(* | message-id)] [COUNT(1 | n)] [WAIT(30 | n)] [ALL] [ALLWAIT(1 - 9999)] [MIGID(yes | no)] [DEBUG]	
Function:	Issues an MVS command with response.	
Syntax:	**CMD** '#command' COUNT(0) TYPE (MVS)	
Function:	Issues an MVS command without response.	
Note:	Response from the command is placed in variables LINE1 through LINExx, where xx is the last line. The total number of lines returned is placed in variable IMFNOL.	
Cond Codes:	**0**	Command completed responding before WAIT time expired (functioned successfully).
	4	Command was partially complete after WAIT time expired. Response is incomplete.
	8	No reply was received within the WAIT time specified.
	16	Command text is greater than 126 characters (MVS/SP Version 4 only).
	20	Severe error; see short message text for more information.
Example:	IMFEXEC CMD '$DA' RESPONSE($HASP605) COUNT(50) "IMFEXEC CMD '$DA' RESPONSE($HASP605) COUNT(50)"	
Syntax:	**CMD** ’/IMS-command’	
Function:	Issues an IMS command without response.	
Note:	Do not enclose the command in quotation marks if minimal response is required.	

Table 4-1 IMFEXEC Command Syntax Summary (Part 10 of 26)

Cond Codes:	**0**	Command issued and first segment of response is in &SYSDVAL.
	4	Generic command format resulted in multiple IMS commands. The variable &SYSDVAL contains first segment of response from the first command.
	8	Command timed out. No response is available.
	12	Command was not issued. Either the target IMS was not available, or no resource matched the generic mask.
Example:	IMFEXEC CMD /STA TRAN TE4C0CNG "IMFEXEC CMD /STA TRAN TE4C0CNG"	
Syntax:	**CMD** '/IMS-command' [COUNT(number | 1) TYPE(IMS)] DBCTL(dbctltgt) [WAIT(30 | n)] [ALL] [ALLWAIT(1 - 9999)]	
Function:	Issues an IMS command with response.	
Note:	Response from the command is placed in variables LINE1 through LINExx, where xx is the last line. The total number of lines returned is placed in variable IMFNOL.	
Cond Codes:	**0**	Command completed responding before WAIT time expired (functioned successfully).
	4	Command was partially complete after WAIT time expired. Response is incomplete.
	8	No reply was received within the WAIT time specified.
	12	Target IMS was not active.
	20	Command response manager is not active.
Example:	IMFEXEC CMD '/DIS TRAN TH*' COUNT(20) TYPE(IMS) "IMFEXEC CMD '/DIS TRAN TH*' COUNT(20) TYPE(IMS)"	
Syntax:	**CNTL** [CMD | NOCMD] [LIST | NOLIST] [PERLIM(number)] [TIMLIM(number)] [SELLIM()] [MAXTPUT()] [GLOBAL | LOCAL]	
Function:	Specifies control options for the currently executing EXEC.	
Note:	Use this command to facilitate EXEC debugging. PERLIM maximum value is 99; TIMLIM maximum value is 9999.	
Cond Codes:	**0**	Command completed successfully.
	8	Invalid command syntax.
Example:	IMFEXEC CNTL LIST NOCMD "IMFEXEC LIST NOCMD"	
Syntax:	**DOM ID**(domid)	
Function:	Deletes a non-scrollable WTO or a WTOR.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 11 of 26)

Cond Codes:	**0**	Message successfully deleted.
	8	DOM ID is missing.
Example:	IMFEXEC DOM ID(&IMFDOMID) "IMFEXEC DOM ID("imfdomid")"	
Syntax:	**EXIT CODE**(number)	
Function:	Sets the return code IMFRC.	
Note:	An EXIT statement must be coded after each IMFEXEC EXIT command for REXX EXECs. The return code will be passed either to the caller of IMFSUBEX or in IMFRC of the initiating EXEC.	
Cond Codes:	**0**	Always zero.
Example:	IMFEXEC EXIT CODE(&code) "IMFEXEC EXIT CODE("code")"	
Syntax:	**HB** [INTERVAL(30 | number)]	
Function:	Changes the interval, which MAINVIEW AutoOPERATOR and Elan use to exchange heartbeats. The number is an interval in seconds.	
Cond Codes:	**0**	Interval successfully reset.
	8	Interval is either missing or invalid.
Example:	IMFEXEC HB INTERVAL(&interval) "IMFEXEC INTERVAL("interval")"	
Syntax:	**IMFC** command [options] [TARGET=name] [IMAGE=YES | NO] [USERID(name)] [SCROLL=YES | NO]	
Function:	Issues an IMF or MAINVIEW for DB2 service to invoke an analyzer display, to start and stop monitors, to invoke a monitor display, or to invoke a display for automatic image logging.	
Note:	The maximum length of an IMFC command is 68, except for the SET command. Variables LINE4 through LINE43 are returned with the screen image from the service. IMFNOL is not set.	
Cond Codes:	**0**	Command executed successfully.
	8	Target missing, error in an analyzer or monitor, or syntax.
	16	Handling program was not found.
Example:	IMFC STAT TARGET=&QIMSNAME IMAGE=NO "IMFC STAT TARGET="qimsname" IMAGE=NO"	
Syntax:	**IMFC SET PRG=CALLX | ALL** EXECname[USRID=userid] [TARGET=target name]	
Function:	Uses SET PRG=CALLX | ALL to terminate a time-initiated EXEC.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 12 of 26)

Cond Codes:	**0**	Command executed successfully.
	8	One of the following conditions: • TARGET= is missing. • An error in the monitor or analyzer service occurred (Msg PM0334E issued). • Syntax error found in SET command (Msg PM0337E issued).
	16	Handling program was not found.
Example:	IMFEXEC IMFC SET PRG=CALLX IMSCHECK "IMFEXEC IMFC SET PRG=CALLX IMSCHECK"	
Syntax:	**IMFC SET REQ=CALLX** exec-name [TARGET=ssid] [START=hh:mm:ss] [STOP=hh:mm:ss] [STOPCNT=( )] [I=00:01:00 | hh:mm:ss] [USRID=userid] TARGET=target name	
Function:	Defines time-initiated requests from an EXEC.	
Note:	STOP and START times can be specified with I (interval) to have an EXEC start at a specified time of day and then have the EXEC reissued at specified intervals until the STOP time you specify. Use STOPCNT to specify the number of executions of the EXEC.	
Cond Codes:	**0**	Command executed successfully.
	8	Target missing, error in an analyzer or monitor, or syntax.
	16	Handling program was not found.
Example:	IMFEXEC IMFC SET REQ=CALLX @HOURLY START=6:00:00 STOP=16:00:00 + I=01:00:00 TARGET=&IMFORGSS USERID=JDB1 "IMFEXEC IMFC SET REQ=CALLX @HOURLY START=6:00:00 STOP=16:00:00" , "I=01:00:00 TARGET="imforgss" USERID=JDB1"	
Syntax:	**IMSTRAN** trancode ['operands']	
Function:	Initiates an IMS transaction.	
Note:	The maximum length of the operands and transaction code is 255 characters. All transactions to be initiated in this fashion must be predefined in the AAOTRN00 member of BBPARM. If the transaction requires no operands, you must minimally code ' ' as an operand.	
Cond Codes:	**0**	Command executed successfully.
	12	Error occurred. A message will be logged to the BBI-SS journal.
Example:	IMFEXEC IMSTRAN ADDPART 'ab960C10,RIVET,74' "IMFEXEC IMSTRAN ADDPART 'ab960C10,RIVET,74'"	
Syntax:	'**JES3** command'	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 13 of 26)

Function:	Issues a JES3 command through the subsystem interface.	
Note:	The maximum length is 127 bytes. The JES3 command character specified in BBPARM member BBISSP00 is automatically appended to the front of the command. Refer to the <HP1 italic>MAINVIEW Common Customization Guide for details.	
Cond Codes:	0	Command was executed successfully
	8	Invalid syntax used
Example:	IMFEXEC JES3CMD I,J=PROD1234 "IMFEXEC JES3CMD I,J=PROD1234"	
Syntax	**JESALLOC** DDNAME [CLASS] [SYSOUT]	
Function:	Allocate a SYSOUT data set to the given DD name.	
Note:	Unlike the TSO ALLOCATE command, this command may be used to allocate a subsystem data set even when a JES connect was performed (such as, JES was started after the AutoOPERATOR subsystem).	
Cond Codes:	0	Command successfully executed
	4	JES rejected the allocation request
	8	Not connected to JES (started under MSTR and no JESCNCT card in bbissp00)
	16	Syntax error
	20	DD name in use (use TSO free command first)
Example:	IMFEXEC JESALLOC MYPRINT SYSOUT CLASS(A) “IMFEXEC JESALLOC MYPRINT SYSOUT CLASS(A)”	
Syntax	**JESSUBM** DSNAME | DS | DA | DSN STEM	
Function:	Submits a JOB from a DD name or stem variables.	
Note:	Unlike the TSO SUBMIT command, this command may be used to allocate a subsystem data set even when a JES connect was performed (such as, JES was started after the AutoOPERATOR subsystem).	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 14 of 26)

Cond Codes:	0	Command successfully executed
	4	JES rejected the allocation request
	8	INTRDR cannot be dynamically allocated. This error can happen if JES has not yet started.
	16	Syntax error occurred.
	20	Error processing input variables.
	24	Not connected to JES (started under MSTR and no JESCNCT card is in BBISSP00).
	28	Invalid input data set or error writing to INTRDR.
	12	Specified data set cannot be allocated or opened or data set LRECL is less than or greater than 80, or data set name is too long.
Example:	IMFEXEC JESSUBM DA(’BAORAE.JCL.CNTL(IEFBR14)’) “IMFEXEC JESSUBM DA(’BAORAE.JCL.CNTL(IEFBR14)’)”	
Syntax	**LOGOFF** [SESSION(sessionid)] [DISCONNECT]	
Function:	Terminates an established OSPI session between the EXEC and a VTAM application.	
Note:	If DISCONNECT is not specified, LOGOFF issues a VTAM TERMSESS macro against the application and closes the VTAM ACB that is used to communicate with the application. This action results in an unconditional LOGOFF from the application. All internal resources associated with this session are then freed. DISCONNECT leaves the OSPI session intact but releases control of the OSPI session from the EXEC.	
Cond Codes:	**0**	Command executed successfully.
	8	Syntax error or the session specified is not valid.
Example:	IMFEXEC LOGOFF SESSION(&TSOSESS) "IMFEXEC LOGOFF SESSION("tsosess")"	
Syntax:	**LOGON** [APPLID(name) | ACB(name] [DATA(text)] [USERDATA(text)] [PREFIX(prefix | OSI)] [SESSION(sessionid)] [REQACB(name)] [LOGMODE(modename | D6327802)] [DEBUG | NODEBUG] [NORECEIVE]	
Function:	Establishes an initial session between an EXEC and any VTAM application.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 15 of 26)

Note:	LOGON supplies the first screen output to the EXEC. You can have as many sessions simultaneously active as you have active OSPI ACBs in SYS1.VTAMLST. The SESSION parameter is required only if reconnecting back to a session. Use the NORECEIVE keyword when logging on to an application that does not display an initial panel.	
Cond Codes:	**0**	LOGON was successful.
	4	LOGON SESSION (reconnect) issued but failed to reestablish the session.
	8	LOGON failed.
Example:	IMFEXEC LOGON APPLID(TSO) DATA(SYSUSER) LOGMODE(D6327803) PREFIX(TSO) "IMFEXEC LOGON APPLID(TSO) DATA(SYSUSER) LOGMODE(D6327803) PREFIX(TSO)"	
Syntax:	**MSG** ’msg-text’ [TARGET(name | LOCAL)]	
Function:	Logs a message to the BBI-SS journal specified within the TARGET field.	
Note:	Maximum length of msg-text is 252 bytes.	
Cond Codes:	**0**	Message logged successfully.
	8	NODE not found when using TARGET (check the BBINOD00 member in BBPARM).
	12	TARGET is not found (check the BBIJNT00 member).
	16	NODE is not available.
Example:	IMFEXEC MSG ’Manufacturing Database is Off-line’ TARGET(&TGTNAME) "IMFEXEC MSG ’Manufacturing Database is Off-Line’ TARGET("tgtname")"	
Syntax:	**NOTIFY** [NAME(name)] [INFO(text)]	
Function:	Pages a person’s nameand supplies information text.	
Note:	The person being paged must be defined to Elan. Contents of INFO can be up to 12 characters in length.	
Cond Codes:	**0**	Person was paged.
	8	Request timed out.
	12	Elan could not process the request.
	16	Communications with Elan could not be established.
Example:	IMFEXEC NOTIFY NAME(TAT2) INFO(CICSP DOWN) "IMFEXEC NOTIFY NAME(TAT2) INFO(CICSP DOWN)"	
Syntax:	**POST** [name] [CODE(code)] [TARGET(targetname)]	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 16 of 26)

Function:	Notifies an EXEC that has issued the IMFEXEC WAIT command that it can continue execution.	
Note:	This statement is used in conjunction with the IMFEXEC WAIT command statement. Refer to the WAIT command on page 4-26 for more information.	
Cond Codes:	**0**	Name successfully posted.
	4	No waiting EXEC was found.
	8	Node was not found.
	12	Target was not available.
	16	Node was not available.
Example:	IMFEXEC POST TEST CODE(ABC) TARGET(SYSA) "IMFEXEC POST TEST CODE(ABC) TARGET(SYSA)"	
Syntax:	**RECEIVE** [SESSION(sessionid)] [TIMEOUT(nn)]	
Function:	Attempts to receive a packet of data from the application.	
Note:	This command should be used only with VTAM applications that do not follow standard protocol.	
Cond Codes:	**0**	Data was received.
	4	Request timed out.
Example:	IMFEXEC RECEIVE SESSION(&CNMSESS) "IMFEXEC RECEIVE SESSION("cnmsess")"	
Syntax:	**RES** command WAIT (60 | nnnn)	
Function:	Issues SYSPROG commands.	
Note:	Responses are returned into variables from LINE1 to LINE*nn*, where *nn* is the last line. IMFNOL is set to the number of lines returned.	
Cond Codes:	**0**	Command was successful
	8	No SYSPROG service parameter was passed.
	12	Command timed out (60 seconds).
Example:	IMFEXEC RES ASM IMFEXEC VDCL ASML1 LIST(W1 W2 W3 W4 f IPLTYPE W6 W7) IMFEXEC VGET LINE1 INTO(ASML1) LOCAL IMFEXEC VPUT IPLTYPE IMFEXEC MSG ’An IPL &IPLTYPE start was performed at &SYSTIME.’ "IMFEXEC RES ASM" "IMFEXEC VGET LINE1 LOCAL" "IMFEXEC MSG ’An IPL "subword(line1,5)" on "date()" at "time()" '"	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 17 of 26)

Syntax:	**SCAN** [SESSION(sessionid)] [ROW(row | 1)] [COL(column |1)] [TEXT(text)] [VARIABLE(var | OSIVAR)] [LENGTH(number)] [CASE | NOCASE] [TRIM | NOTRIM] [POSITION]	
Function:	SCAN performs these functions: • Finds a specific character string. • Positions the cursor in the first input field following a specified character string or screen location. • Retrieves data from the screen buffer into user-defined variables.	
Note:	If VARIABLE is not specified, POSITION is then required. VARIABLE and LENGTH are mutually inclusive. Do not specify a leading ampersand (&) in the VARIABLE operand.	
Cond Codes:	**0**	Text found.
	4	No text was found.
	8	Syntax error, conflicting parameters specified, or invalid session occurred.
Example:	IMFEXEC SCAN TEXT(USERID) VAR(MYUSER) LENGTH(25) TRIM NOCASE SESSION(&TSOSESS) "IMFEXEC SCAN TEXT(USERID) VAR(MYUSER) LENGTH(25) TRIM NOCASE SESSION("tsosess")"	
Syntax:	**SELECT** PGM(name) [PARM(p1 ... pn)]	
Function:	Invokes a user-written program.	
Note:	All user-written programs should begin with IMFU to prevent name conflicts with other MAINVIEW AutoOPERATOR programs.	
Cond Codes:	**0**	Program called successfully. IMFRC contains return code from called program.
	8	Either the program is not found or the program name is invalid.
	12	Program name does not begin with IMFU.
Example:	IMFEXEC SELECT PGM(IMFU100) PARM(&TERMID &SYSDATE &SYSTIME) "IMFEXEC SELECT PGM(IMFU00) PARM("termid" "date" "time")"	
Syntax:	**SELECT** EXEC(name [p1 ... pn]) [WAIT(NO | YES)] [TARGET(name | LOCAL)] [PRI(HIGH | NORMAL)]	
Function:	Schedules an EXEC either on the current MAINVIEW AutoOPERATOR or on another MAINVIEW AutoOPERATOR.	
Note:	WAIT(YES) and TARGET are mutually exclusive. The target-name must be specified in the BBIJNT00 member of BBPARM. If WAIT(YES) is specified, the variable IMFRC is set with the last IMFEXEC EXIT CODE issued by the called EXEC.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 18 of 26)

Cond Codes:	**0**	EXEC scheduled successfully.
	8	TARGET was not found in BBIJNT00; EXEC was not found in the SYSPROC DD-statement libraries; or invalid syntax was specified.
	12	EXEC name length is greater than 8.
	16	EXEC tried to initiate itself (recursion).
Example:	IMFEXEC SELECT EXEC(ALERT IMSPROD DOWN) TARGET(IMFM) WAIT(NO) PRI(HIGH) "IMFEXEC SELECT EXEC(ALERT IMSPROD DOWN) TARGET(IMFM) WAIT(NO) PRI(HIGH)"	
Syntax:	**SEND** 'msg text' [LTERM(xxx)] [USER(name)]	
Function:	Sends a message to either an IMS user (LTERM keyword) or a TSO user (USER keyword).	
Note:	LTERM and USER are mutually exclusive keywords. Maximum length for LTERM messages is 252, and for USER messages is 120.	
Cond Codes:	**0**	Message was sent.
	8	Message exceeds maximum length.
	12	No destination was specified.
	16	Invalid syntax.
Example:	IMFEXEC SEND 'This is a test message' USER(&USER) "IMFEXEC SEND 'This is a test message' USER("user")"	
Syntax:	**SESSINF** SESSION(sessionid)	
Function:	Writes all information relevant to the session to the OSPISNAP file.	
Cond Codes:	**0**	Command was successful.
	8	Invalid command syntax, or session ID was invalid or not found.
Example:	IMFEXEC SESSINF SESSION(&IMSSESS) "IMFEXEC SESSINF SESSION("imssess")"	
Syntax:	**SETTGT** target name	
Function:	Sets the target CICS system ID to the specified value. The &IMFSYSID variable is also altered accordingly.	
Note:	The target specified also must be specified within the BBIJNT00 member in BBPARM. All subsequent CICS commands will go to this target. This command is used for CICS regions only.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 19 of 26)

Cond Codes:	**0**	Target was altered.
	8	Target was not found in BBIJNT00.
Example:	IMFEXEC SETTGT 'CICSPROD' "IMFEXEC SETTGT 'CICSPROD'"	
Syntax:	**SHARE** Variable name | (var1 var2...var*n*)	
Function:	Returns the specified local variables to the AOEXEC caller EXEC. Issued by AOEXEC called EXEC.	
Cond Codes:	**12**	This EXEC is not running under AOEXEC control; therefore, the IMFEXEC SHARE statement is inapplicable.
	16	Syntax error.
Example:	IMFEXEC SHARE (LINE*) "IMFEXEC SHARE (LINE*)"	
Syntax:	**STDTIME** var1 var2 var3 var4	
Function:	Instructs Elan to obtain the Greenwich date and time and local date and time. Greenwich date, Greenwich time, local date, and local time are returned in variables var1 through var4 respectively.	
Cond Codes:	**0**	GMT time was obtained.
	4	Variable name was invalid.
	8	Request timed out.
	12	Elan could not execute request.
	16	Communications with the Elan workstation could not be established
Example:	IMFEXEC STDTIME V1 V2 LDATE LCLOCK IMFEXEC CMD #SET DATE=&LDATE,CLOCK=&LCLOCK "IMFEXEC STDTIME V1 V2 LDATE LCLOCK" "IMFEXEC CMD #SET DATE="ldate",CLOCK="lclock	
Syntax:	**SUBMIT** data-set-name	
Function:	Submits jobs to be executed.	
Notes:	SUBMIT does not function if AO is started BEFORE JES. The following keywords are NOT supported: HOLD, NOHOLD, NOTIFY, NONOTIFY, PASSWORD, NOPASSWORD, USER( ), NOUSER. All submit exits are honored.	
Cond Codes:	**0**	Job submitted.
	16	Error; job was not submitted.

Table 4-1 IMFEXEC Command Syntax Summary (Part 20 of 26)

Example:	IMFEXEC SUBMIT ’TAT2.PRIVATE.CNTL(&MEMBER)’ "IMFEXEC SUBMIT ’TAT2.PRIVATE.CNTL("member")'"	
Syntax:	**TAILOR** DD MEMIN() | TEMPIN MENOUT() | TEMPOUT STEMIN() | TEMP INCLUDE() | TEMPIN [SEARCH()] | TSO STEMOUT() | TEMPTLR [DEBUG] | NODEBUG	
Function:	Reads a member of a partitioned data set, substitutes various variables, processes tailoring-specific directives and issues a TSO SUBMIT to the generated results.	
Cond Codes:	0	Success completion
	4	An error reading input for tailoring from variables or a PDS member. Messages indicate the specific condition.
	8	A catastrophic error processing the input
	12	An error writing output to variables or a PDS member
	16	A syntax error parsing parameters, such as mutually exclusive or inclusive
Example:	IMFEXEC TAILOR STEMIN(INPUT) STEMOUT(OUTPUT) SEARCH(‘’) “IMFEXEC TAILOR DD(AOJCL) MEMIN(RECOVER) SEARCH(TSO LOCAL)STEMOUT(TEMPTLR)”	
Syntax:	**TRANSMIT** [SESSION(sessionid)] [ENTER | CLEAR | PFn | PAn]	
Function:	Transmits the modified virtual screen buffer back to the application for processing. This action is equivalent to an operator pressing ENTER, CLEAR, PFn, PAn.	
Cond Codes:	**0**	Buffer transmitted.
	4	Time out occurred.
	8	Invalid session ID specified.
Example:	IMFEXEC TRANSMIT SESSION(&TSOSESS) ENTER "IMFEXEC TRANSMIT SESSION("tsosess") ENTER"	
Syntax:	**TYPE** [TEXT(text)] [SESSION(sessionid)] [TAB | BACKTAB | ERASEEOF | HOME | RESET] [ROW(row)] [COL(column)]	
Function:	Allows data entry into any VTAM buffer image. With this command, you can specify any keystroke that does not generate a host interrupt.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 21 of 26)

Note:	ENTER is an alias name for TYPE, but TYPE is preferred. Maximum value for row is 43, for column is 80, and for text is 2.	
Cond Codes:	**0**	Buffer updated.
	4	Time out occurred.
	8	Session ID is invalid or not found.
Example:	IMFEXEC TYPE TAB TEXT(’catalog’) SESSION(&TSOSESS) "IMFEXEC TYPE TAB TEXT(’catalog’) SESSION("tsosess")"	
Syntax:	**VCKP**	
Function:	Provides checkpoint processing of PROFILE variables. All updated PROFILE variables are physically written out to the BBIVARS data set by this command.	
Note:	Normally, all PROFILE variables updated by an EXEC are copied to the BBIVARS data set when the EXEC ends. BMC Software recommends using this command in EXECs that run for extended periods of time.	
Cond Codes:	**0**	PROFILE variables successfully written to BBIVARS data set.
	8	Invalid command syntax.
Example:	IMFEXEC VCKP "IMFEXEC VCKP"	
Syntax:	**VDCL** variable LIST(var1 ... varn)	
Function:	Equates a global variable to a list of local EXEC variables so they can be processed together.	
Note:	The global variable is parsed into the LOCAL variables using a VGET with the INTO keyword. The parsing delimiter is a space.	
Cond Codes:	**0**	Command was successful.
	12	Invalid command syntax
Example:	IMFEXEC VDCL IMFL20 LIST(W1 W2 W3 ... W12) "IMFEXEC VDCL IMFL20 LIST(W1 W2 W3 ... W12)"	
Syntax:	**VDEL** variable | (v1 v2 ... vn)] [ALL | LOCAL | SHARED | PROFILE [TARGET(target-name)]	
Function:	Deletes variables from a particular pool.	
Cond Codes:	**0**	Command was successful.
	8	Variable was not found.
	20	Severe internal error occurred.

Table 4-1 IMFEXEC Command Syntax Summary (Part 22 of 26)

Example:	IMFEXEC VDEL ABC* "IMFEXEC VDEL ABC*"	
Syntax:	**VDELL** variable | pattern | (v1 ... vn) [LOCAL | SHARED | PROFILE)]	
Function:	Deletes one or more long variables from one of the AutoOPERATOR variable pools.	
Cond Codes:	**0**	The variable existed in the target pool and has been deleted.
	8	No long variable with this name has been found in the target pool.
	12	An attempt to delete a read-only variable (for example, Q-type variable was specified which cannot be deleted with VDELL).
	16	Syntax error occurred.
	20	Variable pool was not found (BBIVARS not allocated).
Example:	IMFEXEC VDELL (DSN*) SHARED "IMFEXEC VDELL (DSN*) SHARED"	
Syntax:	**VDEQ** name	
Function:	Performs an MVS dequeue using the minor name specified and a major name of BBIUSER.	
Cond Codes:	**0**	Resource dequeued.
	8	Either no minor name was passed or the minor name length exceeded 16 characters.
Example:	IMFEXEC VDEQ '&MINOR' "IMFEXEC VDEQ"'minor'""	
Syntax:	**VENQ** name [SHR | EXC] [TEST]	
Function:	Perform an MVS enqueue using the minor name specified and a major name of BBIUSER.	
Note:	Maximum length of the minor enqueue name is 255 alphanumeric characters. Use the TEST keyword to specify that no ENQ is obtained but the availability of the ENQ is tested.	
Cond Codes:	**0**	EXEC did not have control of the resource but could have obtained it if it was requested at the time of the command.
	4	Another task has control of the resource.
	8	The EXEC already has control of the resource
Example:	IMFEXEC VENQ '&MINOR' SHR "IMFEXEC VENQ "'minor"' SHR"	
Syntax:	**VGET** (list) [INTO(variable)] [LOCAL | SHARED | PROFILE] [DECRYPT(xyz)] [DELIM(',')] [TARGET(target-name)]	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 23 of 26)

Function:	Obtains one or more variables from a particular pool.	
Note:	Use the DECRYPT keyword only in conjunction with the ENCRYPT keyword on the IMFEXEC VPUT statement. Use it to decrypt the data (xyz) that is encrypted by IMFEXEC VPUT ENCRYPT(XYZ) statement. Refer to the VPUT command on page 4-25 for more information.	
Cond Codes:	**0**	Variable was obtained.
	8	Variable does not exist.
	12	Severe internal error occurred.
Example:	IMFEXEC VGET (TERMID) LOCAL IMFEXEC VDCL CICSL20 LIST(W1 W2 W3 W4 W5 W6) IMFEXEC VGET LINE20 INTO(CICSL20) LOCAL	
Syntax:	**VGETL** Variable name | (v1 ... vn) [LOCAL | SHARED | PROFILE)]	
Function:	Copies one or more long variables from one of the AutoOPERATOR pools into the TSO pool.	
Note:	If more than one variable is specified, the variable names must be enclosed in parentheses.Each variable name can be up to 30 characters. The maximum length of the combined variable values is 252 bytes.	
Cond Codes:	**0**	The variable existed in the target pool and has been retrieved.
	8	No long variable with this name has been found in the target pool.
	16	Syntax error occurred.
	20	Variable pool was not found (BBIVARS not allocated).
Example:	IMFEXEC VGETL (DSNTEST) SHARED "IMFEXEC VGETL (DSNTEST) SHARED"	
Syntax:	**VLST** variable-pattern [SHARED | PROFILE] [REXX]	
Function:	Retrieves variable names in the designated pool that match the variable or mask specified.	
Note:	Variable names are returned in local variables LINE1 through LINE*nn*, where *nn* is the last line. IMFNOL contains the total number of lines.	
Cond Codes:	**0**	List created successfully.
	8	No variables was found in pool.
	12	Pool was not available.
Example:	IMFEXEC VLST RETRY* SHARED "IMFEXEC VLST RETRY* SHARED"	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 24 of 26)

Syntax:	**VLSTL** Variable pattern [SHARED | PROFILE]	
Function:	Retrieves long variable names in the designated pool that match the variable or mask specified.	
Note:	This variable operation only supports a subset of the functions available for the short variables. It ONLY affects and searches for long variables. If a short variable (created with VPUT instead of VPUTL) with the specified name exists, it is ignored.	
Cond Codes:	**0**	At least one variable has been found.
	8	No long variable with this name has been found in the target pool.
	16	Syntax error occurred.
	20	Variable pool was not found (BBIVARS not allocated).
Example:	IMFEXEC VLSTL DSN* SHARED "IMFEXEC VLSTL DSN* SHARED"	
Syntax:	**VPUT** (name-list) [FROM(variable)] [LOCAL | SHARED | PROFILE] [USING(v1...vn)] [ENCRYPT(xyz)] [TARGET(targetname)]	
Function:	Stores one or more variables within a particular pool.	
Note:	• Use VCKP in conjunction with VPUT PROFILE to ensure variable integrity. • Use the USING keyword when you want to set the MAINVIEW AutoOPERATOR variables from the LOCAL variable pool. • Use the ENCRYPT keyword only in conjunction with the DECRYPT keyword on the IMFEXEC VGET statement. Use it to encrypt data (xyz) that will be decrypted by IMFEXEC VGET DECRYPT(XYZ). Refer to VGET on page 4-24 for more information.	
Cond Codes:	**0**	Variable was updated successfully.
	4	Variable created. Did not exist previously in pool.
	8	Invalid syntax specified.
	12	Q-type or READ-ONLY variable specified. Operation aborted.
	16	Internal error occurred.
	20	Either no variables are in the list or a variable name is invalid.
Example:	IMFEXEC VPUT (ABENDS) LOCAL "IMFEXEC VPUT (ABENDS) LOCAL"	
Syntax:	**VPUTL** Variable name | (v1 ... vn)[LOCAL | SHARED | PROFILE)]	
Function:	Stores one or more long variables within a particular pool.	

BMC Software, Inc., Confidential and Proprietary Information

Table 4-1 IMFEXEC Command Syntax Summary (Part 25 of 26)

Cond Codes:	**0**	The variable existed in the target pool and has been overwritten.
	4	The variable did not exist in the pool and has been created.
	8	Error occurred during operation. Possible Out-of-Space condition for the PROFILE pool.
	12	An attempt was made to set a read-only variable (for example, Q-type variable was specified which cannot be VPUT).
	16	Syntax error occurred.
	20	Variable pool was not found. BBIVARS not allocated.
Example:	IMFEXEC VPUTL (DSNTEST) SHARED "IMFEXEC VPUTL (DSNTEST) SHARED"	
Syntax:	**WAIT** [seconds] [NAME(name)]	
Function:	Suspends the execution of the EXEC by the number of seconds specified or until a value (name) is posted by another EXEC using the IMFEXEC POST statement.	
Cond Codes:	**0**	Command completed normally.
	8	Invalid number of seconds specified.
	16	Syntax error occurred.
Example:	IMFEXEC WAIT 15 IMFEXEC WAIT 15 NAME(TOKEN) "IMFEXEC WAIT 15" "IMFEXEC WAIT 15 NAME(TOKEN)"	
Syntax:	**WAITLIST** pattern	
Function:	Returns the number of EXECs in a WAIT mode in variables LINE1 through LINE*xx*.	
Note:	The variable IMFNOL contains the number of lines returned.	
Cond Codes:	**0**	The names of one or more waiting EXECs are returned.
	4	No waiting EXECs were found.
	12	Syntax error occurred.
Example:	IMFEXEC WAITLIST "IMFEXEC WAITLIST"	
Syntax:	**WTO** 'msg text' [ROUTCDE(n1 ... nn)] [JOBID(name)] [DESC(n1 ... nn)] [CONSOLE(nn)] [NAME(consolename)] [SSID(YES | NO)]	
Function:	Sends a message to a system console.	
Note:	Variable IMFWTDOM contains the DOM ID of the message written to the console.	

Table 4-1 IMFEXEC Command Syntax Summary (Part 26 of 26)

Cond Codes:	**0**	Message was sent.
	8	Message length exceeded 100 bytes. Operation aborted.
Example:	IMFEXEC WTO 'SHIFT CHANGE AT 6PM' ROUTCDE(1 5 14) DESC(1) "IMFEXEC WTO 'SHIFT CHANGE AT 6PM' ROUTCDE(1 5 14) DESC(1)"	
Syntax:	**WTOR** 'msg text' REPLY(var-name) [JOBID(n)] [ROUTCDE(n1 ... nn)] [DESC(n1 ... nn)] [CONSOLE | CN(n)] [NAME(consolename)] [WAIT(n)] [SSID(YES | NO)]	
Function:	Sends message to system console and waits for an operator reply.	
Cond Codes:	**0**	Command was successful.
	8	A syntax error occurred; the message length was greater than 100 characters; or a time out occurred (the operator did not respond within the WAIT time specified).
Example:	IMFEXEC WTOR 'Network coming up in 1 minute, Reply "S" to STOP' WAIT(&SEC) REPLY(REPLY) "IMFEXEC WTOR 'Network coming up in 1 minute, Reply ""S"" to STOP' WAIT("sec") REPLY(REPLY)"	

BMC Software, Inc., Confidential and Proprietary Information

ACCESS NV Command Syntax Summary

MAINVIEW AutoOPERATOR Access NV provides a two-way conduit of communication between NetView and MAINVIEW AutoOPERATOR. The IMFEXEC NETVIEW command statements link MAINVIEW AutoOPERATOR EXECs to NetView and the NAIEXEC command statements link EXECs from NetView to MAINVIEW AutoOPERATOR. Access NV sets the following two NetView variables after executing the request:

Variables	Function
NAICC	Indicates completion status of the NAIEXEC request.
NAIRC	Indicates completion status of the MAINVIEW AutoOPERATOR request.

BMC Software, Inc., Confidential and Proprietary Information

NAIEXEC Completion Codes

NAICC can have one of the following return codes for all NAIEXEC requests:

Table 5-1 NAIEXEC Completion Codes

Completion Codes	Description
0	Command completed successfully.
4	Command timed out.
8	TARGET is not available or is invalid.
12	Syntax error occurred in NAIEXEC command.
16	Command is incomplete.
20	Invalid or unknown NAIEXEC request.
24	Internal error occurred during NAIEXEC processing.
28	Undetermined error occurred during NAIEXEC processing.
32	Unable to set NetView variables or the NAISVAR CLIST in not available.
36	Unable to establish session.

NAIRC values are described in the following NAIEXEC command descriptions under Condition Codes.

Table 5-2 MAINVIEW AutoOPERATOR Access NV Command Summary (Part 1 of 3)

Syntax:	**NAIEXEC ALERT** alertkey 'alert text' [ALARM(NO | YES)] [COLOR(RED | PINK | YELLOW | LTBLUE | DKBLUE | GREEN | WHITE)] [EXEC('execname p1 p2 ... pn')] [HELP(panelname)] [PCMD('command')] [PRI(CRITICAL | MAJOR | MINOR | WARNING | INFORMATIONAL | CLEARING)] [QUEUE(MAIN | queue)] [TARGET(name)] [ORIGIN(name)] [USER(name)]
Function:	Allows ALERT messages to be generated from NetView.

Table 5-2 MAINVIEW AutoOPERATOR Access NV Command Summary (Part 2 of 3)

Note:	This command is issued from within a NetView EXEC. The ALERT is displayed in the MAINVIEW AutoOPERATOR ALERT application. Multi-line ALERTs can be achieved by embedding a /N to force a new line within the ALERT text. Condition code is returned in the variable NAIRC.
Cond Codes:	See IMFEXEC ALERT command on page 4-2.
Example:	NAIEXEC ALERT &NODE.DOWN ’TERMINAL, &NODE IS DOWN’ PRI(WARNING) QUEUE(NET) ’NAIEXEC ALERT "node"DOWN 'TERMINAL, "node" is down.' PRI(WARNING) QUEUE(NET)"
Syntax:	**NAIEXEC SELECT** EXEC(name [p1 ... pn]) [TARGET(name)] [WAIT(nn)]
Function:	Schedules an EXEC either on the current BBI-SS PAS or on another BBI-SS PAS.
Note:	Schedules an EXEC in any associated MAINVIEW AutoOPERATOR region defined in the Jobs Name Table (BBIJNT00).
Cond Codes:	See “NAIEXEC Completion Codes” on page 5-2
Example:	NAIEXEC SELECT EXEC(ALERT IMSPROD DOWN) TARGET(IMFM) "NAIEXEC SELECT EXEC(ALERT IMSPROD DOWN) TARGET(IMFM)"
Syntax:	**NAIEXEC VDEL** var | (var1 var2 ...) [TARGET(name)]
Function:	Deletes one or more MAINVIEW AutoOPERATOR SHARED variables from a NetView EXEC.
Cond Codes:	See “NAIEXEC Completion Codes” on page 5-2
Example:	NAIEXEC VDEL (OPER1 OPER2) TARGET(PROD) "NAIEXEC VDEL (OPER1 OPER2) TARGET(PROD)"
Syntax:	**NAIEXEC VGET** nvvar [FROM(var)] [TARGET(name)]
Function:	Retrieves the value of a MAINVIEW AutoOPERATOR shared variable and makes it available in the NetView address space. The contents of this variable are placed into one of the OST's global variables.
Cond Code:	See “NAIEXEC Completion Codes” on page 5-2.
Example:	NAIEXEC VGET OPER2 FROM('OPER2') TARGET(PROD) "NAIEXEC VGET OPER2 FROM('OPER2') TARGET(PROD)"
Syntax:	**NAIEXEC VPUT** var-name [FROM(nvvar)] [TARGET(name)]
Function:	Updates the MAINVIEW AutoOPERATOR shared variable with the contents of a NetView variable nvvar.

BMC Software, Inc., Confidential and Proprietary Information

Table 5-2 MAINVIEW AutoOPERATOR Access NV Command Summary (Part 3 of 3)

Cond Code:	See “NAIEXEC Completion Codes” on page 5-2.
Example:	NAIEXEC VPUT (OPER2) FROM('OPER2') TARGET(PROD) "NAIEXEC VPUT (OPER2) FROM('OPER2') TARGET(PROD)"

BMC Software, Inc., Confidential and Proprietary Information

Rule Statement Usage

The following sections list the Selection Criteria and Action Specification fields that you can use when creating Rules.

Selection Criteria Fields

Not every selection criteria field can be used for every event type. The first column shows which event types can use a specific selection criteria field. Depending on which event type you create a Rule for, a panel for that event type appears and will contain a subset of the criteria listed in Table 6-1.

The following table lists all of the available selection criteria fields for all the event types. Note that Time-initiated Rules do not use any of the selection criteria listed in this table.

BMC Software, Inc., Confidential and Proprietary Information

Table 6-1 Selection Criteria Fields (Part 1 of 2)

Event Type	Selection Criteria Field
All event types	Text ID Text String
ALRM	Alarm ID Context Priority Queue Scope Type User ID
ALRT	Key Queue
CMD	Acct Info Console ID Console Name Jobclass Job name RACF Group RACF User Type
CICS	CICS TDQ
DB2	Job name
EXT	Acct Info Jobclass Job name RACF Group RACF User Type
IMS	Acct Info Jobclass Job name RACF Group RACF User Type

BMC Software, Inc., Confidential and Proprietary Information

Table 6-1 Selection Criteria Fields (Part 2 of 2)

Event Type	Selection Criteria Field
JES3	Acct Info Console ID Console Name Desc codes Jobclass Job name RACF Group RACF User Route codes Type
JRNL	Origin
MQS	Managers Queue ID Format Event Type Msgid CorrelId Msg Buffer
MSG	Acct Info Console ID Console Name Desc codes Jobclass Job name RACF Group RACF User Route codes Type
VAR	Journal Name Variable is

BMC Software, Inc., Confidential and Proprietary Information

Action Specification Fields

Not every action specification field can be used for every event type. The first column in Table 6-2 shows which event types can use a specific action specification field. Depending on which event type you create a Rule for, a panel for that event type appears and will contain a subset of the actions listed below.

The following table lists all of the available actions you can select for all the event types.

Table 6-2 Action Specification Fields (Part 1 of 2)

Event Type	Action Specification Field
All event types	Cmd (Type xxx) DOM ID DOM Exec EXEC name/Parms Info Issue WTO Msg Notify Send (TSO) Set variable
ALRT	Display at dest. Journal Route codes Reword ALERT
CICS	Display at dest. Journal
CMD	Display at dest. Journal Reword Msg Reject Command
DB2	Display at dest. Journal
EXT	Journal
IMS	Display at dest. Journal
JES3	Display at dest. Journal

BMC Software, Inc., Confidential and Proprietary Information

Table 6-2 Action Specification Fields (Part 2 of 2)

Event Type	Action Specification Field
JRNL	Display at dest.
MQS	Keep Message Destination Queue Remove DLH Destination QMgr
MSG	Console ID Console Name Descriptor codes Display at dest. Journal SYSLOG Display Update Rout/Desc Codes
TIME	Journal
VAR	Journal

Basic SYSPROG Services Command Summary

The commands in this section are available with MAINVIEW AutoOPERATOR for OS/390. The full set of advanced SYSPROG services can be used if made available to MAINVIEW AutoOPERATOR for OS/390. The EXEC listed is a provided utility EXEC to assist with the interface between an EXEC and a SYSPROG service.

Table 6-3 Basic SYSPROG Services Command Summary (Part 1 of 2)

Command	Operands	EXEC
ASM	[,MAP]	RASM
CPU	[,time | 30]	RCPU
CSA	[,MAP]	None
ENQUEUES	[,dsname]	None
	[,minor-name] [,major-name]	
	[,dsname/member,SPFEDIT]	
	[,,major-name]	

BMC Software, Inc., Confidential and Proprietary Information

Table 6-3 Basic SYSPROG Services Command Summary (Part 2 of 2)

Command	Operands	EXEC
ESTORAGE	[MAP][, interval | 5]	None
INFO		None
IO	[,volser | device-number]	RIO
MDEVICE	[,device-number[-range]] [,interval |15]	RMDE
MONITOR	,address-space-name	RMON
MPATH	[,path[-range]][, interval |15]	RMPA
MTP		RMTP
PAGING		RPAG
PROGRESS	,address-space-name	RPRO
REPLIES		RREP
RESERVES		RRES
RSM	[,MAP]	RRSM
SOFTFRR	[,recovery- modname] [,MAP] [,yy.ddd] [,errorid]	None
SPACE	[,dev-number] [,generic-or-esoteric-name] [,volser] [,partial-volser]	None
STATUS	[,(ALL | IN | TSO | as_name)] [,SHORT]	RSTA
SYSDUMP		RSYS
TIOT	[,ACT]	
TPIO	[,device-number]	RTPI
TSULIST		RTSU
USING	[,dvn | volser] [,ALL | IN]	None
VMCMD	cpcmd[#cpcmd ...]	None

BMC Software, Inc., Confidential and Proprietary Information

MQSeries Variables

MQSeries variables can be passed from Rules and from EXECs that are fired from those Rules or from the IMFEXEC MQI API.

General Purpose Variables for MQI API Only

Table 7-1 General Purpose Variables for MQI API Only

IMFMQI_STRUCTURES IMFMQI_OFFSETS IMFMQI_REASON

BMC Software, Inc., Confidential and Proprietary Information

Options and Miscellaneous Variables for MQI API Only

Table 7-2 Options and Miscellaneous Variables for MQI API Only

Variable	Call Type(s)
IMFMQI_CO_OPTIONS	CLOSE
IMFMQI_GMO_OPTIONS	GET
IMFMQI_PMO_OPTIONS	PUT, PUT1
IMFMQI_OO_OPTIONS	OPEN
IMFMQI_BUFFER	GET, PUT, PUT1
IMFMQI_BUFFLEN	GET, PUT, PUT1
IMFMQI_DATALEN	GET

BMC Software, Inc., Confidential and Proprietary Information

Message Descriptor Variables

Table 7-3 Message Descriptor Variables (Part 1 of 2)

IMFMQI_MD_MSGTYPE	IMFMQI_MD_REPORT
MQMT_DATAGRAM	MQRO_EXCEPTION
MQMT_REQUEST	MQRO_EXCEPTION_WITH_DATA
MQMT_REPLY	MQRO_EXCEPTION_WITH_FULL_DATA
MQMT_REPORT	MQRO_EXPIRATION
IMFMQI_MD_STRUCID	MQRO_EXPIRATION_WITH_DATA
IMFMQI_MD_VERSION	MQRO_EXPIRATION_WITH_FULL_DATA
IMFMQI_MD_FEEDBACK	MQRO_COA
MQFB_NONE	MQRO_COA_WITH_DATA
MQFB_COA	MQRO_COA_WITH_FULL_DATA
MQFB_COD	MQRO_COD
MQFB_EXPIRATION	MQRO_COD_WITH_DATA
MQFB_PAN	MQRO_COD_WITH_FULL_DATA
MQFB_NAN	MQRO_PAN
MQFB_QUIT	MQRO_NAN
IMFMQI_MD_ENCODING	MQRO_NEW_MSG_ID
MQENC_NATIVE	MQRO_PASS_MSG_ID
	MQRO_COPY_MSG_ID_TO_CORREL_ID
	MQRO_PASS_CORREL_ID
	MQRO_DEAD_LETTER_Q
	MQRO_DISCARD_MSG
	MQRO_NONE

BMC Software, Inc., Confidential and Proprietary Information

Table 7-3 Message Descriptor Variables (Part 2 of 2)

IMFMQI_MD_REPLYTOQMGR
IMFMQI_MD_USERIDENTIFIER
IMFMQI_MD_ACCOUNTINGTOKEN
  MQACT_NONE
IMFMQI_MD_APPLIDENTITYDATA
IMFMQI_MD_EXPIRY
  MQEI_UNLIMITED
IMFMQI_MD_BACKOUTCOUNT
IMFMQI_MD_REPLYTOQ
IMFMQI_MD_PRIORITY
IMFMQI_MD_PERSISTENCE
  MQPER_PERSISTENT
  MQPER_NOT_PERSISTENT
IMFMQI_MD_FORMAT
  MQFMT_NONE
  MQFMT_ADMIN
  MQFMT_CHANNEL_COMPLETED
  MQFMT_CICS
  MQFMT_COMMAND_1
  MQFMT_COMMAND_2
  MQFMT_DEAD_LETTER_HEADER
  MQFMT_EVENT
  MQFMT_IMS
  MQFMT_IMS_VAR_STRING
  MQFMT_MD_EXTENSION
  MQFMT_PCF
  MQFMT_REF_MSG_HEADER
  MQFMT_STRING
  MQFMT_TRIGGER
  MQFMT_WORK_INFO_HEADER
  MQFMT_XMIT_Q_HEADER
  MQFMT_RF_HEADER
  MQFMT_RF_HEADER_2
IMFMQI_MD_MSGID
IMFMQI_MD_OFFSET
IMFMQI_MD_MSGFLAGS
  MQMF_SEGMENTATION_INHIBITED
  MQMF_SEGMENTATION_ALLOWED
  MQMF_MSG_IN_GROUP
  MQMF_LAST_MSG_IN_GROUP
  MQMF_SEGMENT
  MQMF_LAST_SEGMENT
  MQMF_NONE
IMFMQI_MD_MSGSEQNUMBER
IMFMQI_MD_CODEDCHARSETID
  MQCCSI_Q_MGR
  MQCCSI_EMBEDDEDO
IMFMQI_MD_ORIGINALLENGTH
  MQOL_UNDEFINED.
IMFMQI_MD_PUTAPPLNAME
IMFMQI_MD_PUTDATE
IMFMQI_MD_PUTTIME
IMFMQI_MD_APPLORIGINDATA
IMFMQI_MD_GROUPID
IMFMQI_MD_PUTAPPLTYPE
  MQAT_AIX
  MQAT_BROKER
  MQAT_CICS
  MQAT_CICS_BRIDGE
  MQAT_CICS_VSE
  MQAT_DEFAULT
  MQAT_DOS
  MQAT_DQM
  MQAT_GUARDIAN
  MQAT_IMS
  MQAT_IMS_BRIDGE
  MQAT_JAVA
  MQAT_MVS
  MQAT_NOTES_AGENT
  MQAT_NSK
  MQAT_OS2
  MQAT_OS390
  MQAT_OS400
  MQAT_QMGR
  MQAT_UNKNOWN
  MQAT_UNIX
  MQAT_VMS
  MQAT_VOS
  MQAT_WINDOWS
  MQAT_WINDOWS_NT
  MQAT_XCF
IMFMQI_MD_CORRELID
  MQCI_NONE
  MQCI_NEW_SESSION

BMC Software, Inc., Confidential and Proprietary Information

Object Descriptor Variables

Table 7-4 Object Descriptor Variables

IMFMQI_OD_STRUCID IMFMQI_OD_VERSION IMFMQI_OD_OBJECTTYPE IMFMQI_OD_OBJECTNAME IMFMQI_OD_OBJECTQMGRNAME IMFMQI_OD_DYNAMICQNAME IMFMQI_OD_ALTERNATEUSERID IMFMQI_OD_RECSPRESENT MFMQI_OD_KNOWNDESTCOUNT IMFMQI_OD_UNKNOWNDESTCOUNT IMFMQI_OD_INVALIDDESTCOUNT IMFMQI_OD_OBJECTRECOFFSET IMFMQI_OD_RESPONSERECOFFSET IMFMQI_OD_OBJECTRECPTR IMFMQI_OD_RESPONSERECPTR IMFMQI_OD_ALTERNATESECURITYID IMFMQI_OD_RESOLVEDQNAME IMFMQI_OD_RESOLVEDQMGRNAME

Common MQseries Variables

In the following tables, prefix is

- "IMFQ" for MQ Rules and for EXECs fired from those Rules
- "IMFMQI" for the AutoOperator MQI (IMFEXEC MQI)

BMC Software, Inc., Confidential and Proprietary Information

Event Variables

Table 7-5 Event Variables

prefix_EVENT_TYPE	prefix_EVENT_AUXERRORDATAINT2
prefix_EVENT_QMGRNAME	prefix_EVENT_AUXERRORDATASTR1
prefix_EVENT_QNAME	prefix_EVENT_AUXERRORDATASTR2
prefix_EVENT_BASEQNAME	prefix_EVENT_AUXERRORDATASTR3
prefix_EVENT_APPLNAME	prefix_EVENT_TIMESINCERESET
prefix_EVENT_OBJECTQMGRNAME	prefix_EVENT_HIGHQDEPTH
prefix_EVENT_CHANNEL_NAME	prefix_EVENT_MSGENQCOUNT
prefix_EVENT_CONNECTIONNAME	prefix_EVENT_MSGDEQCOUNT
prefix_EVENT_XMITQNAME	prefix_EVENT_BRIDGENAME
prefix_EVENT_FORMAT	prefix_EVENT_BRIDGETYPE
prefix_EVENT_QTYPE	prefix_EVENT_OPTIONS
prefix_EVENT_APPLTYPE	prefix_EVENT_COMMAND
prefix_EVENT_REASONQUALIFIER	prefix_EVENT_PROCESSNAME
MQRQ_CHANNEL_STOPPED_OK	prefix_EVENT_USERIDENTIFIER
MQRQ_CHANNEL_STOPPED_ERROR	prefix_EVENT_AUXERRORDATAINT1
MQRQ_CHANNEL_STOPPED_RETRY	prefix_EVENT_CURDEPTH
MQRQ_CHANNEL_STOPPED_DISABLED	prefix_EVENT_MAXDEPTH
prefix_EVENT_ERRORIDENTIFIER	prefix_EVENT_PERCENTFULL
prefix_EVENT_ERRORMSGID	prefix_EVENT_AUXERRORDATAINT1
	prefix_EVENT_AUXERRORDATAINT2

BMC Software, Inc., Confidential and Proprietary Information

Dead Letter Header Variables

Table 7-6 Dead Letter Header Variables

prefix_DLH_STRUCID	prefix_DLH_ENCODING
prefix_DLH_VERSION	prefix_DLH_CODEDCHARSETID
prefix_DLH_REASON	prefix_DLH_PUTAPPLNAME
prefix_DLH_DESTQNAME	prefix_DLH_PUTDATE
prefix_DLH_DESTQMGRNAME	prefix_DLH_PUTTIME
prefix_DLH_FORMAT	prefix_DLH_PUTAPPLTYPE
MQFMT_NONE (IBM default)	MQAT_AIX
MQFMT_ADMIN	MQAT_CICS
MQFMT_CHANNEL_COMPLETED	MQAT_CICS_BRIDGE
MQFMT_CICS	MQAT_CICS_VSE
MQFMT_COMMAND_1	MQAT_DOS
MQFMT_COMMAND_2	MQAT_GUARDIAN
MQFMT_DEAD_LETTER_HEADER	MQAT_IMS
MQFMT_EVENT	MQAT_IMS_BRIDGE
MQFMT_IMS	MQAT_MVS
MQFMT_IMS_VAR_STRING	MQAT_NOTES_AGENT
MQFMT_MD_EXTENSION	MQAT_NSK
MQFMT_PC	MQAT_OS2
MQFMT_REF_MSG_HEADER	MQAT_OS390
MQFMT_STRING	MQAT_OS400
MQFMT_TRIGGER	MQAT_QMGR
MQFMT_WORK_INFO_HEADER	MQAT_UNIX
MQFMT_XMIT_Q_HEADER	MQAT_VMS
MQFMT_RF_HEADER	MQAT_VOS
MQFMT_RF_HEADER_2	MQAT_WINDOWS
	MQAT_WINDOWS_NT
	MQAT_XCF
	MQAT_UNKNOWN

BMC Software, Inc., Confidential and Proprietary Information

Trigger Message Variables.

Table 7-7 Trigger Message Variables

prefix_TM_STRUCID	prefix_TM_APPLTYPE
prefix_TM_VERSION	MQAT_AIX
prefix_TM_QNAME	MQAT_CICS
prefix_TM_PROCESSNAME	MQAT_CICS_BRIDGE
prefix_TM_TRIGGERDATA	MQAT_CICS_VSE
prefix_TM_APPLID	MQAT_DOS
prefix_TM_ENVDATA	MQAT_GUARDIAN
prefix_TM_USERDATA	MQAT_IMS
	MQAT_IMS_BRIDGE
	MQAT_MVS
	MQAT_NOTES_AGENT
	MQAT_NSK
	MQAT_OS2
	MQAT_OS390
	MQAT_OS400
	MQAT_QMGR
	MQAT_UNIX
	MQAT_VMS
	MQAT_VOS
	MQAT_WINDOWS
	MQAT_WINDOWS_NT
	MQAT_XCF
	MQAT_UNKNOWN

BMC Software, Inc., Confidential and Proprietary Information

CICS Information Header Variables

Table 7-8 CICS Information Header Variables

prefix_CIH_STRUCID	prefix_CIH_COMPCODE
prefix_CIH_VERSION	prefix_CIH_REASON
prefix_CIH_STRUCLENGTH	prefix_CIH_UOWCONTROL
prefix_CIH_ENCODING	MQCUOWC_ONLY
prefix_CIH_CODEDCHARSETID	MQCUOWC_CONTINUE
prefix_CIH_FORMAT	MQCUOWC_FIRST
MQFMT_NONE (IBM default)	MQCUOWC_MIDDLE
MQFMT_ADMIN	MQCUOWC_LAST
MQFMT_CHANNEL_COMPLETED	MQCUOWC_COMMIT
MQFMT_CICS	MQCUOWC_BACKOUT
MQFMT_COMMAND_1	prefix_CIH_GETWAITINTERVAL
MQFMT_COMMAND_2	MQCGWI_DEFAULT
MQFMT_DEAD_LETTER_HEADER	MQWI_UNLIMITED
MQFMT_EVENT	prefix_CIH_LINKTYPE
MQFMT_IMS	MQCLT_PROGRAM
MQFMT_IMS_VAR_STRING	MQCLT_TRANSACTION
MQFMT_MD_EXTENSION	prefix_CIH_OUTPUTDATALENGTH
MQFMT_PC	MQCLT_TRANSACTION
MQFMT_REF_MSG_HEADER	prefix_CIH_FACILITYKEEPTIME
MQFMT_STRING	prefix_CIH_ADSDESCRIPTOR
MQFMT_TRIGGER	MQCADSD_NONE
MQFMT_WORK_INFO_HEADER	MQCADSD_SEND
MQFMT_XMIT_Q_HEADER	MQCADSD_RECV
MQFMT_RF_HEADER	MQCADSD_MSGFORMAT
MQFMT_RF_HEADER_2	prefix_CIH_CONVERSATIONALTASK
prefix_CIH_FLAGS	MQCCT_YES
MQCIH_NONE.	MQCCT_NO
prefix_CIH_RETURNCODE	prefix_CIH_TASKENDSTATUS
MQCRC_OK	MQCTES_NOSYNC
MQCRC_CICS_EXEC_ERROR	MQCTES_COMMIT
MQCRC_MQ_API_ERROR	MQCTES_BACKOUT
MQCRC_BRIDGE_ERROR	MQCTES_ENDTASK
MQCRC_BRIDGE_ABEND	prefix_CIH_FACILITY
MQCRC_APPLICATION_ABEND	MQCFAC_NONE.
MQCRC_SECURITY_ERROR	
MQCRC_PROGRAM_NOT_AVAILABLE	
MQCRC_BRIDGE_TIMEOUT	
MQCRC_TRANSID_NOT_AVAILABLE	

BMC Software, Inc., Confidential and Proprietary Information

Table 7-8 CICS Information Header Variables

prefix_CIH_ABENDCODE	prefix_CIH_FUNCTION
prefix_CIH_AUTHENTICATOR	MQCFUNC_MQCONN
prefix_CIH_RESERVED1	MQCFUNC_MQGET
prefix_CIH_REPLYTOFORMAT	MQCFUNC_MQINQ
MQFMT_NONE (IBM default)	MQCFUNC_MQOPEN
MQFMT_ADMIN	MQCFUNC_MQPUT
MQFMT_CHANNEL_COMPLETED	MQCFUNC_MQPUT1
MQFMT_CICS	MQCFUNC_NONE
MQFMT_COMMAND_1	prefix_CIH_REMOTESYSID
MQFMT_COMMAND_2	prefix_CIH_REMOTETRANSID
MQFMT_DEAD_LETTER_HEADER	prefix_CIH_TRANSACTIONID
MQFMT_EVENT	prefix_CIH_FACILITYLIKE
MQFMT_IMS	prefix_CIH_ATTENTIONID
MQFMT_IMS_VAR_STRING	prefix_CIH_STARTCODE
MQFMT_MD_EXTENSION	MQCSC_START
MQFMT_PC	MQCSC_STARTDATA
MQFMT_REF_MSG_HEADER	MQCSC_TERMINPUT
MQFMT_STRING	MQCSC_NONE
MQFMT_TRIGGER	prefix_CIH_CANCELCODE
MQFMT_WORK_INFO_HEADER	prefix_CIH_NEXTTRANSACTIONID
MQFMT_XMIT_Q_HEADER	prefix_CIH_RESERVED2
MQFMT_RF_HEADER	prefix_CIH_RESERVED3
MQFMT_RF_HEADER_2	prefix_CIH_CURSORPOSITION
prefix_CIH_INPUTITEM	prefix_CIH_ERROROFFSET
prefix_CIH_RESERVED4	

BMC Software, Inc., Confidential and Proprietary Information

IMS Information Header Variables

Table 7-9 IMS Information Header Variables

prefix_IIH_STRUCID	prefix_IIH_REPLYTOFORMAT
prefix_IIH_VERSION	MQFMT_NONE (IBM default)
prefix_IIH_STRUCLENGTH	MQFMT_ADMIN
prefix_IIH_ENCODING	MQFMT_CHANNEL_COMPLETED
prefix_IIH_CODEDCHARSETID	MQFMT_CICS
prefix_IIH_FORMAT	MQFMT_COMMAND_1
MQFMT_NONE (IBM default)	MQFMT_COMMAND_2
MQFMT_ADMIN	MQFMT_DEAD_LETTER_HEADER
MQFMT_CHANNEL_COMPLETED	MQFMT_EVENT
MQFMT_CICS	MQFMT_IMS
MQFMT_COMMAND_1	MQFMT_IMS_VAR_STRING
MQFMT_COMMAND_2	MQFMT_MD_EXTENSION
MQFMT_DEAD_LETTER_HEADER	MQFMT_PC
MQFMT_EVENT	MQFMT_REF_MSG_HEADER
MQFMT_IMS	MQFMT_STRING
MQFMT_IMS_VAR_STRING	MQFMT_TRIGGER
MQFMT_MD_EXTENSION	MQFMT_WORK_INFO_HEADER
MQFMT_PC	MQFMT_XMIT_Q_HEADER
MQFMT_REF_MSG_HEADER	MQFMT_RF_HEADER
MQFMT_STRING	MQFMT_RF_HEADER_2
MQFMT_TRIGGER	prefix_IIH_COMMITMODE
MQFMT_WORK_INFO_HEADER	MQICM_COMMIT_THEN_SEND
MQFMT_XMIT_Q_HEADER	MQICM_SEND_THEN_COMMIT
MQFMT_RF_HEADER	prefix_IIH_SECURITYSCOPE
MQFMT_RF_HEADER_2	prefix_CIH_RESERVED4
prefix_IIH_FLAGS	MQISS_CHECK
MQIIH_NONE.	MQISS_FULL
prefix_IIH_LTERMOVERRIDE	prefix_IIH_RESERVED
prefix_IIH_MFSMAPNAME	prefix_IIH_TRANSTATE
prefix_IIH_AUTHENTICATOR	MQITS_IN_CONVERSATION
MQIAUT_NONE.	MQITS_NOT_IN_CONVERSATION
prefix_IIH_TRANINSTANCEID	MQITS_ARCHITECTED
MQITII_NONE.	

BMC Software, Inc., Confidential and Proprietary Information

Transmission Queue Header Variables

Table 7-10 Transmission Queue Header Variables (Part 1 of 2)

prefix_XQH_STRUCID	prefix_XQH_MD_STRUCID
prefix_XQH_VERSION	prefix_XQH_MD_VERSION
prefix_XQH_REMOTEQNAME	prefix_XQH_MD_REPORT
prefix_XQH_REMOTEQMGRNAME	prefix_XQH_MD_EXPIRY
MQRO_EXCEPTION	MQEI_UNLIMITED
MQRO_EXCEPTION_WITH_DATA	or a decimal between 1 and 999999999
MQRO_EXCEPTION_WITH_FULL_DATA	prefix_XQH_MD_ENCODING
MQRO_EXPIRATION	prefix_XQH_MD_CODEDCHARSETID
MQRO_EXPIRATION_WITH_DATA	MQCCSI_Q_MGR
MQRO_EXPIRATION_WITH_FULL_DATA	MQCCSI_EMBEDDED
MQRO_COA	or a decimal up to 999999999.
MQRO_COA_WITH_DATA	prefix_XQH_MD_FORMAT
MQRO_COA_WITH_FULL_DATA	MQFMT_NONE
MQRO_COD	MQFMT_ADMIN
MQRO_COD_WITH_DATA	MQFMT_CHANNEL_COMPLETED
MQRO_COD_WITH_FULL_DATA	MQFMT_CICS
MQRO_PAN	MQFMT_COMMAND_1
MQRO_NAN	MQFMT_COMMAND_2
MQRO_NEW_MSG_ID	MQFMT_DEAD_LETTER_HEADER
MQRO_PASS_MSG_ID	MQFMT_EVENT
MQRO_COPY_MSG_ID_TO_CORREL_ID	MQFMT_IMS
MQRO_PASS_CORREL_ID	MQFMT_IMS_VAR_STRING
MQRO_DEAD_LETTER_Q	MQFMT_MD_EXTENSION
MQRO_DISCARD_MSG	MQFMT_PCF
MQRO_NONE	MQFMT_REF_MSG_HEADER
prefix_XQH_MD_MSGTYPE	MQFMT_STRING
MQMT_DATAGRAM	MQFMT_TRIGGER
MQMT_REQUEST	MQFMT_WORK_INFO_HEADER
MQMT_REPLY	MQFMT_XMIT_Q_HEADER
MQMT_REPORT	MQFMT_RF_HEADER
prefix_XQH_MD_FEEDBACK	MQFMT_RF_HEADER_2
MQFB_NONE	prefix_XQH_MD_PRIORITY
MQFB_COA	prefix_XQH_MD_PERSISTENCE
MQFB_COD	MQPER_PERSISTENT
MQFB_EXPIRATION	MQPER_NOT_PERSISTENT
MQFB_PAN	prefix_XQH_MD_MSGID
MQFB_NAN	prefix_XQH_MD_CORRELID
MQFB_QUIT	prefix_XQH_MD_BACKOUTCOUNT

BMC Software, Inc., Confidential and Proprietary Information

Table 7-10 Transmission Queue Header Variables (Part 2 of 2)

prefix_XQH_MD_REPLYTOQ	prefix_XQH_MD_PUTAPPLNAME
prefix_XQH_MD_REPLYTOQMGR	prefix_XQH_MD_PUTDATE
prefix_XQH_MD_USERIDENTIFIER	prefix_XQH_MD_PUTTIME
prefix_XQH_MD_ACCOUNTINGTOKEN	prefix_XQH_MD_APPLORIGINDATA
prefix_XQH_MD_APPLIDENTITYDATA	prefix_XQH_MD_GROUPID
prefix_XQH_MD_PUTAPPLTYPE	prefix_XQH_MD_MSGSEQNUMBER
MQAT_AIX	prefix_XQH_MD_OFFSET
MQAT_CICS	prefix_XQH_MD_MSGFLAGS
MQAT_CICS_BRIDGE	If an MQMD_VERSION_2 MD is present, this variable may contain one or more of the following character strings delimited by blanks:
MQAT_CICS_VSE	MQMF_SEGMENTATION_INHIBITED
MQAT_DOS	MQMF_SEGMENTATION_ALLOWED
MQAT_GUARDIAN	MQMF_MSG_IN_GROUP
MQAT_IMS	MQMF_LAST_MSG_IN_GROUP
MQAT_IMS_BRIDGE	MQMF_SEGMENT
MQAT_MVS	MQMF_LAST_SEGMENT
MQAT_NOTES_AGENT	MQMF_NONE
MQAT_NSK	prefix_XQH_MD_ORIGINALLENGTH
MQAT_OS2	MQOL_UNDEFINED
MQAT_OS390	or a decimal between 1 and 999999999
MQAT_OS400	
MQAT_QMGR	
MQAT_UNIX	
MQAT_VMS	
MQAT_VOS	
MQAT_WINDOWS	
MQAT_WINDOWS_NT	
MQAT_XCF	
MQAT_UNKNOWN	

BMC Software, Inc., Confidential and Proprietary Information

PCF Variables

Table 7-11 PCF Variables

prefix_CFH_TYPE	prefix_CFH_MSGSEQNUMBER
MQCFT_EVENT	prefix_CFH_CONTROL
MQCFT_COMMAND	MQCFC_LAST
prefix_CFH_STRUCLENGTH	prefix_CFH_COMPCODE
prefix_CFH_VERSION	MQCC_OK
MQCFH_VERSION_1.	MQCC_WARNING
prefix_CFH_COMMAND	prefix_CFH_REASON
MQCMD_Q_MGR_EVENT	prefix_CFH_PARAMETERCOUNT
MQCMD_PERFM_EVENT	
MQCMD_CHANNEL_EVENT	

Message Descriptor Extension Variables

Table 7-12 Message Descriptor Extension Variables

prefix_MDE_STRUCID	prefix_MDE_FORMAT
prefix_MDE_VERSION	MQFMT_NONE
prefix_MDE_STRUCLENGTH	MQFMT_ADMIN
prefix_MDE_ENCODING	MQFMT_CHANNEL_COMPLETED
prefix_MDE_CODEDCHARSETID	MQFMT_CICS
prefix_MDE_FLAGS	MQFMT_COMMAND_1
prefix_MDE_GROUPID	MQFMT_COMMAND_2
prefix_MDE_MSGSEQNUMBER	MQFMT_DEAD_LETTER_HEADER
prefix_MDE_OFFSET	MQFMT_EVENT
prefix_MDE_MSGFLAGS	MQFMT_IMS
MQMF_SEGMENTATION_INHIBITED	MQFMT_IMS_VAR_STRING
MQMF_SEGMENTATION_ALLOWED	MQFMT_MD_EXTENSION
MQMF_MSG_IN_GROUP	MQFMT_PCF
MQMF_LAST_MSG_IN_GROUP	MQFMT_REF_MSG_HEADER
MQMF_SEGMENT	MQFMT_STRING
MQMF_LAST_SEGMENT	MQFMT_TRIGGER
MQMF_NONE	MQFMT_WORK_INFO_HEADER
prefix_MDE_ORIGINALLENGTH	MQFMT_XMIT_Q_HEADER
MQOL_UNDEFINEDContains	MQFMT_RF_HEADER
or a decimal between 1 and 999999999	MQFMT_RF_HEADER_2

BMC Software, Inc., Confidential and Proprietary Information

Work Information Header Variables

Table 7-13 Work Information Header Variables

prefix_WIH_STRUCID
prefix_WIH_VERSION
prefix_WIH_STRUCLENGTH
prefix_WIH_ENCODING
prefix_WIH_CODEDCHARSETID
prefix_WIH_FLAGS
prefix_WIH_SERVICENAME
prefix_WIH_SERVICESTEP
prefix_WIH_MSGTOKEN
prefix_WIH_RESERVED
prefix_WIH_FORMAT
  MQFMT_NONE
  MQFMT_ADMIN
  MQFMT_CHANNEL_COMPLETED
  MQFMT_CICS
  MQFMT_COMMAND_1
  MQFMT_COMMAND_2
  MQFMT_DEAD_LETTER_HEADER
  MQFMT_EVENT
  MQFMT_IMS
  MQFMT_IMS_VAR_STRING
  MQFMT_MD_EXTENSION
  MQFMT_PCF
  MQFMT_REF_MSG_HEADER
  MQFMT_STRING
  MQFMT_TRIGGER
  MQFMT_WORK_INFO_HEADER
  MQFMT_XMIT_Q_HEADER
  MQFMT_RF_HEADER
  MQFMT_RF_HEADER_2

BMC Software, Inc., Confidential and Proprietary Information

Reference Message Header Variables

Table 7-14 Reference Message Header Variables

prefix_RMH_FORMAT	prefix_RMH_STRUCID
MQFMT_NONE	prefix_RMH_VERSION
MQFMT_ADMIN	prefix_RMH_STRUCLENGTH
MQFMT_CHANNEL_COMPLETED	prefix_RMH_ENCODING
MQFMT_CICS	prefix_RMH_CODEDCHARSETID
MQFMT_COMMAND_1	prefix_RMH_FLAGS
MQFMT_COMMAND_2	MQRMHF_LAST
MQFMT_DEAD_LETTER_HEADER	MQRMHF_NOT_LAST
MQFMT_EVENT	prefix_RMH_OBJECTTYPE
MQFMT_IMS	prefix_RMH_OBJECTINSTANCEID
MQFMT_IMS_VAR_STRING	prefix_RMH_SRCENVLENGTH
MQFMT_MD_EXTENSION	prefix_RMH_SRCENVOFFSET
MQFMT_PCF	prefix_RMH_SRCNAMELENGTH
MQFMT_REF_MSG_HEADER	prefix_RMH_SRCNAMEOFFSET
MQFMT_STRING	prefix_RMH_DESTENVLENGTH
MQFMT_TRIGGER	prefix_RMH_DESTENVOFFSET
MQFMT_WORK_INFO_HEADER	prefix_RMH_DESTNAMELENGTH
MQFMT_XMIT_Q_HEADER	prefix_RMH_DESTNAMEOFFSET
MQFMT_RF_HEADER	prefix_RMH_DATALOGICALLENGTH
MQFMT_RF_HEADER_2	prefix_RMH_DATALOGICALOFFSET
	prefix_RMH_DATALOGICALOFFSET2

Trigger Message 2 (Character format) Variables

Table 7-15 Trigger Message 2 (Character format) Variables

prefix_TMC_STRUCID
prefix_TMC_VERSION
prefix_TMC_QNAME
prefix_TMC_PROCESSNAME
prefix_TMC_TRIGGERDATA
prefix_TMC_APPLTYPE
prefix_TMC_APPLID
prefix_TMC_ENVDATA
prefix_TMC_USERDATA
prefix_TMC_QMGRNAME

BMC Software, Inc., Confidential and Proprietary Information

Rules and Formatting Header Variables

Table 7-16 Rules and Formatting Header Variables

prefix_RFH_STRUCID prefix_RFH_VERSION prefix_RFH_STRUCLENGTH prefix_RFH_ENCODING prefix_RFH_CODEDCHARSETID prefix_RFH_STRUCID prefix_RFH_FLAGS MQRFH_NONE. prefix_RFH_NAMEVALUESTRING IMFQ_RFH_STRUCLENGTH MQRMH_STRUC_LENGTH_FIXED prefix_RFH_FORMAT MQFMT_NONE MQFMT_ADMIN MQFMT_CHANNEL_COMPLETED MQFMT_CICS MQFMT_COMMAND_1 MQFMT_COMMAND_2 MQFMT_DEAD_LETTER_HEADER MQFMT_EVENT MQFMT_IMS MQFMT_IMS_VAR_STRING MQFMT_MD_EXTENSION MQFMT_PCF MQFMT_REF_MSG_HEADER MQFMT_STRING MQFMT_TRIGGER MQFMT_WORK_INFO_HEADER MQFMT_XMIT_Q_HEADER MQFMT_RF_HEADER MQFMT_RF_HEADER_2

BMC Software, Inc., Confidential and Proprietary Information

Rules and Formatting Header Variables Version 2

Table 7-17 Rules and Formatting Header Variables Version 2

prefix_RFH2_STRUCID prefix_RFH2_VERSION_2 prefix_RFH2_STRUCLENGTH prefix_RFH2_ENCODING prefix_RFH2_CODEDCHARSETID prefix_RFH2_FLAGS prefix_RFH2_NAMEVALUECCSID prefix_RFH2_NAMEVALUEDATA prefix_RFH2_FORMAT MQFMT_NONE MQFMT_ADMIN MQFMT_CHANNEL_COMPLETED MQFMT_CICS MQFMT_COMMAND_1 MQFMT_COMMAND_2 MQFMT_DEAD_LETTER_HEADER MQFMT_EVENT MQFMT_IMS MQFMT_IMS_VAR_STRING MQFMT_MD_EXTENSION MQFMT_PCF MQFMT_REF_MSG_HEADER MQFMT_STRING MQFMT_TRIGGER MQFMT_WORK_INFO_HEADER MQFMT_XMIT_Q_HEADER MQFMT_RF_HEADER MQFMT_RF_HEADER_2

BMC Software, Inc., Confidential and Proprietary Information

MQI Command Syntax Summary

Using EXECs, you can use the MAINVIEW AutoOPERATOR for MQSeries EXEC interface to communicate with MQSeries objects.

MQI Commands and Examples

Table 8-1 MQI Commands and Examples (Part 1 of 6)

Syntax:	**MQI BACK** [HCONN(hconn)]
Function:	Backs out changes since the last sync point.
Note:	This parameter specifies a variable created in an earlier CONN command, it is not required.
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI BACK “IMFEXEC MQI BACK”
Syntax:	**MQI CLOSE** [HCONN(hconn)] [HOBJ(hobj)] [COPTS(’options’)]
Function:	Releases access to an object.
Notes:	The **COPTS parameter s**pecifies the options that control the CLOSE action. Valid close options: MQCO_NONE MQCO_DELETE MQCO_DELETE_PURGE
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.

BMC Software, Inc., Confidential and Proprietary Information

Table 8-1 MQI Commands and Examples (Part 2 of 6)

Examples:	IMFEXEC MQI CLOSE COPTS(’MQCO_NONE’) "IMFEXEC MQI CLOSE COPTS(’MQCO_NONE’)"
Syntax:	**MQI CMIT** [HCONN(hconn)]
Function:	Commits all changes since last sync point.
Notes:	This statement invokes the MQSeries CMIT command which commits all new messages and changes to queues opened with the SYNCPoint.
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI CMIT "IMFEXEC MQI CMIT"
Syntax:	**MQI CONN** NAME(qmgr) [HCONN(hconn)]
Function:	Connect to a queue manager
Notes:	With the MQI CONN statement, an application can connect to a queue manager. It returns a queue manager connection handle to the EXEC, which is used by the application for all other IMFEXEC MQI commands.
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI CONN NAME(CSQ1) “IMFEXEC MQI CONN NAME(CSQ1)”
Syntax:	**MQI DISC** [HCONN(hconn)]
Function:	Disconnects from a queue manager.
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI DISC "IMFEXEC MQI DISC"
Syntax:	**MQI GET** [HCONN(hconn)] [HOBJ(hobj)] [GOPTS(’options’)] [BUFFLEN(bufferlength)] [BUFFER(buffer)] [DATALEN(datalength)]
Function:	Gets a message from a local queue.

BMC Software, Inc., Confidential and Proprietary Information

Table 8-1 MQI Commands and Examples (Part 3 of 6)

Notes:	The **GOPTS parameter s**pecifies the options that control the GET action. More than one option can be specified as any combination of the following choices: Valid GET options: MQGMO_WAIT MQGMO_NO_WAIT MQGMO_SYNCPOINT MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_BROWSE_FIRST MQGMO_BROWSE_NEXT MQGMO_MSG_UNDER_CURSOR MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_SET_SIGNAL MQGMO_FAIL_IF_QUIESCING MQGMO_CONVERT MQGMO_NONE REMOVE_DLH
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI GET BUFFER(GETDATA) "IMFEXEC MQI GET BUFFER(GETDATA)"
Syntax:	**MQI OPEN** [HCONN(hconn)] [HOBJ(hobj)] [OOPTS(’options’)]
Function:	Establishes access to an object.

BMC Software, Inc., Confidential and Proprietary Information

Table 8-1 MQI Commands and Examples (Part 4 of 6)

Notes:	The **OOPTS parameter** specifies the options that control the OPEN action. More than one option can be specified as any combination of the following choices: Valid OPEN options: MQOO_BIND_NOT_FIXED MQOO_BIND_ON_OPEN MQOO_FAIL_IF_QUIESCING MQOO_ALTERNATE_USER_AUTHORITY MQOO_SET_ALL_CONTEXT MQOO_SET_IDENTITY_CONTEXT MQOO_PASS_ALL_CONTEXT MQOO_PASS_IDENTITY_CONTEXT MQOO_SAVE_ALL_CONTEXT MQOO_INQUIRE MQOO_OUTPUT MQOO_BROWSE MQOO_INPUT_EXCLUSIVE MQOO_INPUT_SHARED MQOO_INPUT_AS_Q_DEF MQOO_BIND_AS_Q_DEF
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI OPEN OOPTS(’MQOO_OUTPUT’) "IMFEXEC MQI OPEN OOPTS(’MQOO_OUTPUT’)"
Syntax:	**MQI PUT** [HCONN(hconn)] [HOBJ(hobj)] [POPTS(’options’)] [BUFFLEN(bufferlength)] [BUFFER(buffer)]
Function:	Puts one or more messages into a queue.
Notes:	The **POPTS parameter** specifies the options that control the PUT action. Options can be specified as any valid combination of the following choices: MQPMO_SYNCPOINT MQPMO_NO_SYNCPOINT MQPMO_NO_CONTEXT MQPMO_DEFAULT_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT MQPMO_PASS_ALL_CONTEXT MQPMO_SET_IDENTITY_CONTEXT MQPMO_SET_ALL_CONTEXT MQPMO_ALTERNATE_USER_AUTHORITY MQPMO_FAIL_IF_QUIESCING MQPMO_NONE (IBM default)
Cond Codes:	See “Completion Codes for MQI Commands” on page 8-6.
Examples:	IMFEXEC MQI PUT BUFFER(PUTDATA) "IMFEXEC MQI PUT BUFFER(PUTDATA)"

BMC Software, Inc., Confidential and Proprietary Information

Table 8-1 MQI Commands and Examples (Part 5 of 6)

Syntax:	**MQI PUT1** [HCONN(hconn)] [POPTS('options')] [BUFFLEN(bufferlength)] [BUFFER(buffer)]
Function:	Puts one message into a queue with automatic open and close of the queue.
Notes:	The **POPTS parameter** specifies the options that control the PUT1 action. The variable IMFMQI_PMO_OPTIONS is not required. Valid PUT1 options: MQPMO_SYNCPOINT MQPMO_NO_SYNCPOINT MQPMO_NO_CONTEXT MQPMO_DEFAULT_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT MQPMO_PASS_ALL_CONTEXT MQPMO_SET_IDENTITY_CONTEXT MQPMO_SET_ALL_CONTEXT MQPMO_ALTERNATE_USER_AUTHORITY MQPMO_FAIL_IF_QUIESCING MQPMO_NONE (IBM default)
Cond Codes:	See "Completion Codes for MQI Commands" on page 8-6.
Examples:	IMFEXEC MQI PUT1 BUFFER(PUTDATA) "IMFEXEC MQI PUT1 BUFFER(PUTDATA)"
Syntax:	**COPY MQI** STRTYPE(structure IDs) [STRFROM(input prefix)] STRTO(output prefix)
Function:	Copies entire structures based on input parameters.
Examples:	IMFEXEC COPY MQI STRTYPE(&IMFMQI_STRUCTURES) STRFROM(IMFMQI_) STRTO(IMFMQX_) "IMFEXEC COPY MQI STRTYPE("IMFMQI_STRUCTURES") STRFROM(IMFMQI_) STRTO(IMFMQX_)"
Syntax:	**DISPLAY MQI** STRTYPE(structure IDs) [PREFIX(input prefix)]
Function:	Displays the contents of a set (or sets) of variables created from a structure (or multiple structures) found in an MQSeries message in the BBI journal. The variable contents are displayed in character and hexadecimal formats.
Examples:	IMFEXEC DISPLAY MQI STRTYPE(&IMFMQI_STRUCTURES) PREFIX(IMFMQI_) "IMFEXEC DISPLAY MQI STRTYPE("IMFMQI_STRUCTURES") PREFIX(IMFMQI_)""

BMC Software, Inc., Confidential and Proprietary Information

Table 8-1 MQI Commands and Examples (Part 6 of 6)

Syntax:	**CMD** ‘MQ command’ TYPE(MQS) [LM(queue manager name)] [RESPONSE(msgid1,msgid2,…)] [WAIT(30 | nnnn)] [PCF(YES | NO)] [QM(queue-manager-name)] [CQ(command-queue- name)]
Function:	Issues commands to the MQSeries command server. A response is returned to the issuing EXEC.
Cond Codes:	See “Completion Codes for MQI Commands”.
Examples:	**IMFEXEC CMD 'DISPLAY QUEUE(SYSTEM.CHANNEL.*)’ TYPE(MQS)** **"IMFEXEC CMD 'DISPLAY QUEUE(SYSTEM.CHANNEL.*)’ TYPE(MQS), LM(MQS1) RESP(CSQN205I,CSQ9022I)"**

Completion Codes for MQI Commands

IMFMQCC	IMFMQRC	Reason
0	0	successful completion
2	8	MQSeries returned error
	12	buffer too small
4	16	syntax error
	20	variable processing routine not found

BMC Software, Inc., Confidential and Proprietary Information

Continuous State Management (CSM)

CSMACT Commands

Table 9-1 is a list of valid commands you can use with the CSMACT EXEC to affect the state of an object.

Table 9-1 CSMACT Commands (Part 1 of 3)

Syntax:	%CSMACT object-name **BOUNCE**
Function:	Stops and immediately restarts an object.
Note:	When used on an object that is UP, UPEARLY, UPFORCE, UPLOGICAL, or UPLOGICALE, the resulting state is UP. When issued on an object that is COMPLETE, the resulting state is COMPLETE.
Syntax:	%CSMACT object-name **CANCEL**
Function:	Issues the cancel command for the object.
Note:	When issued on an object in any state, the resultant desired state is DOWNFORCE. Requires the DEP or NODEP parameter.
Syntax:	%CSMACT object-name **DOWN** DEP|NODEP
Function:	Stops an object.
Note:	When issued on an object that is DOWN, the resultant desired state is DOWN. When issued on an object that is in any state except DOWN, the resultant desired state is DOWNFORCE. Requires the DEP or NODEP parameter.
Syntax:	%CSMACT object-name **DOWNEARLY** DEP|NODEP
Function:	Stops an object and continues under control of CSM.

BMC Software, Inc., Confidential and Proprietary Information

Table 9-1 CSMACT Commands (Part 2 of 3)

Note:	When issued on any object, the resultant state is DOWNEARLY. Requires the DEP or NODEP parameter.
Syntax:	%CSMACT object-name **DOWNLOGICAL** DEP|NODEP
Function:	Stops a grouping object and removes it from CSM control.
Note:	When issued on an object that is in any state, the resultant desired state is DOWNLOGICAL. Requires the DEP or NODEP parameter.
Syntax:	%CSMACT object-name **DOWNLOGICALE** DEP|NODEP
Function:	Stops a grouping object and continues under CSM control.
Note:	When issued on an object that is in any state, the resultant desired state is DOWNLOGICALE. Requires the DEP or NODEP parameter.
Syntax:	%CSMACT object-name **MOVE** [target-group | LOCAL]
Function:	Moves management of an object to another CSM group.
Note:	Requires the 4-character CSM group name of the targeted CSM group.
Syntax:	**%CSMACT** object-name **RESET DEP|NODEP**
Function:	Sets an object's desired state based on its schedule and returns an object to CSM control.
Note:	When issued on an object that is UP, the resultant state is UP. When issued on an object that is DOWN, the resultant state is DOWN. You can use the DEP or REQ parameters but they are not required
Syntax:	**%CSMACT** object-name **RESETC [DEP|NODEP]**
Function:	Resets an object and the command counter to zero.
Note:	When issued on an object that is UP, the resultant state is UP. When issued on an object that is DOWN, the resultant state is DOWN.
Syntax:	**%CSMACT** object-name **UP REQ|NOREQ**
Function:	Starts an object.
Note:	When issued on an object that is UP, the resultant desired state is UP. When issued on an object that is in any state, the resultant desired state is UPFORCE. Requires the **REQ or NOREQ parameter.**
Syntax:	**%CSMACT** object-name **UPEARLY REQ|NOREQ**
Function:	Starts an object and continues under CSM control.
Note:	When issued on an object that is in any state, the resultant desired state is UPEARLY. Requires the **REQ or NOREQ parameter.**
Syntax:	**%CSMACT** object-name **UPLOGICAL REQ|NOREQ**

BMC Software, Inc., Confidential and Proprietary Information

Table 9-1 CSMACT Commands (Part 3 of 3)

Function:	Starts a grouping object.
Note:	When issued on an object that is in any state, the resultant desired state is UPLOGICAL or COMPLETE. Requires the **REQ or NOREQ parameter.**
Syntax:	**%CSMACT** object-name **UPLOGICALE REQ|NOREQ**
Function:	Starts a grouping object.
Note:	When issued on an object that is in any state, the resultant desired state is UPLOGICALE or COMPLETE. Requires the **REQ or NOREQ parameter.**

CSM Command Line Interface

Table 9-2 CSM Command Line Interface (Part 1 of 2)

Syntax:	**STOP** object-name **[EARLY|LOGICAL|LOGICALE] [DEP|NODEP]**
Function:	Stops an object.
Example:	CSMNS61 **STOP** IMS DEP
Syntax:	**START** object-name **[EARLY|LOGICAL|LOGICALE] [DEP|NODEP]**
Function:	Starts an object.
Example:	CSMNS61 **START** TSO EARLY
Syntax:	**CANCEL** object-name
Function:	Cancels execution of object.
Example:	CSMNS61 **CANCEL** TCPIP
Syntax:	**BOUNCE** object-name
Function:	Stops and then immediately restarts an object.
Example:	CSMNS61 **BOUNCE** TCPIP
Syntax:	**MOVE** object-name [target-group | LOCAL]
Function:	Transfers monitoring and management from one CSM group to another. Execution of the object will move from one OS/390 image to another, if necessary.
Example:	CSMNS61 **MOVE** IMS SYSA
Syntax:	**STATUS** object-name

BMC Software, Inc., Confidential and Proprietary Information

Table 9-2 CSM Command Line Interface (Part 2 of 2)

Function:	Reports the desired and actual state of one or more objects. If an object name is not specified, all objects in a particular group will be reported on.
Example:	CSMNS61 **STATUS**
Syntax:	**RESET** object-name
Function:	Resets object state to that specified in the object's schedule.
Example:	CSMNS61 **RESET** APPC
Syntax:	**ENABLE group-name target-ssid**
Function:	Enables the processing of a group for a local subsystem specified in the local repository,
Example:	CSMNS61 **ENABLE** GRPA SYSA
Syntax:	**DISABLE**
Function:	Disables processing of a group for a local subsystem specified in the local repository.
Example:	CSMNS61 **DISABLE**

BMC Software, Inc., Confidential and Proprietary Information

AOAnywhere Command Summary

Commands and Examples

Table 10-1 AOEXEC Commands (Part 1 of 4)

Syntax:	**AOEXEC ALERT** [KEY( )] [TEXT('text string')] [FUNCTION(ADD | COUNT | CREATEQ | DELETE | DELETEQ | LISTQ | READQ)] [ALARM(YES | NO)] [COLOR(RED | PINK | YELLOW | DKBLUE | LTBLUE | GREEN | WHITE)] [DISPOSE(KEEP | DELETE)] [ESCALATE(UP | DOWN)] [ESCEXEC('execname p1 p2 p3 ... pn')] [EXEC('execname p1 p2 p3 ... pn')] [HELP(panelname)] [INTERVAL(nnnn,nnnn,nnnn,nnnn,nnnn,nnnn)] [PCMD('cmd string')] [POSITION(position)] [PRI(CRITICAL | MAJOR | MINOR | WARNING | INFORMATIONAL | CLEARING)] [PUBLISH(REPLACE | ADD | NO] [QUEUE(MAIN | queue name)] [RETAIN(YES | NO)] SS | SSID(subsystem identifier) [SYSTEM(YES | NO)] [TGTSS(target subsystem identifier)] [ORIGIN(origin)] [UDATA('user data')] [USER(user name)]

BMC Software, Inc., Confidential and Proprietary Information

Table 10-1 AOEXEC Commands (Part 2 of 4)

Function:	Creates and manages exception messages and message queues that can be displayed by any of the STATUS applications and ALERT Management Facility applications.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC ALERT KEY(NETW2) + TEXT(’COMMUNICATION LINES DOWN: /N - DALLAS /N - CHICAGO’) FUNCTION(ADD) QUEUE(NETWORK) + PRIORITY(CRITICAL) COLOR(PINK) SSID(RE61) "AOEXEC ALERT KEY(NETW2)” , “TEXT(’COMMUNICATION LINES DOWN: /N - DALLAS /N - CHICAGO’) FUNCTION(ADD) QUEUE(NETWORK) ", "PRIORITY(CRITICAL) COLOR(PINK) SSID(RE61)"
Syntax:	**AOEXEC MSG** 'Message text' SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)]
Function:	Logs a message in the BBI-SS PAS Journal log.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC MSG 'DATABASE IS OFFLINE' SSID(RE61) TGTSS(CICA) “AOEXEC MSG 'DATABASE IS OFFLINE' SSID(RE61) TGTSS(CICA)”
Syntax:	**AOEXEC NOTIFY** NAME(Elan contact name) [INFO(‘Text’)] SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)]
Function:	Sends a request through AutoOPERATOR to issue a pager call using the MAINVIEW Elan Workstation component (if it is installed).
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC NOTIFY NAME(SYSPROG) INFO(SYSTEM) SSID(RE61) "AOEXEC NOTIFY NAME(SYSPROG) INFO(SYSTEM) SSID(RE61)"
Syntax:	**AOEXEC SELECT** EXEC(‘execname parm1... parm2... parm*n*’) [PRI(NORMAL | HIGH)] [WAIT(NO | YES)] SSID(subsystem-identifier) [TGTSS(target susbsystem identifier)] [VAR(var1.... var2.... var3... var*n*)]
Function:	Invokes an EXEC or a program.
Notes:	If you use WAIT | YES, see IMFEXEC SHARE in Table 4-1, “IMFEXEC Command Syntax Summary,” on page 4-2.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC SELECT EXEC(‘CHKENQ MACLIB’) SSID(RE61) TGTSS(SYSB) "AOEXEC SELECT EXEC(‘CHKENQ MACLIB’) SSID(RE61) TGTSS(SYSB)"

BMC Software, Inc., Confidential and Proprietary Information

Table 10-1 AOEXEC Commands (Part 3 of 4)

Syntax:	AOEXEC SYSINFO [SS | SSID(subsystem identifier)] [GROUP()]
Function:	Returns a list in the SSID# variables that contains AutoOPERATOR subsystems that are available for communications in this XCF group. Variable LCNT contains the count of subsystems. Variable SYSN# contains the MVS system ID for the corresponding AutoOPERATOR.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC SYSINFO "AOEXEC SYSINFO"
Syntax:	**AOEXEC VDEL** [POOL(SHARED | PROFILE)] SS | SSID(subsystem identifier) [TGTSS(target susbsystem identifier)] [VAR(var1....var2....var3...var*n*)]
Function:	Deletes one or more variables from one of the AutoOPERATOR variable pools.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VDEL VAR(LINE1) POOL(SHARED) SSID(RE61) "AOEXEC VDEL VAR(LINE1) POOL(SHARED) SSID(RE61)"
Syntax:	**AOEXEC VGET** [POOL(SHARED | PROFILE)] SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)] VAR(var1 var2 var3 var*n*)
Function:	Copies one or more variables from one of the AutoOPERATOR pools into the EXECs function pool.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VGET VAR(LINE1) POOL(SHARED) SSID(RE61)" "AOEXEC VGET VAR(LINE1) POOL(SHARED) SSID(RE61)"
Syntax:	**AOEXEC VLST** [POOL(SHARED | PROFILE)] SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)] VAR(variable name)
Function:	Lists variable names defined in the AutoOPERATOR pools.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VLST VAR(RETRY*) POOL(SHARED) SSID(RE61) "AOEXEC VLST VAR(RETRY*) POOL(SHARED) SSID(RE61)"
Syntax:	**AOEXEC VPUT** [POOL(SHARED | PROFILE)] SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)] VAR(var1....var2....var3...var*n*)
Function:	Copies one or more variables from the EXECs function pool into one of the AutoOPERATOR pools.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.

BMC Software, Inc., Confidential and Proprietary Information

Table 10-1 AOEXEC Commands (Part 4 of 4)

Examples:	AOEXEC VPUT VAR(ABENDS) POOL(PROFILE) SSID(RE61) "AOEXEC VPUT VAR(ABENDS) POOL(PROFILE) SSID(RE61)"
Syntax:	**AOEXEC VDELL** [POOL(SHARED | PROFILE)] VAR(var1....var2....var3...var*n*) SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)]
Function:	Deletes one or more long variables from one of the AutoOPERATOR variable pools.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VDELL VAR(X) POOL(PROFILE) SSID(RE61) "AOEXEC VDELL VAR(X) POOL(PROFILE) SSID(RE61)"
Syntax:	**AOEXEC VGETL** [POOL(SHARED | PROFILE)] VAR(var1....var2....var3...var*n*) SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)]
Function:	Copies one or more long variables from one of the AutoOPERATOR pools into the TSO pool.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VGETL VAR(X) POOL(PROFILE) SSID(RE61) "AOEXEC VGETL VAR(X) POOL(PROFILE) SSID(RE61)"
Syntax:	**AOEXEC VLSTL** [POOL(SHARED | PROFILE)] VAR(var) SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)]
Function:	Lists variable names defined in the AutoOPERATOR pools.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VLSTL VAR(*) POOL(SHARED) SSID(RE61) "AOEXEC VLSTL VAR(*) POOL(SHARED) SSID(RE61)"
Syntax:	**AOEXEC VPUTL** [POOL(SHARED | PROFILE)] VAR(var1....var2....var3...var*n*) SS | SSID(subsystem identifier) [TGTSS(target subsystem identifier)]
Function:	Creates or sets a long variable from a variable in the TSO pool.
Return Codes:	See “AOAnywhere Return Codes” on page 10-5.
Examples:	AOEXEC VPUTL VAR(A) POOL(SHARED) SSID(RE61) "AOEXEC VPUTL VAR(A) POOL(SHARED) SSID(RE61)"

BMC Software, Inc., Confidential and Proprietary Information

AOAnywhere Return Codes

Return Code	Description
40	security definitions disallowed access to this function on specified subsystem.
48	incompatible AutoOPERATOR versions exist.
52	An attempt was made to executed this function under NetView without a valid Access/NetView product key.
>2048 return code	When specifying WAIT(Y) and the command was executed successfully, a value 2048 is added to the EXEC's exit code before the return code is generated.

BMC Software, Inc., Confidential and Proprietary Information

IMFEXEC ARRAY Commands

The IMFEXEC ARRAY | ARY commands enable you to access data from two-dimensional variable arrays. Arrays bear some resemblance to ISPF tables in the way they can be scanned, sorted, and positioned arrays and are backed by disk space.

IMFEXEC ARRAY Commands and Examples

Table 11-1 IMFEXEC ARRAY Commands and Examples (Part 1 of 5)

Function	Command and Format	
Syntax:	**ARRAY CONNECT** name [ACCESS(UPDATE | READ)] [TOKEN()] [MSG | NOMSG]	
Function:	Establishes a logical connection between one or more EXECs and an array.	
Cond Codes:	**0**	Command was executed successfully.
	8	Array was not found or an error occurred reading from disk and a temp copy could not be copied.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY CONNECT DASDSTATS TOKEN(&arytoken) "IMFEXEC ARRAY CONNECT DASDSTATS TOKEN("arytoken")"	
Syntax:	**ARRAY CREATE** name STEM(stem name) INITIAL() [INC()]	
Function:	Defines a new array by providing definitions of its logical characteristics.	

BMC Software, Inc., Confidential and Proprietary Information

Table 11-1 IMFEXEC ARRAY Commands and Examples (Part 2 of 5)

Function	Command and Format	
Cond Codes:	**0**	Command was executed successfully.
	8	Invalid or incomplete array definition found.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY CREATE DASDSTATS STEM(ARRAY) INITIAL(500) INC(50) "IMFEXEC ARRAY CREATE DASDSTATS STEM(ARRAY) INITIAL(500) INC(50)"	
Syntax:	**ARRAY DELETE** name	
Function:	Deletes a row from an array.	
Cond Codes:	**0**	Command was executed successfully.
	4	Array is empty.
	8	Array is not found.
	12	Array is not in UPDATE access.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY DELETE DASDSTATS "IMFEXEC ARRAY DELETE DASDSTATS"	
Syntax:	ARRAY DISC name [ACTION(SAVE | NOSAVE | DELETE | KEEP)]	
Function:	Terminates a logical connection between one or more EXECs and an array.	
Cond Codes:	**0**	Command was executed successfully.
	8	Array was not found.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY DISC DASDSTATS "IMFEXEC ARRAY DISC DASDSTATS"	
Syntax:	**ARRAY FIND** name [ROW()] [CRITERIA()]	
Function:	Locates a particular row conforming to a set of criteria.	
Cond Codes:	**0**	Command was successfully executed.
	4	Criteria parsing error occurred.
	8	Array was not found.
	12	Row specification past end of array or 0.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY FIND USERID CRITERIA('ACCT,,,=,"3911"') ROW(1) "IMFEXEC ARRAY FIND USERID CRITERIA('ACCT,,,=,"3911"') ROW(1)"	

BMC Software, Inc., Confidential and Proprietary Information

Table 11-1 IMFEXEC ARRAY Commands and Examples (Part 3 of 5)

Function	Command and Format	
Syntax:	**ARRAY GET** name [TRIM | NOTRIM] [SKIP | NOSKIP]	
Function:	Transfers the current array row into local variables.	
Con Codes:	**0**	Command was successfully executed.
	4	Array is empty / no matching rows found for SETVIEW.
	8	Array was not found.
	16	Syntax error occurred.
	20	If SKIP was specified (or was the default and the last row of the table was read, a return code of 20 will be returned.
Examples:	IMFEXEC ARRAY GET DASDSTATS "IMFEXEC ARRAY GET DASDSTATS"	
Syntax:	**ARRAY INFO** name	
Function:	Provides information about an array.	
Cond Codes:	**0**	Command was successfully executed.
	8	Array was not found.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY INFO DASDSTAT "IMFEXEC ARRAY INFO DASDSTAT"	
Syntax:	**ARRAY INSERT** name [POSITION(HERE | FIRST | LAST)]	
Function:	Inserts a new row into an array.	
Cond Codes:	**0**	Command was successfully executed.
	8	Array was not found.
	12	Array was not in UPDATE access.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY INSERT DASDSTATS "IMFEXEC ARRAY INSERT DASDSTATS"	
Syntax:	**ARRAY LIST** [KEPT]	
Function:	Provides information about saved or disconnected arrays (when kept).	
Cond Codes:	**0**	Command was successfully executed.
	16	Syntax error occurred.

BMC Software, Inc., Confidential and Proprietary Information

Table 11-1 IMFEXEC ARRAY Commands and Examples (Part 4 of 5)

Function	Command and Format	
Examples:	IMFEXEC ARRAY LIST KEPT "IMFEXEC ARRAY LIST KEPT"	
Syntax:	**ARRAY PUT** name	
Function:	Sets the current array row from local variables.	
Cond Codes:	**0**	Command was successfully executed.
	8	Invalid array token specified.
	12	Array not in UPDATE access.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY PUT DASDSTATS "IMFEXEC ARRAY PUT DASDSTATS"	
Syntax:	**ARRAY SAVE** name	
Function:	Provides checkpoint processing of the contents of an array to disk.	
Cond Codes:	**0**	Command was successfully executed.
	8	Array was not found or an I/O error occurred when writing to disk.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY SAVE DASDSTATS "IMFEXEC ARRAY SAVE DASDSTATS"	
Syntax:	**ARRAY SET** name [TRIM | NOTRIM]	
Function:	Transfers an array into REXX TSO/E variables.	
Cond Codes:	**0**	Command was successfully executed.
	4	Array is empty.
	8	Array was not found or a column name wider than 26 characters was found.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY SET DASDSTAT “IMFEXEC ARRAY SET DASDSTAT”	
Syntax:	**ARRAY SETVIEW** name [CRITERIA()] [FUNCTION(DELETE | APPEND)]	
Function:	Limits array access to rows matching certain criteria.	

BMC Software, Inc., Confidential and Proprietary Information

Table 11-1 IMFEXEC ARRAY Commands and Examples (Part 5 of 5)

Function	Command and Format	
Cond Codes:	**0**	Command was successfully executed.
	4	Array was not found.
	12	Criteria parsing error occurred.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY SETVIEW DASDSTAT CRITERIA(’STAT,,,=,”ACTIVE’”) “IMFEXEC ARRAY SETVIEW DASDSTAT CRITERIA(’STAT,,,=,”ACTIVE’”)”	
Syntax:	**ARRAY SORT** name [CRITERIA(’colname,[start],[length],[order]’)] [DELETE]	
Function:	Sorts an array according to one or more criteria.	
Cond Codes:	**0**	Command was successfully executed.
	4	Criteria parsing error occurred (invalid or missing criteria definition).
	8	Array was not found.
	16	Syntax error occurred.
Examples:	IMFEXEC ARRAY SORT MYTEST CRITERIA('ROW1,,,A') "IMFEXEC ARRAY SORT MYTEST CRITERIA('ROW1,,,A')"	

BMC Software, Inc., Confidential and Proprietary Information

MAINVIEW API Command Summary

The MAINVIEW API provides for a one-way data exchange between MAINVIEW-based products such as MAINVIEW for CICS, MAINVIEW for OS/390, MAINVIEW for IMS (and others) and AutoOPERATOR REXX or CLIST EXECs. Using the API, AutoOPERATOR EXECs can explicitly request data from a MAINVIEW product.

MAINVIEW API Commands and Examples

Table 12-1 MAINVIEW API Commands and Examples (Part 1 of 2)

Functions	Command and Format
Syntax:	**MAINVIEW | MV CONNECT** var-name [MSG | NOMSG]
Function:	Causes a new channel to be used in subsequent channel requests. A channel must be connected before any MAINVIEW data can be requested using the low-level API.
Cond Codes:	See “MAINVIEW API Return Codes” on page 12-3
Examples:	IMFEXEC MAINVIEW CONNECT JOBCHANNEL “IMFEXEC MAINVIEW CONNECT JOBCHANNEL”
Syntax:	**MAINVIEW | MV CONTEXT** PRODUCT(product name) TARGET(target identifier) [SERVER(server name)] [WAIT(n)] CHANNEL(channelname)
Function:	Connects a channel with a specified context or target.

BMC Software, Inc., Confidential and Proprietary Information

Table 12-1 MAINVIEW API Commands and Examples (Part 2 of 2)

Functions	Command and Format
Cond Codes:	See “MAINVIEW API Return Codes” on page 12-3
Examples:	IMFEXEC MAINVIEW CONTEXT PRODUCT(MVMVS) TARGET(SJSB) CHANNEL(&JOBCHANNEL) “IMFEXEC MAINVIEW CONTEXT PRODUCT(MVMVS) TARGET(SJSB) CHANNEL(“JOBCHANNEL”)”
Syntax:	**MAINVIEW | MV GETDATA** ARRAY(arrayname) [START(n)] [COUNT(n)] [REFRESH] [MSG | NOMSG] CHANNEL(channel-token)
Function:	Returns collected view data.
Cond Codes:	See “MAINVIEW API Return Codes” on page 12-3
Examples:	IMFEXEC MAINVIEW GETDATA CHANNEL(&DASDCHANNEL) ARRAY(DASDSTAT) START(1) COUNT(20) REFRESH "IMFEXEC MAINVIEW GETDATA CHANNEL("DASDCHANNEL") ARRAY(DASDSTAT) START(1) COUNT(20) REFRESH"
Syntax:	**MAINVIEW | MV RELEASE** CHANNEL(channelname)
Function:	Releases all resources associated with a channel.
Cond Codes:	See “MAINVIEW API Return Codes” on page 12-3
Examples:	IMFEXEC MAINVIEW RELEASE CHANNEL(&JOBCHANNEL) "IMFEXEC MAINVIEW RELEASE CHANNEL("JOBCHANNEL")"
Syntax:	**MAINVIEW | MV TRACE** ON | OFF
Function:	Turns TRACE information on or off.
Cond Codes:	See “MAINVIEW API Return Codes” on page 12-3
Examples:	IMFEXEC MAINVIEW TRACE ON “IMFEXEC MAINVIEW TRACE ON”
Syntax:	**MAINVIEW | MV VIEW** NAME(viewname) [STEM(stemname)] [DD(ddname)] [PARMS(parm1... parm2... parm*n*)] CHANNEL(channelname)
Function:	Identifies the view to be used for accessing data.
Cond Codes:	See “MAINVIEW API Return Codes” on page 12-3.
Examples:	IMFEXEC MAINVIEW VIEW NAME(JOVER) CHANNEL(&JOBCHANNEL) "IMFEXEC MAINVIEW VIEW NAME(JOVER) CHANNEL("JOBCHANNEL")"

BMC Software, Inc., Confidential and Proprietary Information

MAINVIEW API Return Codes

Return Codes	Description
0	All requested data was successfully retrieved.
4	The specified array could not be built; it already exists.
8	The requested data could not be retrieved. Examine the accompanying error messages for details. If NOMSG was specified on CONNECT, display the contents of LINExxxx. This return code may also indicate that the START() keyword specified a value that was higher than the number of available records (in which case no records can be returned).
12	The specified channel could not be found.
16	Syntax error was detected or invalid parameters supplied.
20	Internal error occurred.

BMC Software, Inc., Confidential and Proprietary Information

BMC Software, Inc., Confidential and Proprietary Information

END USER LICENSE AGREEMENT
NOTICE

BY OPENING THE PACKAGE, INSTALLING, PRESSING "AGREE" OR "YES" OR USING THE PRODUCT, THE ENTITY OR INDIVIDUAL ENTERING INTO THIS AGREEMENT AGREES TO BE BOUND BY THE FOLLOWING TERMS. IF YOU DO NOT AGREE WITH ANY OF THESE TERMS, DO NOT INSTALL OR USE THE PRODUCT, PROMPTLY RETURN THE PRODUCT TO BMC OR YOUR BMC RESELLER, AND IF YOU ACQUIRED THE LICENSE WITHIN 30 DAYS OF THE DATE OF YOUR ORDER CONTACT BMC OR YOUR BMC RESELLER FOR A REFUND OF LICENSE FEES PAID. IF YOU REJECT THIS AGREEMENT, YOU WILL NOT ACQUIRE ANY LICENSE TO USE THE PRODUCT.

This Agreement ("**Agreement**") is between the entity or individual entering into this Agreement ("You") and BMC Software Distribution, Inc., a Delaware corporation located at 2101 CityWest Blvd., Houston, Texas, 77042, USA or its affiliated local licensing entity ("BMC"). "You" includes you and your Affiliates. "Affiliate" is defined as an entity which controls, is controlled by or shares common control with a party. THIS AGREEMENT WILL APPLY TO THE PRODUCT, UNLESS (1) YOU AGREED TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT, IN WHICH CASE THAT WEB BASED LICENSE AGREEMENT GOVERNS THE USE OF THE PRODUCT, OR (2) IF YOU DID NOT AGREE TO A WEB BASED LICENSE AGREEMENT WITH BMC WHEN ORDERING THE PRODUCT AND YOU HAVE A WRITTEN LICENSE AGREEMENT WITH BMC, THEN THAT WRITTEN AGREEMENT GOVERNS THE USE OF THE PRODUCT. THE ELECTRONIC AGREEMENT PROVIDED WITH THE PRODUCT AS PART OF THE INSTALLATION OF THE PRODUCT WILL NOT APPLY. In addition to the restrictions imposed under this Agreement, any other usage restrictions contained in the Product installation instructions or release notes shall apply to Your use of the Product.

PRODUCT AND CAPACITY. "**Software**" means the object code version of the computer programs provided, via delivery or electronic transmission, to You. Software includes computer files, enhancements, maintenance modifications, upgrades, updates, bug fixes, and error corrections.

"**Documentation**" means all written or graphical material provided by BMC in any medium, including any technical specifications, relating to the functionality or operation of the Software.

"**Product**" means the Software and Documentation.

"**License Capacity**" means the licensed capacity for the Software with the pricing and other license defining terms, including capacity restrictions, such as tier limit, total allowed users, gigabyte limit, quantity of Software, and/or other capacity limitations regarding the Software. For licenses based on the power of a computer, You agree to use BMC's current computer classification scheme, which is available at http://www.bmc.com or can be provided to You upon request.

ACCEPTANCE. The Product is deemed accepted by You, on the date that You received the Product from BMC.

LICENSE. Subject to the terms of this Agreement, as well as Your payment of applicable fees, BMC grants You a non-exclusive, non-transferable, perpetual (unless a term license is provided on an order) license for each copy of the Software, up to the License Capacity, to do the following:

(a) install the Software on Your owned or leased hardware located at a facility owned or controlled by You in the country where You acquired the license;

(b) operate the Software solely for processing Your own data in Your business operations; and

(c) make one copy of the Software for backup and archival purposes only (collectively a "**License**").

If the Software is designed by BMC to permit you to modify such Software, then you agree to only use such modifications or new software programs for Your internal purposes or otherwise consistent with the License. BMC grants You a license to use the Documentation solely for Your internal use in Your operations.

LICENSE UPGRADES. You may expand the scope of the License Capacity only pursuant to a separate agreement with BMC for such expanded usage and Your payment of applicable fees. There is no additional warranty period or free support period for license upgrades.

RESTRICTIONS: You agree to **NOT**:
(a) disassemble, reverse engineer, decompile or otherwise attempt to derive any Software from executable code;

(b) distribute or provide the Software to any third party (including without limitation, use in a service bureau, outsourcing environment, or processing the data of third parties, or for rental, lease, or sublicense); or

(c) provide a third party with the results of any functional evaluation or benchmarking or performance tests, without BMC's prior written approval, unless prohibited by local law.

TRIAL LICENSE. If, as part of the ordering process, the Product is provided on a trial basis, then these terms apply: (i) this license consists solely of a non-exclusive, non-transferable evaluation license to operate the Software for the period of time specified from BMC or, if not specified, a 30 day time period ("**Trial Period**") only for evaluating whether You desire to acquire a capacity-based license to the Product for a fee; and (ii) Your use of the Product is on an AS IS basis without any warranty, and **BMC, ITS AFFILIATES AND RESELLERS, AND LICENSORS DISCLAIM ANY AND ALL WARRANTIES (INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT) AND HAVE NO LIABILITY WHATSOEVER RESULTING FROM THE USE OF THIS PRODUCT UNDER THIS TRIAL LICENSE ("Trial License")**. BMC may terminate for its convenience a Trial License upon notice to You. When the Trial Period ends, Your right to use this Product automatically expires. If You want to continue Your use of the Product beyond the Trial Period, contact BMC to acquire a capacity-based license to the Product for a fee.

TERMINATION. This Agreement shall immediately terminate if You breach any of its terms. Upon termination, for any reason, You must uninstall the Software, and either certify the destruction of the Product or return it to BMC.

OWNERSHIP OF THE PRODUCT. BMC or its Affiliates or licensors retain all right, title and interest to and in the BMC Product and all intellectual property, informational, industrial property and proprietary rights therein. BMC neither grants nor otherwise transfers any rights of ownership in the BMC Product to You. BMC Products are protected by applicable copyright, trade secret, and industrial and intellectual property laws. BMC reserves any rights not expressly granted to You herein.

CONFIDENTIAL AND PROPRIETARY INFORMATION. The BMC Products are and contain valuable confidential information of BMC ("**Confidential Information**"). Confidential Information means non-public technical and non-technical information relating to the BMC Products and Support, including, without limitation, trade secret and proprietary information, and the structure and organization of the Software. You may not disclose the Confidential Information to third parties. You agree to use all reasonable efforts to prevent the unauthorized use, copying, publication or dissemination of the Product.

WARRANTY. Except for a Trial License, BMC warrants that the Software will perform in substantial accordance with the Documentation for a period of one year from the date of the order. This warranty shall not apply to any problems caused by software or hardware not supplied by BMC or to any misuse of the Software.

EXCLUSIVE REMEDY. BMC's entire liability, and Your exclusive remedy, for any defect in the Software during the warranty period or breach of the warranty above shall be limited to the following: BMC shall use reasonable efforts to remedy defects covered by the warranty or replace the defective Software within a reasonable period of time, or if BMC cannot remedy or replace such defective copy of the Software, then BMC shall refund the amount paid by You for the License for that Software. BMC's obligations in this section are conditioned upon Your providing BMC prompt access to the affected Software and full cooperation in resolving the claim.

DISCLAIMER. EXCEPT FOR THE EXPRESS WARRANTIES ABOVE, THE PRODUCT IS PROVIDED "AS IS." BMC, ITS AFFILIATES AND LICENSORS SPECIFICALLY DISCLAIM ALL OTHER WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. BMC DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS CAN BE CORRECTED.

DISCLAIMER OF DAMAGES. IN NO EVENT IS BMC, ITS AFFILIATES OR LICENSORS LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RELATING TO OR ARISING OUT OF THIS AGREEMENT, SUPPORT, AND/OR THE PRODUCT (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, LOST COMPUTER USAGE TIME, AND DAMAGE OR LOSS OF USE OF DATA), EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND IRRESPECTIVE OF ANY NEGLIGENCE OF BMC OR WHETHER SUCH DAMAGES RESULT FROM A CLAIM ARISING UNDER TORT OR CONTRACT LAW.

LIMITS ON LIABILITY. BMC'S AGGREGATE LIABILITY FOR DAMAGES IS LIMITED TO THE AMOUNT PAID BY YOU FOR THE LICENSE TO THE PRODUCT.

SUPPORT. If Your order includes support for the Software, then BMC agrees to provide support (24 hours a day/7 days a week) ("**Support**"). You will be automatically re-enrolled in Support on an annual basis unless BMC receives notice of termination from You as provided below. There is a free support period during the one year warranty period.

(a) **Support Terms**. BMC agrees to make commercially reasonable efforts to provide the following Support: (i) For malfunctions of supported versions of the Software, BMC provides bug fixes, patches or workarounds in order to cause that copy of the Software to operate in substantial conformity with its then-current operating specifications; and (ii) BMC provides new releases or versions, so long as such new releases or versions are furnished by BMC to all other enrolled Support customers without additional charge. BMC may refuse to provide Support for any versions or releases of the Software other than the most recent version or release of such Software made available by BMC. Either party may terminate Your enrollment in Support upon providing notice to the other at least 30 days prior to the next applicable Support anniversary date. If You re-enroll in Support, BMC may charge You a reinstatement fee of 1.5 times what You would have paid if You were enrolled in Support during that time period.

(b) **Fees**. The annual fee for Support is 20% of the Software's list price less the applicable discount or a flat capacity based annual fee. BMC may change its prices for the Software and/or Support upon at least 30 days notice prior to Your support anniversary date.

VERIFICATION. If requested by BMC, You agree to deliver to BMC periodic written reports, whether generated manually or electronically, detailing Your use of the Software in accordance with this Agreement, including, without limitation, the License Capacity. BMC may, at its expense, audit Your use of the Software to confirm Your compliance with the Agreement. If an audit reveals that You have underpaid fees, You agree to pay such underpaid fees. If the underpaid fees exceed 5% of the fees paid, then You agree to also pay BMC's reasonable costs of conducting the audit.

EXPORT CONTROLS. You agree not to import, export, re-export, or transfer, directly or indirectly, any part of the Product or any underlying information or technology except in full compliance with all United States, foreign and other applicable laws and regulations.

GOVERNING LAW. This Agreement is governed by the substantive laws in force, without regard to conflict of laws principles: (a) in the State of New York, if you acquired the License in the United States, Puerto Rico, or any country in Central or South America; (b) in the Province of Ontario, if you acquired the License in Canada (subsections (a) and (b) collectively referred to as the "**Americas Region**"); (c) in Singapore, if you acquired the License in Japan, South Korea, Peoples Republic of China, Special Administrative Region of Hong Kong, Republic of China, Philippines, Indonesia, Malaysia, Singapore, India, Australia, New Zealand, or Thailand (collectively, "**Asia Pacific Region**"); or (d) in the Netherlands, if you acquired the License in any other country not described above. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed in its entirety.

ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL OR INTERNATIONAL, AS APPLICABLE, RULES OF THE AMERICAN ARBITRATION ASSOCIATION. IF THE DISPUTE IS INITIATED IN A COUNTRY IN THE ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN SINGAPORE, SINGAPORE UNDER THE CURRENT UNCITRAL ARBITRATION RULES. IF THE DISPUTE IS INITIATED IN A COUNTRY OUTSIDE OF THE AMERICAS REGION OR ASIA PACIFIC REGION, THE ARBITRATION SHALL BE HELD IN AMSTERDAM, NETHERLANDS UNDER THE CURRENT UNCITRAL ARBITRATION RULES. THE COSTS OF THE ARBITRATION SHALL BE BORNE EQUALLY PENDING THE ARBITRATOR'S AWARD. THE AWARD RENDERED SHALL BE FINAL AND BINDING UPON THE PARTIES AND SHALL NOT BE SUBJECT TO APPEAL TO ANY COURT, AND MAY BE ENFORCED IN ANY COURT OF COMPETENT JURISDICTION. NOTHING IN THIS AGREEMENT SHALL BE DEEMED AS PREVENTING EITHER PARTY FROM SEEKING INJUNCTIVE RELIEF FROM ANY COURT HAVING JURISDICTION OVER THE PARTIES AND THE SUBJECT MATTER OF THE DISPUTE AS NECESSARY TO PROTECT EITHER PARTY'S CONFIDENTIAL INFORMATION, OWNERSHIP, OR ANY OTHER PROPRIETARY RIGHTS. ALL ARBITRATION PROCEEDINGS SHALL BE CONDUCTED IN CONFIDENCE, AND THE PARTY PREVAILING IN ARBITRATION SHALL BE ENTITLED TO RECOVER ITS REASONABLE ATTORNEYS' FEES AND NECESSARY COSTS INCURRED RELATED THERETO FROM THE OTHER PARTY.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software under this Agreement is "commercial computer software" as that term is described in 48 C.F.R. 252.227-7014(a)(1). If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulations ("**FAR**") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("**DOD**"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of this Agreement as specified in 48 C.F.R. 227.7202 of the DOD FAR Supplement and its successors.

MISCELLANEOUS TERMS. You agree to pay BMC all amounts owed no later than 30 days from the date of the applicable invoice, unless otherwise provided on the order for the License to the Products. You will pay, or reimburse BMC, for taxes of any kind, including sales, use, duty, tariffs, customs, withholding, property, value-added (VAT), and other similar federal, state or local taxes (other than taxes based on BMC's net income) imposed in connection with the Product and/or the Support. This Agreement constitutes the entire agreement between You and BMC and supersedes any prior or contemporaneous negotiations or agreements, whether oral, written or displayed electronically, concerning the Product and related subject matter. No modification or waiver of any provision hereof will be effective unless made in a writing signed by both BMC and You. You may not assign or transfer this Agreement or a License to a third party without BMC's prior written consent. Should any provision of this Agreement be invalid or unenforceable, the remainder of the provisions will remain in effect. The parties have agreed that this Agreement and the documents related thereto be drawn up in the English language. Les parties exigent que la présente convention ainsi que les documents qui s'y rattachent soient rédigés en anglais.

Notes

25691