

IBM Operations Manager for z/VM



# Operations Manager Administration Guide

*Version 1 Release 5*



IBM Operations Manager for z/VM



# Operations Manager Administration Guide

*Version 1 Release 5*

**Note:**

Before using this information and the products that it supports, read the information in “Notices” on page 191.

This edition applies to Version 1 Release 5 of IBM Operations Manager for z/VM (5697-J10) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC18-9347-17.

© Copyright Rocket Software Inc., 1996 – 2015. All Rights Reserved.

---

# Contents

<b>Figures</b> . . . . .	<b>xiii</b>
<b>Tables</b> . . . . .	<b>xv</b>
<b>About this information</b> . . . . .	<b>xvii</b>
Who should read this book . . . . .	xvii
Service updates and support information . . . . .	xvii
How to read syntax diagrams. . . . .	xvii
How to look up message explanations. . . . .	xviii
Searching for messages on the web . . . . .	xix
Where to find information . . . . .	xix
How to send your comments . . . . .	xix
<b>Summary of changes</b> . . . . .	<b>xxi</b>
Version 1.5 . . . . .	xxi
<b>Chapter 1. Introduction to Operations Manager</b> . . . . .	<b>1</b>
What does Operations Manager do? . . . . .	1
Benefits of using Operations Manager . . . . .	1
Operations Manager components . . . . .	2
Message processing component . . . . .	2
Action processing component. . . . .	2
User interaction component . . . . .	2
Automated session component . . . . .	2
Component interaction . . . . .	3
Prerequisites . . . . .	3
Hardware requirements. . . . .	3
Software requirements . . . . .	3
<b>Chapter 2. Configuring Operations Manager</b> . . . . .	<b>5</b>
Step 1. Update OPMGRM1 . . . . .	5
Updates to PROFILE EXEC . . . . .	5
Updates to CP directory . . . . .	6
Step 2. Update Action Processing Servers. . . . .	7
Updates to PROFILE EXEC . . . . .	7
Updates to CP directory . . . . .	8
Step 3. Update End User and Administrator IDs . . . . .	9
Step 4. Update IDs monitored by Operations Manager . . . . .	9
Updating the CP directory . . . . .	10
Updating the PROFILE EXEC . . . . .	10
Verifying the console data is being sent to Operations Manager . . . . .	11
Step 5. (Optional) Configure an external security mechanism . . . . .	11
Setup example for IBM Resource Access Control Facility (RACF) . . . . .	11
FACILITY class profiles . . . . .	13
<b>Chapter 3. Creating a startup configuration file</b> . . . . .	<b>21</b>
Entering commands . . . . .	21
Command rules . . . . .	21
Where to enter commands . . . . .	22
Using the sample configuration file . . . . .	23
Copying the sample configuration file . . . . .	23
Adding comments . . . . .	24
Assigning authority . . . . .	25
Control authority . . . . .	25

Configuration authority . . . . .	25
Console access . . . . .	25
Spool access . . . . .	26
TCP/IP application interaction . . . . .	26
Authority assignments with AUTH and external security . . . . .	26
Defining action processing servers. . . . .	27
Defining system options . . . . .	27
Modifying the startup configuration file . . . . .	27
Start the Operations Manager service machines . . . . .	28
Verify the installation and configuration. . . . .	28
Managing the configuration file . . . . .	28

## **Chapter 4. Syntax for System Management Commands. . . . . 31**

Purpose of system management commands . . . . .	31
GOMMAIN . . . . .	31
Authorization . . . . .	31
Purpose . . . . .	31
Operands . . . . .	31
Usage notes . . . . .	32
Return codes . . . . .	32
GOMCMD . . . . .	32
Authorization . . . . .	32
Purpose . . . . .	33
Operands . . . . .	33
Usage notes . . . . .	33
Examples . . . . .	33
Return codes . . . . .	33
GOMSVM . . . . .	34
Authorization . . . . .	34
Purpose . . . . .	34
Operands . . . . .	34
Usage notes . . . . .	34
Return codes . . . . .	34
GOMTDM. . . . .	35
Authorization . . . . .	35
Purpose . . . . .	35
Operands . . . . .	36
Usage notes . . . . .	37
Return codes . . . . .	37

## **Chapter 5. Syntax for Control Commands. . . . . 39**

Purpose of control commands . . . . .	39
Authorization for control commands . . . . .	39
AUTH . . . . .	39
Authorization . . . . .	39
Purpose . . . . .	39
Operands . . . . .	40
Operand value lengths . . . . .	40
Usage notes . . . . .	40
Return codes . . . . .	41
CMDTCPA . . . . .	41
Authority . . . . .	41
Purpose . . . . .	42
Operands . . . . .	42
Usage notes . . . . .	42
CMS. . . . .	42
Authorization . . . . .	43
Purpose . . . . .	43
Operands . . . . .	43
Operand value lengths . . . . .	43

Usage notes . . . . .	43
Return codes . . . . .	43
CONFIG . . . . .	43
Authorization . . . . .	44
Purpose . . . . .	44
Operands . . . . .	44
Operand value lengths . . . . .	45
Usage notes . . . . .	45
Return codes . . . . .	45
CP . . . . .	45
Authorization . . . . .	45
Purpose . . . . .	46
Operands . . . . .	46
Operand value lengths . . . . .	46
Usage notes . . . . .	46
Return codes . . . . .	46
DEFDPS . . . . .	46
Authorization . . . . .	46
Purpose . . . . .	46
Operands . . . . .	46
Operand value lengths . . . . .	47
Usage notes . . . . .	47
Return codes . . . . .	47
DEFOPTN . . . . .	48
Authorization . . . . .	48
Purpose . . . . .	48
Operands . . . . .	48
Operand value lengths . . . . .	50
Usage notes . . . . .	50
Return codes . . . . .	51
DEFSRVM . . . . .	51
Authorization . . . . .	52
Purpose . . . . .	52
Operands . . . . .	52
Operand value lengths . . . . .	52
Usage notes . . . . .	52
Return codes . . . . .	53
DEFTCPA . . . . .	53
Authorization . . . . .	53
Purpose . . . . .	53
Operands . . . . .	53
Operand value lengths . . . . .	54
Usage notes . . . . .	54
Return codes . . . . .	55
DEFVIEW . . . . .	55
Authorization . . . . .	56
Purpose . . . . .	56
Operands . . . . .	56
Operand value lengths . . . . .	56
Usage notes . . . . .	56
Return codes . . . . .	57
DELDPS . . . . .	57
Authorization . . . . .	57
Purpose . . . . .	57
Operands . . . . .	57
Operand value lengths . . . . .	57
Usage notes . . . . .	57
Return codes . . . . .	57
DELSRVM . . . . .	58
Authorization . . . . .	58
Purpose . . . . .	58

Operands . . . . .	58
Operand value lengths . . . . .	58
Usage notes . . . . .	58
Return codes . . . . .	59
DELTCPA . . . . .	59
Authorization . . . . .	59
Purpose . . . . .	59
Operands . . . . .	59
Operand value lengths . . . . .	59
Usage notes . . . . .	59
Return codes . . . . .	60
DELVIEW . . . . .	60
Authorization . . . . .	60
Purpose . . . . .	60
Operands . . . . .	60
Operand value lengths . . . . .	60
Usage notes . . . . .	60
Return codes . . . . .	61
QUIT . . . . .	61
Authorization . . . . .	61
Purpose . . . . .	61
Usage notes . . . . .	61
Return codes . . . . .	61
REVOKE . . . . .	61
Authorization . . . . .	62
Purpose . . . . .	62
Operands . . . . .	62
Operand value lengths . . . . .	62
Usage notes . . . . .	62
Return codes . . . . .	63
SHUTDOWN . . . . .	63
Authorization . . . . .	63
Purpose . . . . .	63
Usage notes . . . . .	63
Return codes . . . . .	63
STATUS . . . . .	64
Authorization . . . . .	64
Purpose . . . . .	64
Operands . . . . .	64
Operand value lengths . . . . .	65
Examples . . . . .	65
Usage notes . . . . .	66
Return codes . . . . .	66
STOP . . . . .	67
Authorization . . . . .	67
Purpose . . . . .	67
Usage notes . . . . .	67
Return codes . . . . .	67
<b>Chapter 6. Syntax for Configuration Commands . . . . .</b>	<b>69</b>
Purpose of configuration commands . . . . .	69
Authorization for configuration commands . . . . .	69
CONFIG . . . . .	69
Authorization . . . . .	70
Purpose . . . . .	70
Operands . . . . .	70
Operand value lengths . . . . .	71
Usage notes . . . . .	71
Return codes . . . . .	71
DEFACTN . . . . .	71
Authorization . . . . .	72



Purpose . . . . .	72
Operands . . . . .	73
Operand value lengths . . . . .	74
Usage notes . . . . .	74
Return codes . . . . .	78
DEFDATE . . . . .	78
Authorization . . . . .	78
Purpose . . . . .	78
Operands . . . . .	78
Operand value lengths . . . . .	79
Usage notes . . . . .	79
Return codes . . . . .	79
DEFEMON . . . . .	79
Authorization . . . . .	80
Purpose . . . . .	80
Operands . . . . .	80
Operand value lengths . . . . .	81
Usage notes . . . . .	82
Return codes . . . . .	82
DEFIMON . . . . .	82
Authorization . . . . .	83
Purpose . . . . .	83
Operands . . . . .	83
Operand value lengths . . . . .	84
Usage notes . . . . .	85
Return codes . . . . .	85
DEFGROUP . . . . .	85
Authorization . . . . .	85
Purpose . . . . .	86
Operands . . . . .	86
Operand value lengths . . . . .	86
Usage notes . . . . .	87
Return codes . . . . .	87
DEFMMON . . . . .	87
Authorization . . . . .	87
Purpose . . . . .	88
Operands . . . . .	88
Operand value lengths . . . . .	89
Usage notes . . . . .	90
Return codes . . . . .	90
DEFPMON . . . . .	90
Authorization . . . . .	91
Purpose . . . . .	91
Operands . . . . .	91
Operand value lengths . . . . .	92
Usage notes . . . . .	93
Return codes . . . . .	93
DEFRULE . . . . .	94
Authorization . . . . .	94
Purpose . . . . .	94
Operands . . . . .	94
Operand value lengths . . . . .	96
Usage notes . . . . .	97
Return Codes . . . . .	98
DEFSCHD . . . . .	98
Authorization . . . . .	99
Purpose . . . . .	99
Operands . . . . .	99
Operand value lengths . . . . .	101
Usage notes . . . . .	101
Return codes . . . . .	102

DEFMON . . . . .	102
Authorization . . . . .	102
Purpose . . . . .	102
Operands. . . . .	103
Operand value lengths . . . . .	104
Usage notes . . . . .	104
Return codes . . . . .	105
DEFACTN . . . . .	105
Authorization . . . . .	105
Purpose . . . . .	105
Operands. . . . .	105
Operand value lengths . . . . .	106
Usage notes . . . . .	106
Return codes . . . . .	106
DELDATE . . . . .	106
Authorization . . . . .	106
Purpose . . . . .	106
Operands. . . . .	106
Operand value lengths . . . . .	106
Usage notes . . . . .	107
Return codes . . . . .	107
DELEMON . . . . .	107
Authorization . . . . .	107
Purpose . . . . .	107
Operands. . . . .	107
Operand value lengths . . . . .	107
Usage notes . . . . .	107
Return codes . . . . .	108
DELGROUP . . . . .	108
Authorization . . . . .	108
Purpose . . . . .	108
Operands. . . . .	108
Operand value lengths . . . . .	108
Usage notes . . . . .	109
Return codes . . . . .	109
DELIMON . . . . .	109
Authorization . . . . .	109
Purpose . . . . .	109
Operands. . . . .	109
Operand value lengths . . . . .	109
Usage notes . . . . .	110
Return codes . . . . .	110
DELMON . . . . .	110
Authorization . . . . .	110
Purpose . . . . .	110
Operands. . . . .	110
Operand value lengths . . . . .	110
Usage notes . . . . .	110
Return codes . . . . .	111
DELPMON . . . . .	111
Authorization . . . . .	111
Purpose . . . . .	111
Operands. . . . .	111
Operand value lengths . . . . .	111
Usage notes . . . . .	111
Return codes . . . . .	111
DELRULE . . . . .	112
Authorization . . . . .	112
Purpose . . . . .	112
Operands. . . . .	112
Operand value lengths . . . . .	112

Usage notes . . . . .	112
Return Codes . . . . .	112
DELSCHD . . . . .	112
Authorization . . . . .	113
Purpose . . . . .	113
Operands. . . . .	113
Operand value lengths . . . . .	113
Usage notes . . . . .	113
Return codes . . . . .	113
DELSMON . . . . .	113
Authorization . . . . .	113
Purpose . . . . .	113
Operands. . . . .	114
Operand value lengths . . . . .	114
Usage notes . . . . .	114
Return codes . . . . .	114
RESUME . . . . .	114
Authorization . . . . .	114
Purpose . . . . .	114
Operands. . . . .	115
Operand value lengths . . . . .	115
Usage notes . . . . .	115
Return codes . . . . .	116
RUN . . . . .	116
Authorization . . . . .	116
Purpose . . . . .	116
Operands. . . . .	116
Operand value lengths . . . . .	117
Usage notes . . . . .	117
Return codes . . . . .	117
STATUS . . . . .	118
Authorization . . . . .	118
Purpose . . . . .	118
Operands. . . . .	118
Operand value lengths . . . . .	119
Examples. . . . .	119
Usage notes . . . . .	120
Return codes . . . . .	120
SUSPEND . . . . .	121
Authorization . . . . .	121
Purpose . . . . .	121
Operands. . . . .	121
Operand value lengths . . . . .	122
Usage notes . . . . .	122
Return codes . . . . .	122

**Chapter 7. Syntax for Console, Log, and Spool Access Commands . . . . . 123**

Purpose of console, log, and spool access commands . . . . .	123
Authorization for console, log, and spool access commands . . . . .	123
Console data spaces . . . . .	123
ALTRCON . . . . .	124
Authority. . . . .	124
Purpose . . . . .	124
Operands. . . . .	124
Operand value lengths . . . . .	125
Usage notes . . . . .	125
Return codes . . . . .	126
LOGTEXT . . . . .	126
Authorization . . . . .	126
Purpose . . . . .	126
Operands. . . . .	126

Operand value lengths . . . . .	126
Usage notes . . . . .	126
Return codes . . . . .	126
VIEWCON . . . . .	127
Authorization . . . . .	127
Purpose . . . . .	127
Operands. . . . .	127
Operand value lengths . . . . .	128
Usage notes . . . . .	128
Return codes . . . . .	131
VIEWLOG . . . . .	131
Authorization . . . . .	132
Purpose . . . . .	132
Operands. . . . .	132
Operand value lengths . . . . .	133
Usage notes . . . . .	133
Return codes . . . . .	135
VIEWSP. . . . .	136
Authorization . . . . .	136
Purpose . . . . .	136
Operands. . . . .	136
Operand value lengths . . . . .	137
Usage notes . . . . .	137
Return codes . . . . .	142
<b>Appendix A. Sample files . . . . .</b>	<b>143</b>
<b>Appendix B. Accounting and Monitoring Information . . . . .</b>	<b>145</b>
Accounting record format . . . . .	145
Monitoring record format . . . . .	146
<b>Appendix C. Log File Management . . . . .</b>	<b>147</b>
Customize log file management . . . . .	147
<b>Appendix D. Operations Manager and TCP/IP interaction . . . . .</b>	<b>149</b>
Collection and processing of syslog data . . . . .	149
Translation tables . . . . .	150
Collection and processing of remote file input . . . . .	151
Send and view file content . . . . .	151
Send and view multiple files . . . . .	152
GOMRSIF . . . . .	152
Authorization . . . . .	152
Purpose . . . . .	153
Operands. . . . .	153
Usage notes . . . . .	153
Return codes . . . . .	153
<b>Appendix E. Testing Operations Manager installation and configuration . . . . .</b>	<b>155</b>
Checking the status of OPMGRM1 . . . . .	155
Verifying consoles are monitored and viewable . . . . .	155
Creating a schedule and corresponding action . . . . .	156
Verifying spool files are viewable. . . . .	156
<b>Appendix F. Application and settings interference . . . . .</b>	<b>159</b>
<b>Appendix G. GOMGLBL facility. . . . .</b>	<b>161</b>
Preparing GOMGLBL . . . . .	161
GOMGLBL . . . . .	162
Operands. . . . .	162

Usage notes . . . . .	163
Restrictions . . . . .	163
Return codes . . . . .	163
<b>Appendix H. Messages and Codes . . . . .</b>	<b>165</b>
<b>Notices . . . . .</b>	<b>191</b>
Trademarks . . . . .	193
<b>Index . . . . .</b>	<b>195</b>



---

## Figures

1. Operations Manager component interaction	3	33. DEFIMON command syntax . . . . .	83
2. GOMMAIN command syntax . . . . .	31	34. DEFGROUP command syntax . . . . .	85
3. GOMCMD command syntax . . . . .	32	35. DEFMMON command syntax . . . . .	87
4. GOMSVM command syntax . . . . .	34	36. DEFPMON command syntax. . . . .	91
5. GOMTDM command syntax . . . . .	35	37. DEFROLE command syntax . . . . .	94
6. AUTH command syntax . . . . .	39	38. DEFSCHD command syntax . . . . .	99
7. CMDTCPA command syntax. . . . .	41	39. DEFSMON command syntax . . . . .	102
8. CMS command syntax . . . . .	43	40. DELACTN command syntax . . . . .	105
9. CONFIG command syntax . . . . .	44	41. DELDATE command syntax . . . . .	106
10. CP command syntax . . . . .	45	42. DELEMON command syntax . . . . .	107
11. DEFDSPS command syntax . . . . .	46	43. DELGROUP command syntax . . . . .	108
12. DEFOPTN command syntax . . . . .	48	44. DELIMON command syntax . . . . .	109
13. DEFSRVM command syntax . . . . .	51	45. DELMMON command syntax . . . . .	110
14. DEFTCPA command syntax . . . . .	53	46. DELPMON command syntax . . . . .	111
15. DEFVIEW command syntax . . . . .	56	47. DELRULE command syntax. . . . .	112
16. DELDSPS command syntax . . . . .	57	48. DELSCHD command syntax . . . . .	112
17. DELSRVM command syntax . . . . .	58	49. DELSMON command syntax . . . . .	113
18. DELTCPA command syntax . . . . .	59	50. RESUME command syntax . . . . .	114
19. DELVIEW command syntax . . . . .	60	51. RUN command syntax . . . . .	116
20. QUIT command syntax . . . . .	61	52. STATUS command syntax . . . . .	118
21. REVOKE command syntax . . . . .	61	53. Summary output from STATUS command	119
22. SHUTDOWN command syntax . . . . .	63	54. Details about all monitored virtual machines	119
23. STATUS command syntax. . . . .	64	55. Details about all spool usage monitors	120
24. Summary output from STATUS command	65	56. Details about users' spool files . . . . .	120
25. Details about all monitored virtual machines	65	57. SUSPEND command syntax . . . . .	121
26. Details about all spool usage monitors . . . . .	66	58. ALTRCON command syntax . . . . .	124
27. Details about users' spool files . . . . .	66	59. LOGTEXT command syntax . . . . .	126
28. STOP command syntax . . . . .	67	60. VIEWCON command syntax . . . . .	127
29. CONFIG command syntax . . . . .	70	61. VIEWLOG command syntax . . . . .	132
30. DEFACTN command syntax . . . . .	72	62. VIEWSPL command syntax . . . . .	136
31. DEFDATE command syntax . . . . .	78	63. GOMRSIF command syntax. . . . .	152
32. DEFEMON command syntax. . . . .	80	64. GOMGLBL command syntax . . . . .	162





---

## Tables

1. General FACILITY class profiles . . . . .	13	57. DEFIMON return codes . . . . .	85
2. Command-level FACILITY class profiles . . . . .	14	58. DEFGROUP operand value lengths . . . . .	86
3. GOMMAIN return codes . . . . .	32	59. DEFGROUP return codes . . . . .	87
4. GOMCMD return codes . . . . .	33	60. DEFMMON operand value lengths . . . . .	89
5. GOMSVM return codes . . . . .	34	61. DEFMMON return codes . . . . .	90
6. GOMTDM return codes . . . . .	37	62. DEFPMON operand value lengths . . . . .	92
7. AUTH operand value lengths . . . . .	40	63. DEFPMON return codes . . . . .	93
8. AUTH return codes . . . . .	41	64. DEFULE operand value lengths . . . . .	96
9. CMS operand value lengths . . . . .	43	65. DEFRULE return codes . . . . .	98
10. CMS return codes . . . . .	43	66. DEFSCHD operand value lengths . . . . .	101
11. CONFIG operand value lengths . . . . .	45	67. DEFSCHD return codes . . . . .	102
12. CONFIG return codes . . . . .	45	68. DEFSSMON operand value lengths . . . . .	104
13. CP operand value lengths . . . . .	46	69. DEFSSMON return codes . . . . .	105
14. CP return codes . . . . .	46	70. DELACTN operand value lengths . . . . .	106
15. DEFDSPS operand value lengths . . . . .	47	71. DELACTN return codes . . . . .	106
16. DEFDSPS return codes . . . . .	47	72. DELDATE operand value lengths . . . . .	106
17. DEFOPTN operand value lengths . . . . .	50	73. DELDATE return codes . . . . .	107
18. DEFOPTN return codes . . . . .	51	74. DELEMON operand value lengths . . . . .	107
19. DEFSRVM operand value lengths . . . . .	52	75. DELEMON return codes . . . . .	108
20. DEFSRVM return codes . . . . .	53	76. DELGROUP operand value lengths . . . . .	109
21. DEFTCPA operand value lengths . . . . .	54	77. DELGROUP return codes . . . . .	109
22. DEFTCPA return codes . . . . .	55	78. DELIMON operand value lengths . . . . .	109
23. DEFVIEW operand value lengths . . . . .	56	79. DELIMON return codes . . . . .	110
24. DEFVIEW return codes . . . . .	57	80. DELMMON operand value lengths . . . . .	110
25. DELDSPS operand value lengths . . . . .	57	81. DELMMON return codes . . . . .	111
26. DELDSPS return codes . . . . .	57	82. DELPMON operand value lengths . . . . .	111
27. DELSRVM operand value lengths . . . . .	58	83. DELPMON return codes . . . . .	111
28. DELSRVM return codes . . . . .	59	84. DELRULE operand value lengths . . . . .	112
29. DELTCPA operand value lengths . . . . .	59	85. DELRULE return codes . . . . .	112
30. DELTCPA return codes . . . . .	60	86. DELSCHD operand value lengths . . . . .	113
31. DELVIEW operand value lengths . . . . .	60	87. DELSCHD return codes . . . . .	113
32. DELVIEW return codes . . . . .	61	88. DELSMON operand value lengths . . . . .	114
33. QUIT return codes . . . . .	61	89. DELSMON return codes . . . . .	114
34. REVOKE operand value lengths . . . . .	62	90. RESUME operand value lengths . . . . .	115
35. REVOKE return codes . . . . .	63	91. RESUME return codes . . . . .	116
36. SHUTDOWN return codes . . . . .	63	92. RUN operand value lengths . . . . .	117
37. STATUS operand value lengths . . . . .	65	93. RUN return codes . . . . .	117
38. STATUS return codes . . . . .	66	94. STATUS operand value lengths . . . . .	119
39. STOP return codes . . . . .	67	95. STATUS return codes . . . . .	120
40. CONFIG operand value lengths . . . . .	71	96. SUSPEND operand value lengths . . . . .	122
41. CONFIG return codes . . . . .	71	97. SUSPEND return codes . . . . .	122
42. DEFACTN operand value lengths . . . . .	74	98. Reserved data space names . . . . .	123
43. DEFACTN substitution variables . . . . .	74	99. ALTRCON operand value lengths . . . . .	125
44. DEFEMON substitution variables . . . . .	75	100. ALTRCON return codes . . . . .	126
45. DEFIMON substitution variables . . . . .	75	101. LOGTEXT operand value lengths . . . . .	126
46. DEFMMON substitution variables . . . . .	75	102. LOGTEXT return codes . . . . .	126
47. DEFPMON substitution variables . . . . .	76	103. VIEWCON operand value lengths . . . . .	128
48. DEFRULE substitution variables . . . . .	76	104. VIEWCON return codes . . . . .	131
49. DEFSCHD and RUN substitution variables . . . . .	77	105. VIEWLOG operand value lengths . . . . .	133
50. DEFSSMON substitution variables . . . . .	77	106. VIEWLOG return codes . . . . .	135
51. DEFACTN return codes . . . . .	78	107. VIEWSPL operand value lengths . . . . .	137
52. DEFDATE operand value lengths . . . . .	79	108. Spool file list indicators . . . . .	138
53. DEFDATE return codes . . . . .	79	109. VIEWSPL return codes . . . . .	142
54. DEFEMON operand value lengths . . . . .	81	110. Accounting record format . . . . .	145
55. DEFEMON return codes . . . . .	82	111. Monitoring record format . . . . .	146
56. DEFIMON operand value lengths . . . . .	84	112. Sample translation table files . . . . .	150

113. GOMRSIF return codes . . . . .	153	115. Severity codes . . . . .	165
114. GOMGLBL return codes . . . . .	163		

---

## About this information

This book provides instructions for installing, configuring, and using IBM Operations Manager for z/VM.

This book is designed to help administrators and operators perform these tasks:

- Plan for and install Operations Manager
- Configure Operations Manager
- Use Operations Manager to monitor and manage z/VM virtual machines.

---

## Who should read this book

This book is intended for those persons responsible for installing and using Operations Manager, and assumes a working knowledge of:

- z/VM operating system
- operations management issues

---

## Service updates and support information

To find service updates and support information, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, refer to the following Web page:

[http://www.ibm.com/support/entry/portal/Overview/Software/Other\\_Software/Operations\\_Manager\\_for\\_z~VM](http://www.ibm.com/support/entry/portal/Overview/Software/Other_Software/Operations_Manager_for_z~VM)

---

## How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

▶▶—*required\_item*—————▶▶

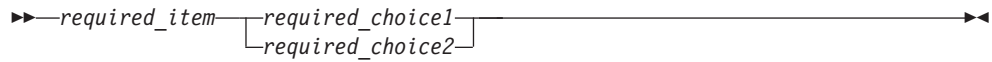
- Optional items appear below the main path.

▶▶—*required\_item*—————▶▶  
                          └—*optional\_item*—┘

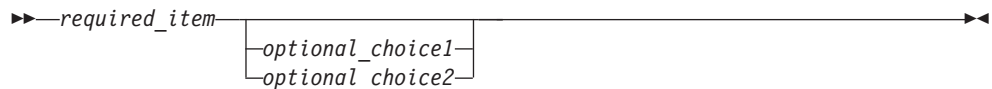
If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.



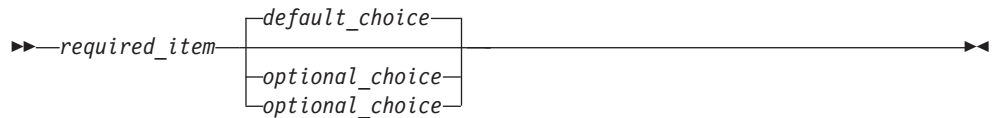
- If you can choose from two or more items, they appear vertically, in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.



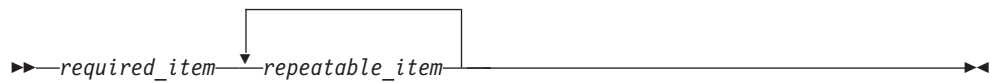
If choosing one of the items is optional, the entire stack appears below the main path.



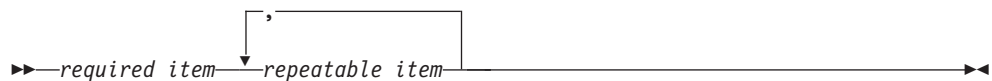
If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).

---

## How to look up message explanations

You can use any of the following methods to search for messages and codes:

## Searching for messages on the web

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you will be presented with links to the message information in IBM® information centers.

---

## Where to find information

The Operations Manager for z/VM Library Web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following Web page:  
<http://www.ibm.com/software/sysmgmt/zvm/operations/>

---

## How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other Operations Manager documentation, send your comments by email to [zvmtools@us.ibm.com](mailto:zvmtools@us.ibm.com). Be sure to include the name of the book, the part number of the book, the version of Operations Manager for z/VM, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).



---

## Summary of changes

---

### Version 1.5

This section summarizes the significant improvements or enhancements for IBM Operations Manager for z/VM V1.5 and refers you to relevant sections of this book for more information. Minor modifications to the text are not listed.

**Note:** Changes to V1.4 of this product can be found in previous releases of this guide.

#### SC18-9347-18 January, 2015

- For the DEFRULE command, you can now specify that a rule only applies for messages that are in a particular class. The new CLASS operand is designed to operate as an additional filter for processing messages received by Operations Manager. See “DEFRULE” on page 94 for more information.
- For the DEFGROUP and DELGROUP commands, you can now specify a MEMBER operand to extend the concept of “groups” beyond a group of users. The MEMBER operand allows groups to be used as a generalized collection of members and searched for inclusion/exclusion as needed. See “DEFGROUP” on page 85 and “DELGROUP” on page 108 for more information.
- For commands that refer to rules, schedules, and monitors, two new operands have been added, SYSGRP and EXSYSGRP, to specify the name of a group that contains system names. If the current z/VM system name is included in the group specified by SYSGRP or excluded from the group specified by EXSYSGRP, then the rule, schedule, or monitor applies to the current system. For more information, refer to the DEFRULE, DEFSCHD, DEFEMON, DEFIMON, DEFMMON, DEFPMON, and DEFSMON commands.

#### SC18-9347-17 October, 2013 - July, 2014

- For the DEFTCPA command, the PARM operand has a new option to include the TCP/IP address as the first field of the captured data. This change only applies when you are also using the TCPAPPL operand with the GOMRSYL parameter, as in TCPAPPL(GOMRSYL). See “Usage notes” on page 54 for more information.
- A new command has been added, CMDTCPA, which allows you to send a command request to a TCP/IP application. Refer to “CMDTCPA” on page 41 for more information.
- A new operand, DATERNG, allows you to specify a beginning and ending date range. Refer to the “VIEWCON” on page 127 and to “VIEWLOG” on page 131 for more information.
- A new section, “Verifying spool files are viewable” on page 156, has been added to Appendix E, “Testing Operations Manager installation and configuration,” on page 155. This section lists the steps to follow to verify that authorized users can view and purge spool files.
- A new appendix lists several sample statements and EXEC files that you can copy, modify, and use for your environment. Refer to Appendix A, “Sample files,” on page 143 for more information.
- Additional information has been provided about the privilege classes needed for Operations Manager's service machines. Refer to “Updates to CP directory” on page 6

page 6 regarding the OPMGRM1 service machine and to “Updates to CP directory” on page 8 regarding the OPMGRS*n* service machine.

- Several new messages have been added or significantly revised: GOM0012E, GOM0111L, GOM0112L, GOM0130W, GOM0237E, GOM0238E, GOM0285E, GOM0286E, GOM2940I, GOM2941I, GOM2942I, GOM0413E, GOM0414E, GOM0415E, GOM0416E, GOM0417I, GOM0418I, GOM0419E, GOM0969I, GOM1055E, and GOM2929I. Refer to Appendix H, “Messages and Codes,” on page 165 in order to review these messages.



---

## Chapter 1. Introduction to Operations Manager

This chapter introduces you to the functionality of Operations Manager, including its major components and some key benefits of using it.

**Related concepts:**

“What does Operations Manager do?”

“Benefits of using Operations Manager”

**Related reference:**

“Operations Manager components” on page 2

“Prerequisites” on page 3

---

### What does Operations Manager do?

Operations Manager is designed to improve the monitoring and management of z/VM virtual machines. This is accomplished by automating routine maintenance tasks and automatically responding to situations requiring intervention.

Operations Manager provides a set of commands that allow you to define automation tasks, define events that require automated responses, monitor users' consoles, and automate console or terminal sessions.

Operations Manager provides these functions:

- *Automating Tasks.* Commands are provided to define schedules and actions. *Schedules* define when to issue an *action* containing routine commands. An action can be issued once, on a specific hour or hourly, on a specific day or daily, on a holiday, monthly, or yearly.
- *Automating Responses.* Commands are provided to define rules and actions. *Rules* identify which messages should be acted upon and define a pattern to match against the messages. When a pattern match is detected, an action is scheduled to issue the appropriate commands.
- *Monitoring.* Commands are provided to monitor a user's console, spool usage, events, and the logs created by Operations Manager. Multiple administrators may interact with a user's console simultaneously as if they were logged onto the user's account.
- *Automating sessions.* Commands are provided to control sessions that will perform tasks at specific times or off-shift without manual intervention. For example, Operations Manager can automatically log on to a user ID, perform a set of routine tasks, log off, and then terminate the session.

---

### Benefits of using Operations Manager

Operations Manager provides several benefits to z/VM operators and administrators:

- *Improved operations accuracy.* By automating routine tasks, Operations Manager removes the responsibility for administrators to issue routine commands repetitively, correctly, and at the appropriate time.
- *Reduced operations requirements.* By automating the response to situations that require intervention, Operations Manager eliminates the requirement for operators to correctly interpret messages, determine corrective actions, and correctly respond to situations.

- *Improved administrative notification.* When a situation occurs that requires intervention, Operations Manager can interact with a variety of external methods to notify key personnel about an event. This allows real-time notification about situations occurring in critical services (such as networking) or critical components (such as a spool) that require expert attention.

---

## Operations Manager components

Operations manager consists of four separate components. These components work together to do all of Operations Manager message and action processing. Each of these components are described below.

**Related reference:**

Appendix F, “Application and settings interference,” on page 159

### Message processing component

The GOMMAIN module, which runs on the OPMGRM1 user ID, acts as a central message processing service. It activates the framework required to receive console messages, process rules and messages, monitor user IDs, monitor spool and page space, run schedules, and run the actions associated with them.

In order to start the GOMMAIN module, you must issue the GOMMAIN command from the command line or an EXEC on the OPMGRM1 user ID. The GOMMAIN command references a start up configuration file containing various commands Operation Manager will use for its monitoring and message processing. Operations Manager provides the sample configuration file OPMGRM1 CONFSAAMP. In addition, a second configuration file named EXAMPLES CONFSAAMP contains examples of rules, monitors, and actions.

**Related reference:**

“GOMMAIN” on page 31

### Action processing component

The GOMSVM module, which runs on action processing servers (OPMGRS1, OPMGRS2, etc.), processes the actions you defined with various Operations Manager commands.

**Related reference:**

“GOMSVM” on page 34

### User interaction component

The GOMCMD module provides a way for users to interact with Operations Manager without starting or stopping it. Users use the GOMCMD command to dynamically issue any Operations Manager control or configuration commands from a user's console.

**Related reference:**

“GOMCMD” on page 32

### Automated session component

The GOMTDM module provides a way to automatically activate a session in response to a situation (message). For example, administrators can have Operations Manager log on to a user ID, perform a set of routine tasks, log off, and then disconnect.

**Related reference:**

## Component interaction

The components of Operations Manager communicate with each other using Inter-User Communications Vehicle (IUCV). The following diagram shows their relationship and interaction:

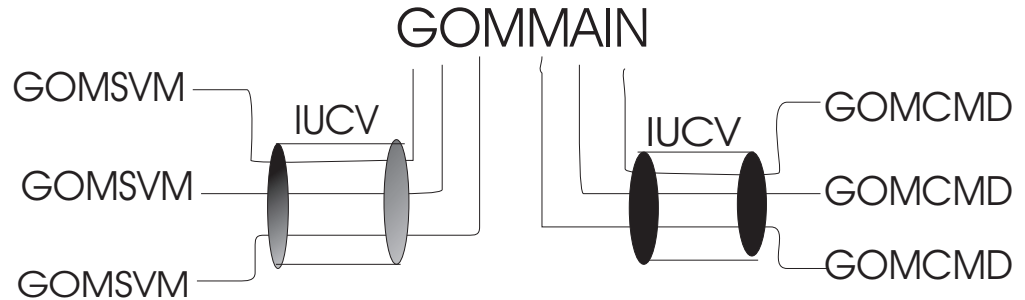


Figure 1. Operations Manager component interaction

---

## Prerequisites

### Hardware requirements

Operations Manager runs on any hardware that supports the required version of z/VM.

### Software requirements

Operations Manager runs on all currently supported versions and releases of z/VM.



---

## Chapter 2. Configuring Operations Manager

To install Operations Manager, follow the installation instructions provided in the *Operations Manager for z/VM Program Directory (GI10-8664)*. After you install Operations Manager, perform the steps described in this chapter to configure Operations Manager for your environment.

### Related tasks:

“Step 1. Update OPMGRM1”

“Step 2. Update Action Processing Servers” on page 7

“Step 3. Update End User and Administrator IDs” on page 9

“Step 4. Update IDs monitored by Operations Manager” on page 9

“Step 5. (Optional) Configure an external security mechanism” on page 11

---

### Step 1. Update OPMGRM1

#### Updates to PROFILE EXEC

##### About this task

You must create a PROFILE EXEC for any user ID that will be running GOMMAIN. The default user ID is OPMGRM1.

To create the PROFILE EXEC on a system with Operations Manager installed on minidisks, follow these steps:

1. Logon to MAINT or MAINT620.
2. LINK 5697J10E 2C2 334 RR
3. ACC 334 E
4. LINK OPMGRM1 191 333 MR
5. ACC 333 F
6. COPYFILE PROFM1 SAMPEXEC E PROFILE EXEC F

To create the PROFILE EXEC on a system with Operations Manager installed in SFS directories, replace steps 2, 3, 4, and 5 with the appropriate ACCESS commands for the SFS directories.

The PROFILE EXEC for OPMGRM1 and any other user ID that will be running GOMMAIN should include the following statements:

```
/* Sample lines to include in OPMGRM1 PROFILE EXEC */
'CP SET RUN ON'
'ACCESS 194 D'
'ACCESS 198 E'
'ACCESS 400 G'
'SET LANGUAGE ( ADD GOM USER )'
/* Replace filename, filetype, fm with startup configuration file */
'GOMMAIN filename filetype fm'
'CP LOGOFF'
EXIT
```

For the GOMMAIN statement, replace *filename filetype fm* with the file name, file type, and file mode of the startup configuration file. We recommend using OPMGRM1 CONFIG E or &SYSNAME CONFIG E where &SYSNAME will be resolved by Operations

Manager to your system name (node ID). Refer to “Copying the sample configuration file” on page 23 for more information on the configuration file name.

These sample statements can be found in the file PROFM1 SAMPEXEC provided on the Operations Manager samples disk 5697J10E 2C2, or the associated SFS directory.

#### Release and detach linked disks

When you have finished copying and updating the PROFILE EXEC, the linked disks must be released and detached. If you performed the steps listed above, the linked disks would be 334 and 333. To release and detach these disks, type the following:

```
REL 334 (DET
REL 333 (DET
```

#### Related reference:

“GOMMAIN” on page 31

Chapter 3, “Creating a startup configuration file,” on page 21

## Updates to CP directory

OPMGRM1, and any other user ID running GOMMAIN, requires the following CP directory statements. These sample statements are located in the file OPMGRM1 SAMPDIR provided on the Operations Manager samples disk 5697J10E 2C2 or associated SFS directory.

```
IDENTITY OPMGRM1 XXXXXXXX 64M 1024M ABCDG
  BUILD ON * USING SUBCONFIG OPMGM1-1
  ACCOUNT 5697-J10 GOM
  MACHINE XC
  XCONFIG ACCESSLIST ALSIZE 1022
  XCONFIG ADDRSPACE MAXNUMBER 1022 TOTSIZE 4G SHARE
  IUCV ALLOW MSGLIMIT 65535
  IUCV ANY MSGLIMIT 65535
  IUCV *MSG MSGLIMIT 65535
  IUCV *SPL MSGLIMIT 65535
  IUCV *VMEVENT MSGLIMIT 65535
  OPTION ACCT APPLMON
  IPL CMS
  CONSOLE 0009 3215
  SPOOL 000C 2540 READER *
  SPOOL 000D 2540 PUNCH A
  SPOOL 000E 1403 A
  LINK MAINT 190 190 RR
  LINK MAINT 19E 19E RR
  LINK MAINT 19D 19D RR
*
SUBCONFIG OPMGM1-1
  MDISK 191 <devtype> <startcyl> <size> <volume> MR rpass wpass mpass
  MDISK 194 <devtype> <startcyl> <size> <volume> MR rpass wpass mpass
  MDISK 198 <devtype> <startcyl> <size> <volume> RR rpass wpass mpass
  MDISK 400 <devtype> <startcyl> <size> <volume> RR rpass wpass mpass
  MDISK 410 <devtype> <startcyl> <size> <volume> RR rpass wpass mpass
```

The default sizes using device type 3390 are the following:

- 191: 005
- 194: 050
- 198: 005
- 400: 005
- 410: 005

The CP directory for the user ID OPMGRM1, and any other user ID running GOMMAIN, requires OPTION ACCT to allow it to use DIAG x'4C' and generate accounting records. This statement is only required if you want accounting records to be created.

The CP directory for the user ID OPMGRM1, and any other user ID running GOMMAIN, requires OPTION APPLMON to allow it to use DIAG x'DC' and generate monitor records. This statement is only required if you want monitoring records to be created.

Privilege classes are required to use certain components of Operations Manager. The required classes are listed below.

- Privilege class A is required for the FORCE command.
- Privilege class A or B is required for the XAUTOLOG command.
- Privilege class A, B, or C is required for the QUERY SECUSER and QUERY OBSERVER commands.
- Privilege class C is required for the SEND command.
- Privilege class D is required for the CHANGE, TRANSFER, and PURGE commands related to spool files.
- Privilege class D is required for the following QUERY commands:
  - ALLOC PAGE
  - MAXSPOOL
  - FILES
  - ALLOC SPOOL
- Privilege class B is required for the CP diagnostic code X'00D4'.
- Privilege class D is required for the CP diagnostic code X'00D8'.
- Privilege class B or E is required for the CP diagnostic code X'02CC'.

The user ID running GOMMAIN also needs authority to run any actions using the ENV(LVM) operand. This operand specifies that the command is to be processed by the local virtual machine (OPMGRM1). Refer to “Operands” on page 73 for more information on the ENV(LVM) operand.

---

## Step 2. Update Action Processing Servers

### Updates to PROFILE EXEC

#### About this task

You must create a PROFILE EXEC for any user ID that will be running GOMSVM to process actions. The default user IDs are OPMGRS1 through OPMGRS4.

To create the PROFILE EXEC on a system with Operations Manager installed on minidisks, follow these steps:

1. Logon to MAINT or MAINT620.
2. LINK 5697J10E 2C2 334 MR
3. ACC 334 E
4. LINK OPMGRS1 191 333 MR
5. ACC 333 F
6. COPYFILE PROFS1 SAMPEXEC E PROFILE EXEC F
7. Repeat these steps for each action processing server.

To create the PROFILE EXEC on a system with Operations Manager installed in SFS directories, replace steps 2, 3, 4, and 5 with the appropriate ACCESS commands for the SFS directories.

The PROFILE EXEC for OPMGRS1 and any other user ID that will be running GOMSVN should include the following statements:

```
/* Sample lines to include in OPMGRS1-4 PROFILE EXEC */
'CP SET RUN ON'
'ACCESS 198 E'
'ACCESS 400 G'
'SET LANGUAGE ( ADD GOM USER )'
'GOMSVN OPMGRM1'
'CP LOGOFF'
EXIT
```

These sample statements can be found in the file PROFS1 SAMPEXEC provided on the Operations Manager samples disk 5697J10E 2C2, or the associated SFS directory.

**Note:** When OPMGRM1 (or any server running GOMMAIN) is stopped, any action processing servers connected to it are also stopped. If the PROFILE EXEC on each action processing server contains 'CP LOGOFF', as shown above, the action processing servers are automatically logged off when OPMGRM1 is stopped.

#### Release and detach linked disks

When you have finished copying and updating the PROFILE EXEC, the linked disks must be released and detached. If you performed the steps listed above, the linked disks would be 334 and 333. To release and detach these disks, type the following:

```
REL 334 (DET
REL 333 (DET
```

## Updates to CP directory

### About this task

User IDs OPMGRS1–4 require the following CP directory statements. These sample statements are located in the file OPMGRS1 SAMPDIR provided on the Operations Manager samples disk 5697J10E 2C2 or associated SFS directory.

```
IDENTITY OPMGRS1 XXXXXXXX 32M 1024M G
BUILD ON * USING SUBCONFIG OPMGS1-1
ACCOUNT 5697-J10 GOM
MACHINE XC
XCONFIG ACCESSLIST ALSIZE 126
XCONFIG ADDRSPACE MAXNUMBER 126 TOTSIZE 512M SHARE
IUCV ALLOW MSGLIMIT 65535
IUCV ANY MSGLIMIT 65535
IUCV *MSG MSGLIMIT 65535
IPL CMS
CONSOLE 0009 3215
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
LINK OPMGRM1 198 198 RR
LINK OPMGRM1 400 400 RR
*
SUBCONFIG OPMGS1-1
MDISK 191 <devtype> <startcyl> <size> <volume> MR rpass wpass mpass
```



The default size using device type 3390 is 191: 005.

The user ID running GOMSVM also needs authority to run any actions using the ENV(SVM) operand. This operand specifies that the command is to be processed by any available action processing server. Refer to “Operands” on page 73 for more information on the ENV(SVM) operand.

---

## Step 3. Update End User and Administrator IDs

### About this task

**Note:** If the USER components of Operations Manager were copied to MAINT 19D/19E during product installation, you can omit the LINK and ACCESS statements from the sample below. However, you must include the SET LANGUAGE statement. For more information on the USER components and MAINT 19D/19E, see "Copy Operations Manager Files Into Production," in the *Operations Manager for z/VM Program Directory (GI10-8664)*.

You must update the PROFILE EXEC for any user ID that will be using GOMCMD to issue commands or GOMTDM to automate console or terminal sessions. The PROFILE EXEC for these user IDs should include the following statements:

```
/* Sample lines to include in user's PROFILE EXEC to use GOMCMD */
/* Sample lines to include in user's PROFILE EXEC to use GOMTDM */
'CP LINK OPMGRM1 410 410 RR'
'ACCESS 410 G'
'SET LANGUAGE ( ADD GOM USER )'
```

These sample statements can be found in the file PROFUSR SAMPEXEC supplied with Operations Manager.

**Note:** Privilege class G is required for the user ID running GOMCMD.

---

## Step 4. Update IDs monitored by Operations Manager

### About this task

Each user ID console that you want Operations Manager to monitor must be updated. Set the SECUSER or OBSERVER user ID to OPMGRM1, or the name of the user ID running GOMMAIN. You must update either the CP directory or the PROFILE EXEC of each monitored user ID. IBM recommends updating the CP directory, allowing Operations Manager to see all messages on the console from the start of the logon process.

**Note:**

- If you use OBSERVER for monitoring users, message HCP150A is not generated. If you use SECUSER, message HCP150A is generated.
- If you use SECUSER, console messages are not sent to the target user ID unless the source user ID is running disconnected. If you use OBSERVER, console messages are sent to the target user ID when both the source user ID is running disconnected and when it is logged onto a console.
- If you use SECUSER, Operations Manager can send commands to the source user ID (through Operations Manager actions) without the need for an additional privilege class. If you use OBSERVER, Operations Manager servers must be able to issue the CP SEND command. By default this requires privilege class C.

- If you use OBSERVER, actions that result in a message being sent to an observed console may not appear in the console output viewable via the Operations Manager VIEWCON command.
- If a disconnected user has a SECUSER or OBSERVER defined and also runs an application which connects to the CP \*MSG service in order to capture CPCONIO and/or VMCONIO output to the virtual machine console, some or all of that output may be sent to the SECUSER or OBSERVER rather than the local \*MSG connection. This may result in the application operating incorrectly.
- If you choose to monitor the Operations Manager action processing servers through Operations Manager (by setting their SECUSER to OPMGRM1 or the user ID running GOMMAIN):
  - Messages from actions run on the action processing servers will appear in VIEWCON for the action processing server user ID(s), rather than only in VIEWLOG
  - Messages from actions run on the action processing servers will be processed against Operations Manager rules and may trigger actions; if an action should produce a message which matches the rule that triggered it, a loop may result.
- Refer to Appendix F, “Application and settings interference,” on page 159 for more information on how OBSERVER and SECUSER settings can impact Operations Manager.

## Updating the CP directory

### About this task

To update the CP DIRECTORY of the monitored user ID, you might use one of the following commands:

```
CONSOLE 009 3215 T OPMGRM1
```

or

```
CONSOLE 009 3215 T OPMGRM1 OBSERVER
```

For more information, see the CONSOLE statement in *z/VM CP Planning and Administration (SC24–60431)*.

## Updating the PROFILE EXEC

### About this task

To update the PROFILE EXEC of the monitored user ID, you might use one of the following commands:

```
CP SET SECUSER OPMGRM1
```

or

```
CP SET OBSERVER OPMGRM1
```

For more information, see the SET SECUSER command in the *z/VM CP Command and Utility Reference (SC24–6008)*.

## Verifying the console data is being sent to Operations Manager

To verify that the console data for a monitored user ID is being sent to Operations Manager, follow the steps below.

### About this task

If you update the CP directory for this user ID to have the console data monitored by Operations Manager, it does not go into effect until the user ID logs off and logs on again (or re-IPLs). The SET SECUSER command causes the user ID to immediately start sending console data to Operations Manager. The next time that the user ID is restarted, the updated directory entry takes effect.

### Procedure

1. Log on to an authorized user ID. This user ID must be authorized to issue the CP QUERY SECUSER and CP SET SECUSER commands.
2. Select a user ID that you configured to have its console monitored by Operations Manager. Refer to “Step 4. Update IDs monitored by Operations Manager” on page 9 for more information.
3. Issue the following command to check if the user ID is being monitored by Operations Manager:

```
CP QUERY SECUSER userid
```

4. Follow the appropriate condition:

- If the user ID is already being monitored by Operations Manager, no action is needed.
- If the user ID is not being monitored by Operations Manager, issue the following command:

```
CP SET SECUSER userid OPMGRM1
```

---

## Step 5. (Optional) Configure an external security mechanism

### About this task

If you use an ESM (external security manager) that supports the RACROUTE macro and the FACILITY class, you can configure Operations Manager to use the ESM as its authorization mechanism instead of using its own internal authorization mechanism. You can control which users have authority to issue which commands through the ESM, as well as which users can view the consoles and spool files of specific or all users.

## Setup example for IBM Resource Access Control Facility (RACF)

### About this task

OPMGRM1, and any other user ID running GOMMAIN, must have authority to issue checks for access to a resource on behalf of another user. An example on how to do this for IBM Resource Access Control Facility (RACF) is provided, however, refer to the latest documentation for your particular release of ESM for details.

To use RACF as the authorization mechanism, perform the following steps:

1. Authorize communication with the RACF/VM server through IUVC.
2. Authorize third-party authorization checks.

3. Identify the RACF/VM server to which RACROUTE will be sent.
4. Set ESM(Y) on the DEFOPTN command.

Each of these steps are described below.

### **Authorize communication with RACF/VM**

Servers communicate with RACF through IUCV. There are two options for authorizing this communication. You should consider which option is appropriate for your system's security administration policies.

- The more general option is to have the RACF/VM server (usually called RACFVM) include an IUCV ALLOW statement in its directory entry. This will permit any user in the system to establish communication with the RACF server.
- The more restricted option is to have Operations Manager service machine OPMGRM1 include an IUCV statement for the RACF/VM server (for example, IUCV RACFVM PRIORITY MSGLIMIT 255) in its directory entry. This will permit only the users with this statement in their directory entry to establish communication with the RACF server.

### **Authorize third-party checks**

Authority to issue third-party authorization checks is controlled by the ICHCONN profile in the FACILITY class in RACF/VM. You must create this profile if it does not exist and give the Operations Manager server UPDATE access to it.

### **Identify the RACF/VM server**

The RACROUTE interface must be able to identify the RACF/VM server. This is done with the RACF SERVMACH file, which is normally loaded to the CMS Y-disk during RACF installation. If you want to use a different RACF server, or your installation has removed the RACF SERVMACH file from general access, you should provide a tailored RACF SERVMACH file to Operations Manager's main server OPMGRM1.

### **Set the ESM(Y) option**

To activate the use of an external security manager, you must set the ESM operand value to Y on the DEFOPTN command in the Operations Manager startup configuration file.

```
DEFOPTN ESM(Y)
```

Even with ESM set to Y, it is recommended that at least one user be given access to Operations Manager in case ESM must be set to "N" temporarily. Refer to "Authority assignments with AUTH and external security" on page 26 for more information.

Refer to DEFOPTN Operands for more information on the available operands.

### **Set the ESMPREF() option**

If you set DEFOPTN ESM(Y), you can use the ESMPREF operand to configure the prefix Operations Manager adds to the RACF profile names. Operations Manager automatically adds GOM. to the front of your prefix. For example, if you specify ESMPREF(OPSMGR), then FACILITY class profiles must be defined with the prefix GOM.OPSMGR.

If you set DEFOPTN ESM(Y) and do not configure the ESMPREF operand, the default prefix is RKT.OPM. In this case, FACILITY class profiles must be defined with the prefix RKT.OPM

All Operations Manager RACF profiles are in the FACILITY class.

Refer to DEFOPTN Operands for more information on using the ESMPREF operand.

## FACILITY class profiles

All of the resources Operations Manager uses are profiles in the FACILITY class. These profiles determine users' level of authorization.

Replace ESMPREF in the tables below with one of the following:

- RKT.OPM - This is the default. Use this value if you have not specified a value for ESMPREF on the DEFOPTN statement in the Operations Manager configuration file.
- GOM.*prefix* - where *prefix* is a name of your choosing. The *prefix* must match the value specified for ESMPREF on the DEFOPTN statement in the Operations Manager configuration file.

Operations Manager can be configured to check profiles at a general level or by specific commands. The following sections describe the available profiles for each option.

### General profiles

If DEFOPTN ESM(Y),AUTHCMD(N) is set, Operations Manager checks these profiles:

Table 1. General FACILITY class profiles

Profile	Authorization
ESMPREF.CONTROL	Users who have READ access to this profile have authority to issue control and configuration commands.
ESMPREF.CONFIG	Users who have READ access to this profile have authority to issue configuration commands.
ESMPREF.ALTRCON. <i>userid</i> Refer to the note below for the valid replacements of <i>userid</i> .	Users who have READ access to this profile have authority to alter the display attributes of a line in a user's console when it is displayed using VIEWCON.
ESMPREF.VIEWCON. <i>userid</i> Refer to the note below for the valid replacements of <i>userid</i> .	Users who have READ access to this profile have authority to view consoles.
	Users who have UPDATE access to this profile have authority to view consoles and issue commands.
ESMPREF.VIEWSPL. <i>userid</i> Refer to the note below for the valid replacements of <i>userid</i> .	Users who have READ access to this profile have authority to view spool files.
	Users who have UPDATE access to this profile have authority to view spool files, purge spool files, and alter a spool file's external attributes.
ESMPREF.CMDTCPA. <i>tcpaid</i> Refer to the note below for the valid replacements of <i>tcpaid</i> .	Users who have READ access to this profile have authority to interact with the TCP/IP program defined as <i>tcpaid</i> .

**Note:** The *userid* value in the *ESMPREF.ALTRCON.userid*, *ESMPREF.VIEWCON.userid*, and *ESMPREF.VIEWSPL.userid* profiles can be replaced with a wildcard pattern or one of the following options:

- Any individual user ID. For example, READ access to profile *ESMPREF.VIEWCON.JONES* allows the user to view the console of the user ID JONES.
- Any combined view defined by the DEFVIEW command. For example, READ access to profile *ESMPREF.VIEWCON.VIEW3* allows the user to view the combined view named VIEW3, which contains USER1, USER2, and USER3.
- A generic matching pattern. For example, READ access to profile *ESMPREF.VIEWCON.\** allows the user to view the console of any user monitored by Operations Manager. Before defining any generic profiles in the FACILITY class, ensure that the FACILITY class can contain generic profiles. Refer to *RACF Security Server Security Administrator's Guide (SC24-6142)*.

**Note:** The *tcpaid* value in the *ESMPREF.CMDTCPA.tcpaid* can be replaced with a wildcard pattern or one of the following options:

- Any individual TCP/IP application name. The TCP/IP application name is defined by the DEFTCPA command NAME(*tcpaid*) operand. For example, READ access to profile *ESMPREF.CMDTCPA.SYS23* allows the user to interact with the TCP/IP application defined with DEFTCPA NAME(SYS23).
- A generic matching pattern. For example, READ access to profile *ESMPREF.CMDTCPA.\** allows the user to interact with any TCP/IP application defined by DEFTCPA. Before defining any generic profiles in the FACILITY class, ensure that the FACILITY class can contain generic profiles. Refer to *RACF Security Server Security Administrator's Guide (SC24-6142)*.

## Command-level profiles

The DEFOPTN ESM(Y), AUTHCMD(Y) command provides the ability to authorize by specific commands. If DEFOPTN ESM(Y), AUTHCMD(Y) is set, Operations Manager checks these profiles:

Table 2. Command-level FACILITY class profiles

Profile	Authority
<i>ESMPREF.VIEWCON.userid</i> Refer to the note above for the valid replacements of <i>userid</i> .	Users who have READ access to this profile have authority to view consoles.
	Users who have UPDATE access to this profile have authority to view consoles and issue commands.
<i>ESMPREF.ALTRCON.userid</i> Refer to the note above for the valid replacements of <i>userid</i> .	Users who have READ access to this profile have authority to alter the display attributes of a line in a user's console when it is displayed using VIEWCON.
<i>ESMPREF.VIEWSPL.userid</i> Refer to the note above for the valid replacements of <i>userid</i> .	Users who have READ access to this profile have authority to view spool files.
	Users who have UPDATE access to this profile have authority to view spool files, purge spool files, and alter a spool file's external attributes.

Table 2. Command-level FACILITY class profiles (continued)

Profile	Authority
ESMPREF.COMDTCPA.tcpaid  Refer to the note above for the valid replacements of <i>tcpaid</i> .	Users who have READ access to this profile have authority to interact with the TCP/IP program defined as <i>tcpaid</i> .
ESMPREF.COMMAND.AUTH	Users who have READ access to this profile have authority for the AUTH command.
ESMPREF.COMMAND.CMS	Users who have READ access to this profile have authority for the CMS command.
ESMPREF.COMMAND.CONFIG	Users who have READ access to this profile have authority for the CONFIG command.
ESMPREF.COMMAND.CONFIG.ACTION	Users who have READ access to this profile have authority for the CONFIG CLEAR(ACTION) command.
ESMPREF.COMMAND.CONFIG.DATE	Users who have READ access to this profile have authority for the CONFIG CLEAR(DATE) command.
ESMPREF.COMMAND.CONFIG.EVENT	Users who have READ access to this profile have authority for the CONFIG CLEAR(EVENT) command.
ESMPREF.COMMAND.CONFIG.IDLE	Users who have READ access to this profile have authority for the CONFIG CLEAR(IDLE) command.
ESMPREF.COMMAND.CONFIG.GROUP	Users who have READ access to this profile have authority for the CONFIG CLEAR(GROUP) command.
ESMPREF.COMMAND.CONFIG.MACHINE	Users who have READ access to this profile have authority for the CONFIG CLEAR(MACHINE) command.
ESMPREF.COMMAND.CONFIG.PAGE	Users who have READ access to this profile have authority for the CONFIG CLEAR(PAGE) command.
ESMPREF.COMMAND.CONFIG.RULE	Users who have READ access to this profile have authority for the CONFIG CLEAR(RULE) command.
ESMPREF.COMMAND.CONFIG.SCHEDULE	Users who have READ access to this profile have authority for the CONFIG CLEAR(SCHEDULE) command.
ESMPREF.COMMAND.CONFIG.SPOOL	Users who have READ access to this profile have authority for the CONFIG CLEAR(SPOOL) command.
ESMPREF.COMMAND.CP	Users who have READ access to this profile have authority for the CP command.
ESMPREF.COMMAND.DEFACTN	Users who have READ access to this profile have authority for the DEFACTN command.

Table 2. Command-level FACILITY class profiles (continued)

Profile	Authority
ESMPREF.COMMAND.DEFDATE	Users who have READ access to this profile have authority for the DEFDATE command.
ESMPREF.COMMAND.DEFDSPS	Users who have READ access to this profile have authority for the DEFDSPS command.
ESMPREF.COMMAND.DEFEMON	Users who have READ access to this profile have authority for the DEFEMON command.
ESMPREF.COMMAND.DEFGROUP	Users who have READ access to this profile have authority for the DEFGROUP command.
ESMPREF.COMMAND.DEFIMON	Users who have READ access to this profile have authority for the DEFIMON command.
ESMPREF.COMMAND.DEFMMON	Users who have READ access to this profile have authority for the DEFMMON command.
ESMPREF.COMMAND.DEFOPTN	Users who have READ access to this profile have authority for the DEFOPTN command.
ESMPREF.COMMAND.DEFPMON	Users who have READ access to this profile have authority for the DEFPMON command.
ESMPREF.COMMAND.DEFRULE	Users who have READ access to this profile have authority for the DEFRULE command.
ESMPREF.COMMAND.DEFSCHD	Users who have READ access to this profile have authority for the DEFSCHD command.
ESMPREF.COMMAND.DEFSMON	Users who have READ access to this profile have authority for the DEFSMON command.
ESMPREF.COMMAND.DEFSRVM	Users who have READ access to this profile have authority for the DEFSRVM command.
ESMPREF.COMMAND.DEFTCPA	Users who have READ access to this profile have authority for the DEFTCPA command.
ESMPREF.COMMAND.DEFVIEW	Users who have READ access to this profile have authority for the DEFVIEW command.
ESMPREF.COMMAND.DELACTN	Users who have READ access to this profile have authority for the DELACTN command.
ESMPREF.COMMAND.DELDATE	Users who have READ access to this profile have authority for the DELDATE command.



Table 2. Command-level FACILITY class profiles (continued)

Profile	Authority
ESMPREF.COMMAND.DELDSPS	Users who have READ access to this profile have authority for the DELDSPS command.
ESMPREF.COMMAND.DELEMON	Users who have READ access to this profile have authority for the DELEMON command.
ESMPREF.COMMAND.DELGROUP	Users who have READ access to this profile have authority for the DELGROUP command.
ESMPREF.COMMAND.DELIMON	Users who have READ access to this profile have authority for the DELIMON command.
ESMPREF.COMMAND.DELMMON	Users who have READ access to this profile have authority for the DELMMON command.
ESMPREF.COMMAND.DELPMON	Users who have READ access to this profile have authority for the DELPMON command.
ESMPREF.COMMAND.DELRULE	Users who have READ access to this profile have authority for the DELRULE command.
ESMPREF.COMMAND.DELSCHD	Users who have READ access to this profile have authority for the DELSCHD command.
ESMPREF.COMMAND.DELSMON	Users who have READ access to this profile have authority for the DELSMON command.
ESMPREF.COMMAND.DELSRVM	Users who have READ access to this profile have authority for the DELSRVM command.
ESMPREF.COMMAND.DELTCPA	Users who have READ access to this profile have authority for the DELTCPA command.
ESMPREF.COMMAND.DELVIEW	Users who have READ access to this profile have authority for the DELVIEW command.
ESMPREF.COMMAND.LOGTEXT	Users who have READ access to this profile have authority for the LOGTEXT command.
ESMPREF.COMMAND.QUIT	Users who have READ access to this profile have authority for the QUIT command.
ESMPREF.COMMAND.RESUME	Users who have READ access to this profile have authority for the RESUME command.
ESMPREF.COMMAND.REVOKE	Users who have READ access to this profile have authority for the REVOKE command.

Table 2. Command-level FACILITY class profiles (continued)

Profile	Authority
ESMPREF.COMMAND.RUN	Users who have READ access to this profile have authority for the RUN command.
ESMPREF.COMMAND.SHUTDOWN	Users who have READ access to this profile have authority for the SHUTDOWN command.
ESMPREF.COMMAND.STATUS	Users who have READ access to this profile have authority for the STATUS command.
ESMPREF.COMMAND.STATUS.ACTION	Users who have READ access to this profile have authority for the STATUS DETAIL(ACTION) command.
ESMPREF.COMMAND.STATUS.ACTIONQ	Users who have READ access to this profile have authority for the STATUS DETAIL(ACTIONQ) command.
ESMPREF.COMMAND.STATUS.AUTH	Users who have READ access to this profile have authority for the STATUS DETAIL(AUTH) command.
ESMPREF.COMMAND.STATUS.CONSOLE	Users who have READ access to this profile have authority for the STATUS DETAIL(CONSOLE) command.
ESMPREF.COMMAND.STATUS.DATASP	Users who have READ access to this profile have authority for the STATUS DETAIL(DATASP) command.
ESMPREF.COMMAND.STATUS.DATE	Users who have READ access to this profile have authority for the STATUS DETAIL(DATE) command.
ESMPREF.COMMAND.STATUS.EVENT	Users who have READ access to this profile have authority for the STATUS DETAIL(EVENT) command.
ESMPREF.COMMAND.STATUS.GROUP	Users who have READ access to this profile have authority for the STATUS DETAIL(GROUP) command.
ESMPREF.COMMAND.STATUS.IDLE	Users who have READ access to this profile have authority for the STATUS DETAIL(IDLE) command.
ESMPREF.COMMAND.STATUS.IUCV	Users who have READ access to this profile have authority for the STATUS DETAIL(IUCV) command.
ESMPREF.COMMAND.STATUS.MACHINE	Users who have READ access to this profile have authority for the STATUS DETAIL(MACHINE) command.
ESMPREF.COMMAND.STATUS.OPTION	Users who have READ access to this profile have authority for the STATUS DETAIL(OPTION) command.
ESMPREF.COMMAND.STATUS.PAGE	Users who have READ access to this profile have authority for the STATUS DETAIL(PAGE) command.

Table 2. Command-level FACILITY class profiles (continued)

Profile	Authority
ESMPREF.COMMAND.STATUS.RULE	Users who have READ access to this profile have authority for the STATUS DETAIL(RULE) command.
ESMPREF.COMMAND.STATUS.SCHEDULE	Users who have READ access to this profile have authority for the STATUS DETAIL(SCHEDULE) command.
ESMPREF.COMMAND.STATUS.SERVICE	Users who have READ access to this profile have authority for the STATUS DETAIL(SERVICE) command.
ESMPREF.COMMAND.STATUS.SPOOL	Users who have READ access to this profile have authority for the STATUS DETAIL(SPOOL) command.
ESMPREF.COMMAND.STATUS.SPOOLUSR	Users who have READ access to this profile have authority for the STATUS DETAIL(SPOOLUSR) command.
ESMPREF.COMMAND.STATUS.TCPAPPL	Users who have READ access to this profile have authority for the STATUS DETAIL(TCPAPPL) command.
ESMPREF.COMMAND.STATUS.VIEW	Users who have READ access to this profile have authority for the STATUS DETAIL(VIEW) command.
ESMPREF.COMMAND.STOP	Users who have READ access to this profile have authority for the STOP command.
ESMPREF.COMMAND.SUSPEND	Users who have READ access to this profile have authority for the SUSPEND command.
ESMPREF.COMMAND.VIEWLOG	Users who have READ access to this profile have authority for the VIEWLOG command.

**Notes:**

1. If DEFOPTN ESM(Y),AUTHCMD(Y) is set, Operations Manager does *not* check these profiles:
  - ESMPREF.CONTROL
  - ESMPREF.CONFIG
2. Generic profiles can also be used based on your External Security Manager. For example, READ access to profile ESMPREF.COMMAND.\* gives a user the authority for all Operations Manager control and configuration commands. Before defining any generic profiles in the FACILITY class, ensure that the FACILITY class can contain generic profiles. Refer to *RACF Security Server Security Administrator's Guide (SC24-6142)*.

**Related reference:**

Chapter 5, "Syntax for Control Commands," on page 39  
 Chapter 6, "Syntax for Configuration Commands," on page 69  
 "DEFOPTN" on page 48  
 "DEFVIEW" on page 55

“VIEWCON” on page 127

---

## Chapter 3. Creating a startup configuration file

This chapter describes how to create a startup configuration file using the sample startup configuration file supplied with Operations Manager as a guide.

**Related concepts:**

“Entering commands”

**Related tasks:**

“Copying the sample configuration file” on page 23

“Adding comments” on page 24

“Defining action processing servers” on page 27

**Related reference:**

“Using the sample configuration file” on page 23

“Assigning authority” on page 25

“Verify the installation and configuration” on page 28

---

### Entering commands

Operations Manager requires specific rules when entering commands. Commands consist of a single command and optionally one or more operands and their values. Below is a basic example of a command input format:

```
COMMAND OPERAND1(operand value),OPERAND2(operand value),OPERAND3(operand
(value)),OPERAND4('operand ( value')
```

### Command rules

The following rules must be applied when entering commands to properly configure Operations Manager:

- The command must start with a **COMMAND**. Below is an example of a command with no operands:  
STATUS
- If operands are required, one or more spaces must separate the **COMMAND** from the first operand.
- The operand value follows the operand and is enclosed in parentheses. No spaces are allowed between:
  - the operand and the left parenthesis.
  - the end of the operand value and the right parenthesis.

Below is an example of a command with one operand:

```
STATUS DETAIL(RULE)
```

- The minimum operand value length is one character and the maximum operand value length is determined by the command.
- If multiple operands are required, a comma must follow the right parenthesis. Below is an example of a command with multiple operands:  
DEFRULE NAME(RULE01),MATCH(\*HCPMID6001I\*),ACTION(MIDNITE)
- Spaces are allowed in between operands. The spaces must follow the comma of the previous operand. The following examples are equivalent:

```
DEFRULE NAME(ARULE),MATCH(*HENRY*),ACTION(HR1),PARM(ARULE)
```

```
DEFRULE NAME(ARULE), MATCH(*HENRY*), ACTION(HR1), PARM(ARULE)
```

- An operand may be entered only once for a command.
- If an operand value contains parentheses, the parentheses must be balanced or single quotes must be used. Below is an example of balanced parentheses in the COMMAND operand:

```
DEFACTN NAME(ACTN01),COMMAND(PRINT A FILE ( NOCC )),ENV(GOM)
```

- Any operand value may be entered in single quotes ('). Quoting an operand value allows the operand value to bypass the balanced parentheses rule. The rules to quoting are:
  - the beginning quote must follow the left parenthesis with no spaces.
  - the operand value begins in the position after the beginning single quote and the operand value ends in the position before the ending quote.
  - the ending quote must be followed by the right parenthesis with no spaces.
  - if a single quote is required in a quoted operand value, use two single quotes to produce a single quote in the operand value.

Below is an example of a quoted operand value in the COMMAND operand:

```
DEFACTN NAME(ACTN01),COMMAND('PRINT A FILE ( NOCC)'),ENV(LVM)
```

- Operands may be entered in any order. For example, the following commands are equivalent:

```
DEFRULE NAME(RULE01), MATCH(*HCPMID6001I*), ACTION(MIDNITE)
DEFRULE MATCH(*HCPMID6001I*),NAME(RULE01),ACTION(MIDNITE)
DEFRULE NAME(RULE01), ACTION(MIDNITE), MATCH(*HCPMID6001I*)
```

## Where to enter commands

Commands may be entered through any of the following commands or files:

- The console of OPMGRM1, or any other user ID running the GOMMAIN command.

You must be logged onto the virtual machine running the GOMMAIN command.

- The GOMCMD command.
- The command line in VIEWLOG.
- An Operations Manager configuration file.

An Operations Manager configuration file can be specified and processed using:

- the GOMMAIN command. This file is commonly called the startup configuration file.
- the CONFIG command.

### Notes:

1. The CMS file containing Operations Manager commands can have fixed or variable length records and a maximum record length of 255.
2. Any line beginning with an asterisk (\*) is considered a comment and is not parsed by Operations Manager.
3. For improved readability, the Operations Manager commands in a CMS file may be broken into multiple lines joined by a continuation indicator: the plus sign (+). Below is an example of a CMS file containing an Operations Manager command broken into multiple lines:

```

*
*DEFINE A SCHEDULE TO CHECK PRIMARY SPACE ALLOCATION HOURLY
*
DEFSCHD NAME(SCHD01),+
EVERY(01:00),+
HOURS(01,02,03,04,05,06,07,18,19,20,21,22,23),+
DAYS(MTWT***),+
CONFLICT(SCHD02,SCHD03),+
MAXDEFER(20),+
ACTION(CHECKSP),+
PARM(PRIMARY)

```

**Related reference:**

“GOMCMD” on page 32

“VIEWLOG” on page 131

“GOMMAIN” on page 31

“CONFIG” on page 43

## Using the sample configuration file

You can configure Operations Manager by issuing commands directly on the console of the server running GOMMAIN, by issuing commands to the server using the GOMCMD command, or by using a startup configuration file.

Operations Manager provides a sample startup configuration file named OPMGRM1 CONFSAMP. This file includes commands that prepare your Operations Manager environment. You can use the file OPMGRM1 CONFSAMP as a general guide to compose your own startup configuration file. In addition, Operations Manager provides some examples of rules, monitors, and actions in EXAMPLES CONFSAMP. You can use these as a general guide for your own rules, monitors, and actions.

After you create your own startup configuration file, specify this file name on the GOMMAIN command in the PROFILE EXEC of OPMGRM1.

The sections that follow describe the possible components of a startup configuration file and shows examples from the startup configuration file OPMGRM1 CONFSAMP.

**Related reference:**

Chapter 5, “Syntax for Control Commands,” on page 39

Chapter 6, “Syntax for Configuration Commands,” on page 69

“GOMMAIN” on page 31

## Copying the sample configuration file

### About this task

You must copy the sample startup configuration file to the OPMGRM1 198 disk before modifying it. The following steps provide a guideline to copy the sample configuration file:

1. Log on to the VM account your installation will use to modify Operations Manager configuration files. Do *not* use OPMGRM1 or OPMGRS1–4.
2. Link to the minidisk that contains the samples:  
CP LINK 5697J10E 2C2 333 RR
3. Access the sample minidisk:  
ACCESS 333 I

4. Link to the minidisk that contains the configuration files:

```
CP LINK OPMGRM1 198 198 MR
```

5. Access the configuration minidisk:

```
ACCESS 198 E
```

6. Copy the sample file:

```
COPYFILE OPMGRM1 CONFSAMP I filename CONFIG E, where filename is the name you plan to use for your configuration file.
```

IBM recommends that you use one of the following for the filename:

- **OPMGRM1** -- This file can contain one of the following:
  - any or all of your configuration information for a single system
  - all of the common configuration information for multiple members of a z/VM Single System Image (SSI) cluster.

If you have unique configuration information for each member of an SSI cluster, it can be stored in another configuration file whose filename is the name of the member, and imbedded in this configuration file using the CONFIG statement with the FN(&SYSNAME) parameter. At Operations Manager startup, &SYSNAME will be replaced with the name of the system.

- **System name** -- The filename of the configuration file is actually the system name for your z/VM system. This is convenient in an SSI environment because the configuration file contains only the Operations Manager configuration information unique to this member (system name) of the cluster. Within this configuration file, you can use the CONFIG statement to imbed another configuration file that contains all the configuration information that is common across all members of the cluster.

If you choose to specify the system name, be sure to update or remove the CONFIG statement, if any, within your configuration file. By default, the sample configuration file (OPMGRM1 CONFSAMP) provided with Operations Manager includes a CONFIG statement that imbeds &SYSNAME CONFIG E, resulting in your startup configuration file imbedding itself.

7. Modify the startup configuration file:

```
XEDIT OPMGRM1 CONFIG E
```

The following sections describe the possible areas of the configuration file you can modify.

**Related tasks:**

“Adding comments”

“Defining action processing servers” on page 27

“Defining system options” on page 27

**Related reference:**

“Assigning authority” on page 25

“Modifying the startup configuration file” on page 27

---

## Adding comments

### About this task

Operations Manager provides a way to add comments in the startup configuration file. You might do this when describing the purpose of a specific command sequence. To add a comment, place an asterisk (\*) before the comment. For



example, if you want to assign authority to a user ID, you might add a comment like this before the AUTH command that assigns the authority:

```
* Define users with control authority to Operations Manager
AUTH USER(user1),CONTROL(Y)
AUTH USER(user2),CONTROL(Y)
```

---

## Assigning authority

Operations Manager provides the AUTH command to assign different types of authority to specific user IDs. These commands are only used if an External Security Manager is *not* in use.

### Related tasks:

“Step 5. (Optional) Configure an external security mechanism” on page 11

### Related reference:

“AUTH” on page 39

## Control authority

### About this task

To define a specific user ID so that it has authority to use Operations Manager control commands, issue the commands shown below. In this example, user1 and user2 are granted control authority.

```
* Define users with control authority to Operations Manager
AUTH USER(MAINT),CONTROL(Y)
AUTH USER(MAINT630),CONTROL(Y)
```

### Related reference:

Chapter 5, “Syntax for Control Commands,” on page 39

## Configuration authority

### About this task

To give a specific user ID authority to use Operations Manager configuration commands, issue the command shown below. In this example, user3 is granted configuration authority.

```
* Define users with config authority to Operations Manager.
AUTH USER(MAINT),CONFIG(Y)
AUTH USER(MAINT630),CONFIG(Y)
```

### Related reference:

Chapter 6, “Syntax for Configuration Commands,” on page 69

## Console access

### About this task

To give a specific user ID authority to access all consoles, issue the commands shown below.

**Note:** The commands in these examples give users the ability to view a console and issue commands to that console. For view only authorization, an ESM is required.

```
*Define users who can access all monitored consoles in Operations Manager
AUTH USER(MAINT),CONSOLE(*)
AUTH USER(MAINT630),CONSOLE(*)
```

To give a specific user ID authority to use the OPERATOR console in Operations Manager, issue the command shown below.

\* Define users who can access only the OPERATOR console in Operations Manager.  
AUTH USER(ADMIN),CONSOLE(OPERATOR)

## Spool access

### About this task

To give a specific user ID authority to access all the spool files on the system using Operations Manager, issue the commands shown below.

**Note:** The commands in these examples give users the ability to view another user's spool files and change the external attributes of those spool files. For view only authorization, an ESM is required.

\*Define users who can access all spool files on the system using Operations Manager  
AUTH USER(MAINT),SPOOL(\*)  
AUTH USER(MAINT630),SPOOL(\*)

To give a specific user ID authority to access the spool files owned by OPERATOR using Operations Manager, issue the command shown below.

\*Define users who can access only the spool files owned by OPERATOR using Operations Manager  
AUTH USER(ADMIN),SPOOL(OPERATOR)

## TCP/IP application interaction

### About this task

To give a specific user ID authority to interact with TCP/IP applications, issue the commands shown below. In this example, user1 is granted authority to interact with the TCP/IP applications named SYS23 and REMOTE1 and user2 is granted authority to interact with all TCP/IP applications.

\*Define users and the TCP/IP applications they can interact with Operations Manager  
AUTH USER(user1),CMDTCPA(SYS23)  
AUTH USER(user1),CMDTCPA(REMOTE1)  
AUTH USER(user2),CMDTCPA(\*)

## Authority assignments with AUTH and external security

### About this task

If you plan to use an external security mechanism (DEFOPFN ESM(Y)) for controlling Operations Manager access and authority, IBM recommends assigning authorization with the AUTH command to at least one user. Having a user authorized with AUTH will provide a method to access Operations Manager if the external security mechanism must be disabled temporarily (DEFOPFN ESM(N)).

\*Define authority to users for control access  
AUTH USER(MAINT),CONTROL(Y)  
AUTH USER(MAINT630),CONTROL(Y)

**Note:** Refer to “Step 5. (Optional) Configure an external security mechanism” on page 11 for additional information.

---

## Defining action processing servers

### About this task

To define service machines to act as action processing servers, issue the commands shown below. In this example, the names of the service machines are OPMGRS1, OPMGRS2, OPMGRS3, and OPMGRS4.

\* Define service machines to Operations Manager

```
DEFSRVM USER(OPMGRS1),XAUTOLOG(Y)
```

```
DEFSRVM USER(OPMGRS2),XAUTOLOG(Y)
```

```
DEFSRVM USER(OPMGRS3),XAUTOLOG(Y)
```

```
DEFSRVM USER(OPMGRS4),XAUTOLOG(Y)
```

**Related reference:**

“DEFSRVM” on page 51

---

## Defining system options

### About this task

After you authorize users and define action processing servers, you want to define the active options of the Operations Manager system. In this example, the options instruct Operations Manager to treat mixed case input as not case sensitive and to define the search and repeat commands in VIEWCON and VIEWLOG as /, \, and =.

\*Define options to Operations Manager

```
DEFOPTN MIXCASE(N),SYMBOLS(/\=)
```

You may also define other options that effect various components of Operations Manager. These options include:

- Activate or deactivate ESM.
- Specify attributes to apply to held lines in a user's console.
- Specify how schedules are handled during daylight saving time.
- Specify if command level authorization is used.

**Related reference:**

“DEFOPTN” on page 48

---

## Modifying the startup configuration file

After you authorize users, define action processing servers, and define system options, you can include commands in the startup configuration file to create your automated operations environment. The following areas may be defined:

- rules for console message matching
- schedules
- monitors for machines, spool files, and events
- actions to correspond to the above rules, schedules, and monitors
- holidays
- groups

Chapter 6, “Syntax for Configuration Commands,” on page 69 provides a complete list of the configuration commands. Sample commands are also provided in the file EXAMPLES CONFSAMP on the 5697J10E 2C2 minidisk.

---

## Start the Operations Manager service machines

### About this task

After you modify the startup configuration file, you must modify or vary the GOMMAIN command in the PROFILE EXEC of OPMGRM1 to specify the startup file, and then start the virtual machine. The following steps provide a guideline to modify the GOMMAIN command:

1. Logon to OPMGRM1 or the user ID running GOMMAIN.
2. When the VM READ appears, enter the following so that the PROFILE EXEC is *not* executed:

```
ACC (NOPROF
```

3. Modify the PROFILE EXEC:

```
XEDIT PROFILE EXEC
```

The following example details how to modify the PROFILE EXEC:

```
/* Sample lines to include in OPMGRM1 PROFILE EXEC */
```

```
'CP SET RUN ON'  
'ACCESS 194 D'  
'ACCESS 198 E'  
'ACCESS 400 G'  
'SET LANGUAGE ( ADD GOM USER )'  
/* Replace filename, filetype, fm with startup configuration file */  
'GOMMAIN filename filetype fm'  
'CP LOGOFF'  
EXIT
```

**Note:** Replace the *filename*, *filetype*, and *fm* with your startup configuration file. It is recommended to use OPMGRM1 CONFIG E. The value &SYSNAME can be used for the *filename* or *filetype* operand or both operands. The &SYSNAME value is replaced with the system name.

4. Save your changes and exit.
  5. Start the virtual machine:
- ```
#CP IPL CMS
```
6. When VM READ appears, press ENTER.
  7. Disconnect from the account running the GOMMAIN command:

```
#CP DISC
```

---

## Verify the installation and configuration

To test your Operations Manager installation and configuration, follow the instructions in Appendix E, "Testing Operations Manager installation and configuration," on page 155.

---

## Managing the configuration file

### About this task

There are many ways to manage your Operations Manager configuration information once Operations Manager is up and running. IBM suggests the following:

## Procedure

1. From an authorized user, link and access the disk containing the configuration file(s). For example:

```
LINK OPMGRM1 198 198 MR  
ACC 198 H
```

2. Make and save your changes to the configuration file (OPMGRM1 CONFIG, by default) or any embedded configuration file. Use the CONFIG statement to embed a configuration file within another configuration file.
3. Issue the following command to have OPMGRM1 re-access the disk and see the changed file:

```
GOMCMD OPMGRM1 CMS CMD(ACC 198 E)
```

4. Reload the configuration file. For example:

```
GOMCMD OPMGRM1 CONFIG FN(OPMGRM1),FT(CONFIG),FM(E),CLEAR(ALL)
```

### Related reference:

“CONFIG” on page 43

“CMS” on page 42



---

## Chapter 4. Syntax for System Management Commands

This chapter describes the Operations Manager system management commands and their syntax.

---

### Purpose of system management commands

Operations Manager provides system management commands that provide control of the Operations Manager environment.

---

### GOMMAIN

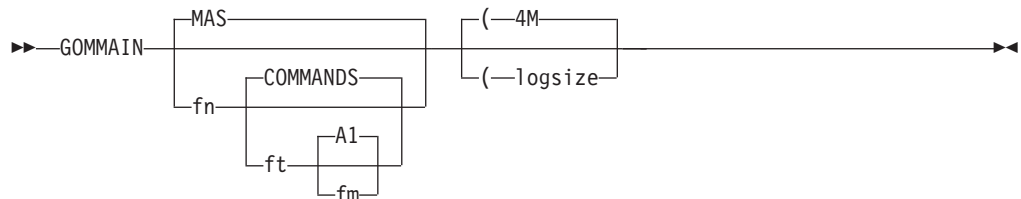


Figure 2. GOMMAIN command syntax

### Authorization

No specific authority is required for this command.

### Purpose

Issue the GOMMAIN command on the main Operations Manager service machine (OPMGRM1) to activate the framework required to run Operations Manager. The GOMMAIN module acts as a central message processing and monitoring service.

#### Related reference:

“Using the sample configuration file” on page 23

### Operands

- fn** Specifies the one- to eight-character name of the file to be processed. The default is MAS.
- ft** Specifies the one- to eight-character type of the file to be processed. The default is COMMANDS. In order to specify a file type, the file name operand must also be specified.
- fm** Specifies the one- to two-character mode of the file to be processed. The default is A1. In order to specify a file mode, the file type operand must also be specified.
- logsize** Specifies the amount of storage used for the log data space storage. The format is *nM* where *n* is between 1 and 2048. The default is 4M.

## Usage notes

1. If no file is specified on the GOMMAIN command and the default file MAS COMMAND A1 does not exist, an error will not occur. Operations Manager runs the command without a file to process.
2. GOMMAIN checks the version, release, and service level of GOMMAIN, GOMSVM, and GOMCMD to ensure compatibility. All of these components must be at the same version, release, and service level to correctly operate.
3. The value &SYSNAME can be used for the fn or ft operand or both operands. The &SYSNAME value is replaced with the system name.

## Return codes

Table 3 lists the return codes for the GOMMAIN command.

Table 3. GOMMAIN return codes

| Return code | Description                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0           | Finished correctly                                                                                                                                                                                                                                   |
| -100        | Error                                                                                                                                                                                                                                                |
| -101        | An ANCHOR is already active during startup. An attempt was made to run a GOM component in a machine that is already running another GOM component. Re-IPL the virtual machine that received the error. If the problem persists, contact IBM Support. |
| -102        | Unable to establish ANCHOR during startup. Contact IBM Support.                                                                                                                                                                                      |
| -200        | Abend                                                                                                                                                                                                                                                |
| Other       | Return code from a failed z/VM command issued by GOMMAIN. Refer to the error messages for information on the failed command                                                                                                                          |

---

## GOMCMD

►► GOMCMD—userid—command ◀◀

Figure 3. GOMCMD command syntax

## Authorization

No specific authority is required for this command.

The user issuing GOMCMD must have the authority to issue the specified command. This is accomplished through the Operations Manager AUTH command or through an External Security Manager. If the user is not authorized to issue the specified command, the user receives an error message. It is suggested that the appropriate users be authorized using an ESM or in the configuration file specified at Operations Manager startup.

### Related tasks:

“Step 5. (Optional) Configure an external security mechanism” on page 11

### Related reference:

“AUTH” on page 39



## Purpose

Issue the GOMCMD command to control and monitor Operations Manager. Any valid Operations Manager control or configuration command can be specified.

## Operands

**userid** Specifies the user ID of the Operations Manager server that is running GOMMAIN. In a default installation this is OPMGRM1.

**command**

Specifies any valid Operations Manager control or configuration command.

## Usage notes

1. GOMCMD checks the version, release, and service level of GOMMAIN, GOMSVM, and GOMCMD to ensure compatibility. All of these components must be at the same version, release, and service level to correctly operate.
2. GOMCMD can be run from an EXEC started by an Operations Manager action.
3. The return code from GOMCMD is set to the return code of the command that is running.

## Examples

This section contains two examples on how to issue the GOMCMD command. One example shows how to determine the current status of Operations Manager. The second example shows how to determine the current CMSLEVEL.

### Example 1: Status of Operations Manager

If the VM user ID OPMGRM1 is running GOMMAIN, the following command would be issued to determine the current status of Operations Manager:

```
GOMCMD OPMGRM1 STATUS
```

The output from the command is displayed on your screen.

### Example 2: Current CMSLEVEL

If the VM user ID OPMGRM2 is running GOMMAIN, the following command would be issued to determine the current CMSLEVEL on OPMGRM2:

```
GOMCMD OPMGRM2 CMS CMD(QUERY CMSLEVEL)
```

The output from the command is displayed on your screen.

## Return codes

Table 4 lists the return codes for the GOMCMD command.

Table 4. GOMCMD return codes

| Return code | Description                                                                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0           | Finished correctly                                                                                                                                                                                                                                             |
| -100        | Error                                                                                                                                                                                                                                                          |
| -101        | An ANCHOR is already active during startup. An attempt was made to run a GOM component in a machine that is already running another GOM component. Re-IPL the virtual machine that received the error. If the problem persists, contact IBM Technical Support. |
| -102        | Unable to establish ANCHOR during startup. Contact IBM Technical Support.                                                                                                                                                                                      |

Table 4. GOMCMD return codes (continued)

| Return code | Description                                                  |
|-------------|--------------------------------------------------------------|
| -200        | Abend                                                        |
| Other       | Return code from the command that was passed through GOMCMD. |

## GOMSVM

►► GOMSVM—userid ◀◀

Figure 4. GOMSVM command syntax

### Authorization

No specific authority is required for this command.

### Purpose

Issue the GOMSVM command on the Operations Manager action processing servers to activate the framework required to process actions.

### Operands

**userid** Specifies the user ID of the Operations Manager server that is running GOMMAIN. In a default installation this is OPMGRM1.

### Usage notes

GOMSVM checks the version, release, and service level of GOMMAIN, GOMSVM, and GOMCMD to ensure compatibility. All of these components must be at the same version, release, and service level to correctly operate.

### Return codes

Table 5 lists the return codes for the GOMSVM command.

Table 5. GOMSVM return codes

| Return code | Description                                                                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0           | Finished correctly                                                                                                                                                                                                                                             |
| -100        | Error                                                                                                                                                                                                                                                          |
| -101        | An ANCHOR is already active during startup. An attempt was made to run a GOM component in a machine that is already running another GOM component. Re-IPL the virtual machine that received the error. If the problem persists, contact IBM Technical Support. |
| -102        | Unable to establish ANCHOR during startup. Contact IBM Technical Support.                                                                                                                                                                                      |
| -200        | Abend                                                                                                                                                                                                                                                          |
| Other       | Return code from a failed z/VM command issued by GOMSVM. Refer to the error messages for information on the failed command                                                                                                                                     |

# GOMTDM

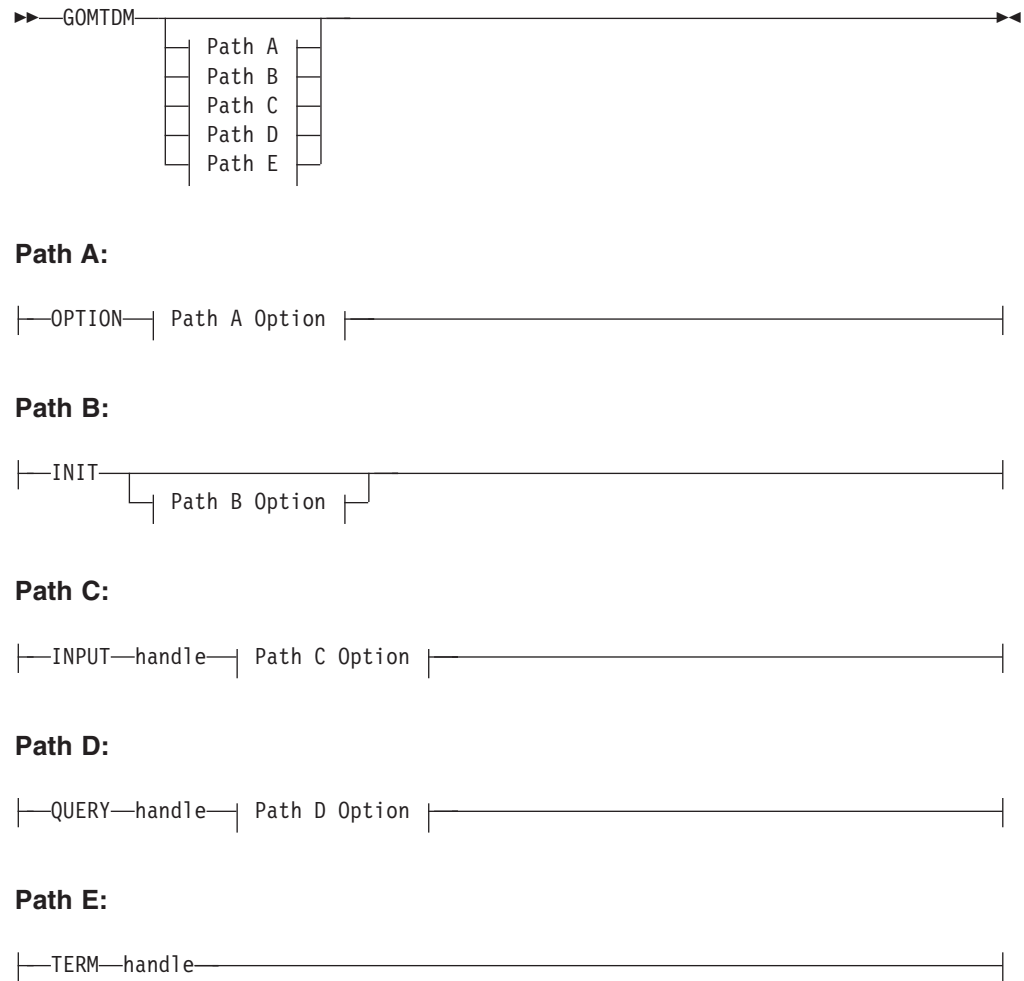


Figure 5. GOMTDM command syntax

## Authorization

Privilege class G is required for the user ID running GOMTDM. For more information on privilege classes, refer to Chapter 2, “Configuring Operations Manager,” on page 5.

## Purpose

Issue the GOMTDM command to interact with the terminal device automation infrastructure. The available paths are:

### GOMTDM OPTION

sets options for all sessions. When an option is changed it effects all sessions. The default options are MSG, TDELAY 0, QDELAY 0, STKFIFO, CCSID 0, NOTRACE, NOWAIT.

### GOMTDM INIT

starts a new session. The handle to the session is returned on the program stack in the order requested.

- GOMTDM INPUT**  
simulates keyboard input to the session. The handle is the handle returned when the session was started.
- GOMTDM QUERY**  
allows the program to examine information about the session. The handle is the handle returned when the session was started.
- GOMTDM TERM**  
terminates a session. The handle is the handle returned when the session was started.

## Operands

**handle** Specifies the handle returned when the session was started.

### Path A options

- MSG|NOMSG**  
Controls the error message output. MSG turns on error output to the screen. NOMSG turns off error message output to the screen and suppresses all error messages.
- TDELAY n**  
Sets the time delay for session output tracing to the screen. After the session input or output data is placed on the screen, the tracing is delayed for *n* seconds.
- QDELAY n**  
Sets the maximum delay time when querying the session virtual console for a string.
- STKLIFO|STKFIFO**  
Controls the stacking order for values returned on the CMS stack.
- CCSID n**  
Sets the coded character set identification. This option should be set before the first session is INITed.
- TRACE|NOTRACE**  
Controls the session tracing setting. This option allows all 3270 data streams to be traced to a disk file named LDA TRACE A1.
- WAIT|NOWAIT**  
Controls the behavior of INPUT operations. WAIT causes the processing to delay until the keyboard input inhibited and keyboard locked are inactive. NOWAIT skips the inhibited and locked checks.

### Path B options

**TRACE** Causes all input and output data to be traced to the screen. If you specify this option your screen size must be defined as a model 4 with 45 lines and 80 columns.

### Path C options

- DATA 'value'**  
Inputs a data string. The string must be enclosed in single quotes.
- CURSOR row col**  
Simulates moving the cursor to the row and column.

**KEY keyname**

Simulates a keyboard key. Valid keys are:

- ENTER
- CLEAR
- PA01-PA03
- PF01-PF24
- FT (forward tab)
- BT (backward tab)
- UP (cursor up)
- DOWN (cursor down)
- LEFT (cursor left)
- RIGHT (cursor right)
- NEWLINE
- HOME
- ERASEEOF

**Path D options**

**CURSOR** Returns the row and column of the current cursor position on the stack.

**VALUE 'string'**

Returns the row and column of the string on the screen on the stack.

Return code 0 indicates the string was located and the row and column are on the stack. Return code 4 indicates the string was not found and no value was placed on the stack.

**IMAGEFIL fn**

Places a copy of the session screen in the CMS file fn SCREEN A1.

**VERSION**

Returns the build date and time on the stack in the form YYYY/MM/DD hh:mm.

**Usage notes**

1. Only one path may be specified on a GOMTDM command.
2. Only one option may be specified on a GOMTDM command.

**Restrictions**

1. A maximum of eight (8) simultaneous sessions are allowed from a virtual machine.

**Return codes**

Table 6 lists the return codes for the GOMTDM command.

*Table 6. GOMTDM return codes*

| Return code | Description                                      |
|-------------|--------------------------------------------------|
| 0           | Finished correctly                               |
| 4           | QUERY function did not find the string           |
| 8           | Syntax or parameter error                        |
| 12          | Abend                                            |
| 32          | Could not set ANCHOR value. Contact IBM Support. |

Table 6. GOMTDM return codes (continued)

| Return code | Description                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------|
| Other       | Return code from a failed z/VM command issued by GOMTDM. Refer to the error messages for information on the failed command. |

---

## Chapter 5. Syntax for Control Commands

This chapter describes the Operations Manager control commands and their syntax.

---

### Purpose of control commands

Control commands allow you to control processes relating to Operations Manager. With a control command, you can perform various tasks such as check the status of a system, assign and revoke authorities, and define and delete service machines from Operations Manager's use.

You can type a control command at the console and it runs immediately or you can put control commands into a file so that each time you start Operations Manager, the same checks, authorities, and service machines are in effect and you do not have to reenter them.

Some of these control commands are also valid as configuration commands.

**Related reference:**

Chapter 3, "Creating a startup configuration file," on page 21

Chapter 6, "Syntax for Configuration Commands," on page 69

---

### Authorization for control commands

You must use an ESM or the AUTH command to give a specific user the authority to issue control commands. To give users access to specific control commands, an ESM is required.

**Related tasks:**

"Step 5. (Optional) Configure an external security mechanism" on page 11

---

## AUTH

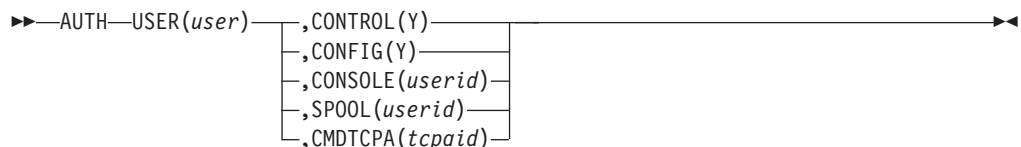


Figure 6. AUTH command syntax

### Authorization

The user issuing this command must be authorized for control commands.

### Purpose

Use the AUTH command to authorize a VM user to issue certain Operations Manager commands, view the consoles and spooled output of specific users, and interact with selected TCP/IP applications. If you are using an ESM, any AUTH statements are ignored.

**Related tasks:**

“Step 5. (Optional) Configure an external security mechanism” on page 11

## Operands

**USER(user)**

Identifies the user being authorized for functions.

**CONTROL(Y)**

Specifies that the user is authorized to use Operations Manager control commands.

**CONFIG(Y)**

Specifies that the user is authorized to use Operations Manager configuration commands.

**CONSOLE(userid)**

Specifies that the user is authorized to view and interact with the console belonging to userid or a combined view named userid. Userid may be a wildcard pattern. The wildcard rules are explained in the DEFRULE command.

**SPOOL(userid)**

Specifies that the user is authorized to view spooled output belonging to userid. Userid may be a wildcard pattern. The wildcard rules are explained in the DEFRULE command.

**CMDTCPA(tcpaid)**

Specifies that the user is authorized to interact with the TCP/IP application defined as tcpaid. The tcpaid variable may be a wildcard pattern. The wildcard rules are explained in the DEFRULE command.

**Related reference:**

“DEFRULE” on page 94

## Operand value lengths

Table 7 lists the operand value lengths for the AUTH command.

*Table 7. AUTH operand value lengths*

| Operand | Length |
|---------|--------|
| CMDTCPA | 1-8    |
| CONTROL | 1      |
| CONFIG  | 1      |
| CONSOLE | 1-8    |
| USER    | 1-8    |
| SPOOL   | 1-8    |

## Usage notes

1. Each AUTH command may have only one operand (CONFIG, CONTROL, CONSOLE, SPOOL, or CMDTCPA). Issue multiple AUTH commands to give a user additional access.
2. When a user is authorized for CONTROL, they are also authorized for CONFIG when one of the following is also true:
  - a. ESM is set to N on the DEFOPTN command, or



- b. ESM is set to Y and AUTHCMD is set to N on the DEFOPTN command.
- 3. A user authorized for CONTROL is *not* authorized for CONFIG if ESM is set to Y and AUTHCMD is set to Y on the DEFOPTN command.

**Related reference:**

“DEFOPTN” on page 48

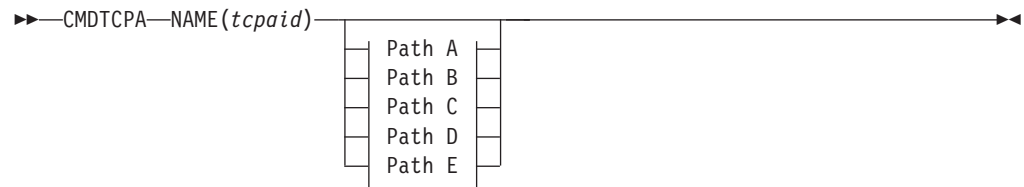
## Return codes

Table 8 lists the return codes for the AUTH command.

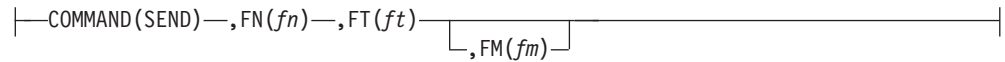
Table 8. AUTH return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

## CMDTCPA



**Path A:**



**Path B:**



**Path C:**



**Path D:**



Figure 7. CMDTCPA command syntax

## Authority

The user issuing this command must be authorized for interaction with the TCP/IP application *tcpaid*.

## Purpose

Issue the CMDTCPA command to send a command request to a TCP/IP application.

## Operands

### NAME(*tcpaid*)

Specifies the name of the TCP/IP application. The value must match the NAME operand's value of a DEFTCPA command.

### Path **A, B, C, D**

This path sends requests to a TCP/IP application using TCPAPPL(GOMRSGW).

### COMMAND(**SEND**)

Sends data from a file, operand, or REXX variable.

**FN(*fn*)** Specifies the filename of the file to send.

**FT(*ft*)** Specifies the filetype of the file to send.

**FM(*fm*)** Specifies the filemode of the file to send. The default is any accessed CMS disk.

### FILE(*fn ft fm*)

Specifies the filename, filetype, and filemode of the file to send. Values for *fn* and *ft* are required. If *fm* is omitted, the default is any accessed CMS disk.

### DATA(*data*)

Specifies the data to send.

### REXXVAR(*varname*)

Specifies a REXX variable name containing the data to send.

## Usage notes

1. Processing applications and supported commands:

### GOMRSIF

No commands supported.

### GOMRSYL

No commands supported.

### GOMRSGW

Supports the SEND command.

2. The operands on the various paths are mutually exclusive with operands on other paths.
3. No code page translation is done before the data is sent. If code page translation is required, a sample EXEC is provided that translates EBCDIC to ASCII. Refer to the EBC2ASC SAMPEXEC information in Appendix A, "Sample files," on page 143.

---

## CMS

▶▶—CMS—CMD(*command*)—————▶▶

Figure 8. CMS command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Use the CMS command to present a command to CMS for processing on the main Operations Manager server (OPMGRM1) and display the results.

## Operands

### CMD (command)

Specifies the CMS command to process.

## Operand value lengths

Table 9 lists the operand value lengths for the CMS command.

Table 9. CMS operand value lengths

| Operand | Length |
|---------|--------|
| CMD     | 1-120  |

## Usage notes

None.

## Return codes

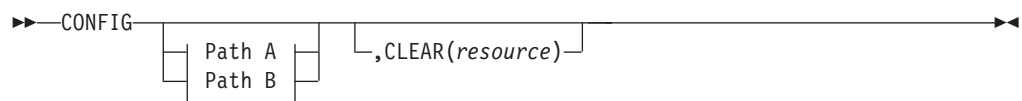
Table 10 lists the return codes for the CMS command.

Table 10. CMS return codes

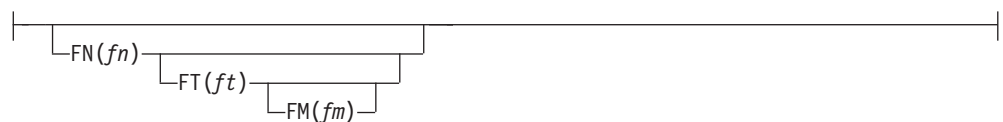
| Return code | Description                                             |
|-------------|---------------------------------------------------------|
| 0           | Finished correctly                                      |
| -100        | Error                                                   |
| Other       | Error code from a called interface, command, or service |

---

## CONFIG



### PATH A:



## PATH B:



Figure 9. CONFIG command syntax

## Authorization

The user issuing this command must be authorized for configuration or control commands.

## Purpose

Issue the CONFIG command to process the contents of a file containing Operations Manager commands.

## Operands

**FN(fn)** Specifies the file name of the file to be processed. The default is MAS.

**FT(ft)** Specifies the file type of the file to be processed. The default is COMMANDS.

**FM(fm)** Specifies the file mode of the file to be processed. The default is A1. This is the file mode on OPMGRM1.

### **FILE (fn ft fm)**

Specifies the file name, file type, and file mode to be processed. The default is MAS COMMANDS A1.

### **CLEAR(resource)**

Specifies that all existing instances for the specified resource type are removed before processing the specified file. The default is to not remove any resource. Multiple resources may be specified in a comma separated list.

The valid resources are:

- ALL (clears all of the following resources)
- RULE
- MACHINE
- SCHEDULE
- ACTION
- GROUP
- DATE
- SPOOL
- EVENT
- PAGE
- IDLE

## Operand value lengths

Table 11 lists the operand value lengths for the CONFIG command.

Table 11. CONFIG operand value lengths

| Operand | Length |
|---------|--------|
| CLEAR   | 1-120  |
| FILE    | 1-20   |
| FM      | 1-2    |
| FN      | 1-8    |
| FT      | 1-8    |

## Usage notes

1. The CONFIG command can be used within a configuration file containing Operations Manager commands, allowing you to embed other configuration files. You can nest the CONFIG command within included files. There is no restriction on the depth of nested files. You can use this feature in the startup configuration file for the GOMMAIN command to organize your configuration data into multiple files.
2. If the FT(ft) operand is entered, the FN operand must be entered.
3. If the FM(fm) operand is entered, the FT operand must be entered.
4. The operands on one path are mutually exclusive with operands on other paths.
5. The value &SYSNAME can be used for either or both of the *fn* or *ft* values in the FN, FT, and FILE operands. The &SYSNAME value is replaced with the system name.

## Return codes

Table 12 lists the return codes for the CONFIG command.

Table 12. CONFIG return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## CP

▶▶—CP—CMD(*command*)—————▶▶

Figure 10. CP command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Use the CP command to present a command to CP for processing on OPMGRM1 and display the results.

## Operands

### **CMD (command)**

Specifies the CP command to process.

## Operand value lengths

Table 13 lists the operand value lengths for the CP command.

*Table 13. CP operand value lengths*

| Operand | Length |
|---------|--------|
| CMD     | 1-120  |

## Usage notes

None.

## Return codes

Table 14 lists the return codes for the CP command.

*Table 14. CP return codes*

| Return code | Description                                             |
|-------------|---------------------------------------------------------|
| 0           | Finished correctly                                      |
| -100        | Error                                                   |
| Other       | Error code from a called interface, command, or service |

---

## DEFDSPS

▶▶ DEFDSPTS—USER(*user*)—,SIZE(*size*)—▶▶

*Figure 11. DEFDSPTS command syntax*

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DEFDSPS command to define the amount of storage used to capture the data for a monitored user's console or combined view.

## Operands

### **USER(user)**

Specifies a name or a wildcard pattern to match against a user's name or a view name. The wildcard rules are explained in the DEFRULE command.

### SIZE(size)

Specifies the size used to create the data space that is used to contain monitored users output. The size may be specified in bytes or megabytes.

If bytes are used, the valid range is 1-1073741824. The input is rounded up to the nearest 1048576 and the startup default is 1048576.

If megabytes are used, the size must be suffixed by M. The startup default is 1M.

#### Related reference:

“DEFRULE” on page 94

“DEFVIEW” on page 55

## Operand value lengths

Table 15 lists the operand value lengths for the DEFDSPS command.

Table 15. DEFDSPS operand value lengths

| Operand | Length |
|---------|--------|
| SIZE    | 1-10   |
| USER    | 1-8    |

## Usage notes

1. If you have multiple DEFDSPS commands, the order of the commands is important; each occurrence does not override the previous command. In the following example, the second command does not override the first command:

```
DEFDSPS USER(USER1),SIZE(2M)
DEFDSPS USER(USER*),SIZE(3M)
DEFDSPS USER(*),SIZE(1M)
```

In this example, USER1's console has a 2M data space, the consoles of all other user IDs starting with USER have a 3M data space, and all other consoles have a 1M data space.

2. The processing of a DEFDSPS command does not allocate a data space. A data space is allocated when the initial input from a monitored user's console is received by Operations Manager.
3. The maximum number of data spaces and the maximum amount of storage for all data spaces allocated by Operations Manager is limited by the CP directory. The CP directory's XCONFIG ADDRSPACE statement controls the maximum number of data spaces and the maximum amount of storage for all data spaces. If a large number of data spaces are expected to exceed 16MB in size, examine the CP directory XCONFIG ADDRSPACE statement and adjust the TOTSIZE parameter as needed.

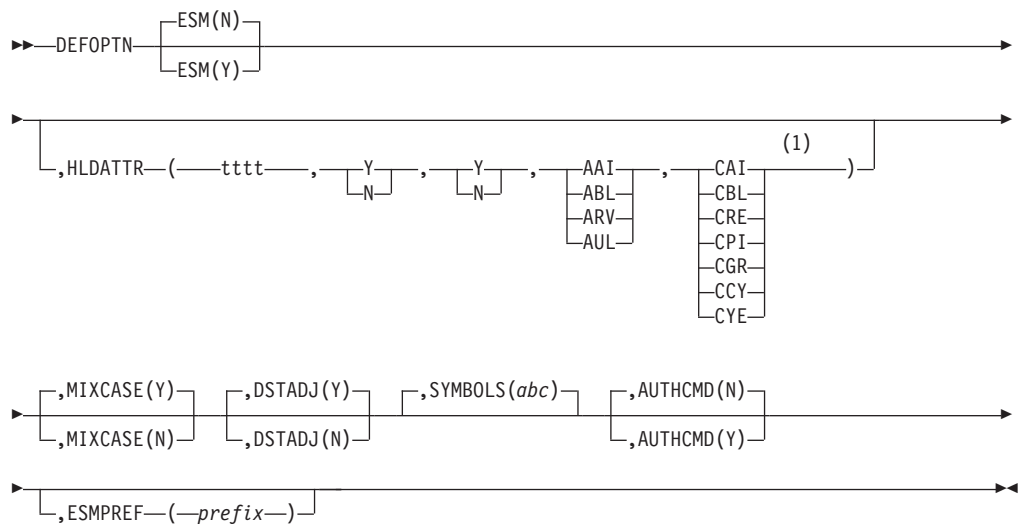
## Return codes

Table 16 lists the return codes for the DEFDSPS command.

Table 16. DEFDSPS return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

# DEFOPTN



## Notes:

- 1 You must specify all five values for HLDATTR with a comma separating each value.

Figure 12. DEFOPTN command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DEFOPTN command to change the active options.

## Operands

### ESM(Y|N)

Activates or deactivates the usage of an External Security Manager to check for authorization. The startup default is N.

### HLDATTR(tttt,a,h,xat,clr)

Specifies a time and a set of attributes to apply to held lines in a user's console. The startup default is 0060,N,Y,AAI,CAI.

- tttt. Specifies a number of seconds. Must be a four digit number. If a "held" line in a user's console has not been responded to within the number of seconds specified, the following attributes are applied to the line:
  - a. Specifies if the audible alarm is sounded. Values are Y or N.
  - h. Specifies if the line is highlighted. Values are Y or N.
  - xat. Specifies the extended attributes applied to the line. Valid values are:
    - AAI. Leave as is.
    - ABL. Blinking.



- ARV. Reverse video.
- AUL. Underline.
- c1r. Specifies the color applied to the line. Valid values are:
  - CAI. Leave as is.
  - CBL. Blue.
  - CRE. Red.
  - CPI. Pink.
  - CGR. Green.
  - CCY. Cyan.
  - CYE. Yellow.
  - CWH. White.

#### **MIXCASE(Y|N)**

Specifies if mixed case data is treated as case sensitive. This option effects command input and user console input handling. The default is Y. This operand applies to:

- text matching for rules
- input commands from the OPMGRM1 console or using the GOMCMD command interface
- input commands in a file, such as the startup configuration file or specified on the CONFIG command
- text search operations executed while viewing the results from the VIEWLOG, VIEWCON, or VIEWSPL commands.

#### **DSTADJ(Y|N)**

Specifies how the schedule processing time is handled when a change in time zone offset (such as daylight savings time) is detected. The default is Y.

- DSTADJ(Y) causes Operations Manager to trigger schedules (defined with DEFSCHD) strictly by wall clock time.

When a time zone change causes the local time to move forward, schedules, which would have occurred within the period of time represented by the time zone change, are not executed during the time change. They are executed at their next scheduled execution time.

When a time zone change causes the local time to move backward, schedules, which should occur within the period of time represented by the time zone change, are executed twice since the specified time occurs twice.

- DSTADJ(N) causes Operations Manager to trigger schedules (defined with DEFSCHD) based on the time zone change.

When a time zone change causes the local time to move forward, schedules, which would have been eligible for execution within the period of time represented by the time zone change, are immediately scheduled for execution at the start of the time change.

When a time zone change causes the local time to move backward, schedules, which are eligible for execution within the period of time represented by the time zone change, are executed only once. They are not repeated.

#### **SYMBOLS(abc)**

Specifies the symbols to use in various components. Each symbol is one

character and is positional in the list. If you do not wish to change a symbol, you must input a space. The symbols in the list and the defaults are:

- a — Forward search command character in VIEWCON, VIEWLOG, and VIEWSP. The default is forward slash (/).
- b — Backward search command character in VIEWCON, VIEWLOG, and VIEWSP. The default is backward slash (\).
- c — Repeat command character. The default is the equal sign (=).

#### **AUTHCMD(Y|N)**

Specifies command level authorization should be used instead of the CONTROL and CONFIG level authorization. If command level authorization is used, ESM authorization checking must be active. If AUTHCMD(Y) is specified without also specifying ESM(Y), it has no effect. The startup default is N.

#### **ESMPREF(prefix)**

Specifies the customized portion of the prefix for the external security manager profiles for this instance of Operations Manager.

- If no value is specified, Operations Manager looks for profiles with a prefix of RKT.OPM.
- If you specify a value for ESMPREF, Operations Manager looks for profiles with a prefix of GOM.*esmprefix*, where *esmprefix* is the value you specified on ESMPREF. For example, if you specify ESMPREF(OPSMGR) in the Operations Manager configuration file, Operations Manager will look for profiles with the prefix GOM.OPSMGR.

#### **Related tasks:**

“Step 5. (Optional) Configure an external security mechanism” on page 11

## **Operand value lengths**

Table 17 lists the operand value lengths for the DEFOPTN command.

*Table 17. DEFOPTN operand value lengths*

| <b>Operand</b> | <b>Length</b> |
|----------------|---------------|
| AUTHCMD        | 1             |
| DSTADJ         | 1             |
| ESM            | 1             |
| ESMPREF        | 1-48          |
| HLDATTR        | 16            |
| MIXCASE        | 1             |
| SYMBOLS        | 1-16          |

## **Usage notes**

1. Options not specified are not changed.
2. To hold lines on a user’s console, you must specify the HLD value on the INPUT operand of the DEFACTN command.
3. If MIXCASE(Y) is set, many commands and operands automatically convert to uppercase for processing. For example, if the following command is input:  

```
defsmn name(s1),action(s1act),usage(081-100)
```

It is converted to the following for processing:

DEFSMON NAME(S1),ACTION(S1ACT),USAGE(081-100)

The following command-operand combinations do *not* automatically convert:

- ALTRCON EXCLUDE
- ALTRCON MATCH
- CMDTCPA DATA
- CMDTCPA REXXVAR
- DEFACTN COMMAND
- DEFEMON PARM
- DEFMMON PARM
- DEFOPTN SYMBOLS
- DEFRULE EXCLUDE
- DEFRULE MATCH
- DEFRULE PARM
- DEFSCHD PARM
- DEFSMON PARM
- LOGTEXT TEXT

**Related reference:**

“DEFACTN” on page 71

## Return codes

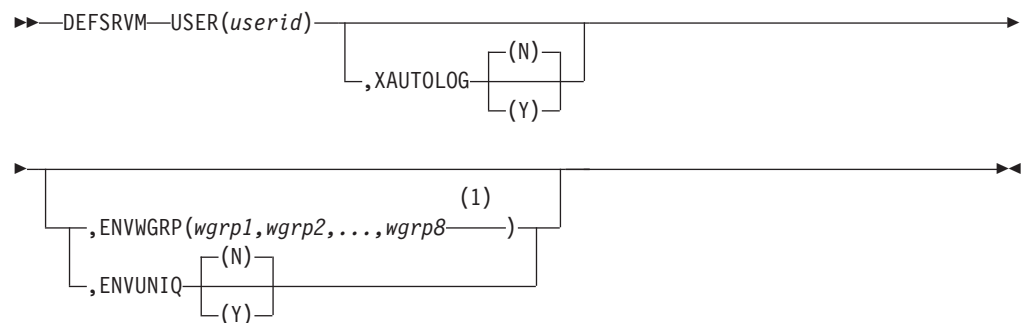
Table 18 lists the return codes for the DEFOPTN command.

Table 18. DEFOPTN return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DEFSRVM



**Notes:**

- 1 Specify up to eight (8) values with each value separated by a comma.

Figure 13. DEFSRVM command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DEFSRVM command to define an action processing service machine. The service machine is available to process actions assigned to it by OPMGRM1, or the server running GOMMAIN.

## Operands

### USER(*userid*)

Specifies the VM user ID that is setup to act as an action processing server running GOMSVM.

### XAUTOLOG(Y|N)

Specifies if at startup OPMGRM1 is to issue the CP XAUTOLOG command to auto-log the action processing server. This operand is optional, but highly recommended. The default is XAUTOLOG(N).

### ENVUNIQ(Y|N)

Specifies if the service machine only processes an action (specified by DEFACTN) where the ENV() operand value matches the *userid* of the service machine. ENVUNIQ(Y) prevents the service machine from processing an action where ENV(SVM) is specified. The default is ENVUNIQ(N).

### ENWGRP(*wgrp1,wgrp2,...,wgrp8*)

Specifies that the service machine only processes an action where the ENV() operand value matches one of the work groups. A service machine may participate in up to eight (8) work groups. This prevents the service machine from processing an action where ENV(SVM) is specified.

### Related reference:

“DEFACTN” on page 71

## Operand value lengths

Table 19 lists the operand value lengths for the DEFSRVM command.

Table 19. DEFSRVM operand value lengths

| Operand  | Length |
|----------|--------|
| ENWGRP   | 1-72   |
| ENVUNIQ  | 1      |
| USER     | 1-8    |
| XAUTOLOG | 1      |

## Usage notes

1. If XAUTOLOG(Y) is specified, the VM user ID running GOMMAIN must have PRIV A or B to operate correctly. For more information on privilege classes, refer to Chapter 2, “Configuring Operations Manager,” on page 5.
2. ENVUNIQ and ENWRGP are mutually exclusive.

## Return codes

Table 20 lists the return codes for the DEFSRVM command.

Table 20. DEFSRVM return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DEFTCPA

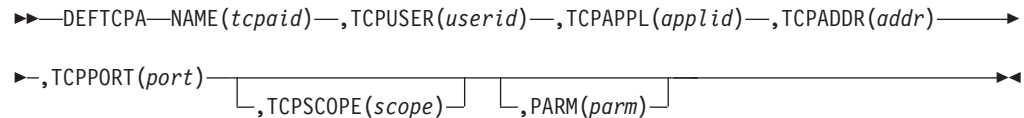


Figure 14. DEFTCPA command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DEFTCPA command to establish a TCP/IP application and connection information. This connection may be used to send data from remote systems to Operations Manager or for Operations Manager to send data to other systems.

When Operations Manager receives data, the data is processed as if it were a console; rules are processed and any actions matching the criteria are issued. The data is viewable by issuing the VIEWCON and VIEWLOG commands.

When Operations Manager sends the data (using CMDTCPA), it is sent exactly as passed to Operations Manager. If you need to translate the data to another code page, this must be done before the CMDTCPA is issued.

### Related reference:

Appendix D, “Operations Manager and TCP/IP interaction,” on page 149

## Operands

### NAME(tcpaid)

Specifies the name of this TCP/IP connection definition.

### TCPUSER(userid)

Specifies the name of the VM userid providing the TCP/IP communications support.

### TCPAPPL(applid)

Specifies the name of the Operations Manager processing program. See “Usage notes” on page 54 for the available processing applications.

### TCPADDR(addr)

Specifies the TCP/IP address for the application.

For IPV4 addresses, the format is in dotted decimal notation: (n.n.n.n).

For IPV6 addresses, the format is in hex notation: (hhh:hhh:...:hhh).

**TCPPORT(port)**

Specifies the TCP/IP port on which Operations Manager listens for remote input or sends to remote locations. Leading zeros are not required. The format can range from n to nnnn.

If you are using DEFTCPA to receive syslog data from Linux or send syslog data to Linux, you will typically specify port number 514, since this is often the default port that Linux routes syslog data.

If you are using DEFTCPA to receive data from z/VM systems, specify any available port. This must match the port number specified on the GOMRSIF command.

**TCPSCOPE(scope)**

Specifies the TCP/IP address scope for certain IPV6 networks.

**PARM(parm)**

Specifies the parameter passed to the processing program.

## Operand value lengths

Table 21 lists the operand value lengths for the DEFTCPA command.

*Table 21. DEFTCPA operand value lengths*

| Operand  | Length |
|----------|--------|
| NAME     | 1-8    |
| PARM     | 1-120  |
| TCPADDR  | 2-37   |
| TCPAPPL  | 1-8    |
| TCPPORT  | 1-5    |
| TCPSCOPE | 1-10   |
| TCPUSER  | 1-8    |

## Usage notes

1. The available processing applications and the supported PARM values are:

**GOMRSIF**

Provides support to receive and process a file sent from a local or remote z/VM system and filter the file contents against rules. No PARM is used. The data can be sent using the GOMRSIF EXEC.

**GOMRSYL**

Provides support to process input from a remote syslog and filter against rules. PARM describes the following fields:

- Position 01-08 — USER to be passed for rule processing. If you specify blanks, the internal program name is used.
- Position 09-16 — File name of the translation table. The file type is TABLE. This file may be located on any accessed disk. Sample translation tables that perform ASCII to EBCDIC translation or no translation are provided on the samples disk (5697J10E 2C2 or associated SFS directory.) If you specify blanks, the internal ACSII to EBCDIC table is used.

- Position 17-20 — Character encoding scheme (IS08 or UTF8). If you specify blanks, IS08 is used.

- Position 21 — Displays sender's address (Y, S, or N):

**Y** Displays the address in a fixed, full format and with no leading zero suppression:

- IPV4 shows the TCP/IP address as nnn.nnn.nnn.nnn.
- IPV6 shows the TCP/IP address as hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh.

For example:

- 88.168.1.1 displays as 088.168.001.001
- 2001:db8:0:0:1:0:0:1 displays as 2001:0db8:0000:0000:0001:0000:0000:0001

**S** Displays the address in a shorthand format and with leading zero suppression:

- IPV4 shows the TCP/IP address as n.n.n.n
- IPV6 shows the TCP/IP address as h::h

For example:

- 88.168.1.1 displays as 88.168.1.1
- 2001:db8:0:0:1:0:0:1 displays as 2001:db8::1:0:0:1

**N** Does not display the address. If you specify a blank or null, "N" is used.

#### **GOMRSGW**

Provides support for a general TCP/IP writer.

- Position 01-03 — Protocol to be used for the connection (TCP or UDP). If you specify blanks, TCP is used.

2. Depending on your TCP/IP configuration, you may need to authorize OPMGRM1 (and any other user ID running GOMMAIN) to use the ports specified in your DEFTCPA statements. Refer to the PORT or OBEY statements in *z/VM TCP/IP Planning and Customization (SC24-6125)* for information on enabling an application to use specific ports.

**Related reference:**

"GOMRSIF" on page 152

## **Return codes**

Table 22 lists the return codes for the DEFTCPA command.

*Table 22. DEFTCPA return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## **DEFVIEW**

▶▶—DEFVIEW—NAME(*viewid*)—,USER(*user*)—◀◀

Figure 15. DEFVIEW command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DEFVIEW command to define a combined view of multiple users' consoles. This command adds users to a combined view. Issue the VIEWCON command to view the output.

**Related reference:**

“VIEWCON” on page 127

## Operands

**NAME(viewid)**

Specifies a name for the view.

**USER(user)**

Specifies a user ID or a wildcard pattern to match against a user's ID. Output from user IDs matching the pattern is added to the view. The wildcard rules are explained in the DEFRULE command.

**Related reference:**

“DEFRULE” on page 94

## Operand value lengths

Table 23 lists the operand value lengths for the DEFVIEW command.

Table 23. DEFVIEW operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |
| USER    | 1-8    |

## Usage notes

1. If you want to add multiple user IDs or a wildcard pattern to a combined view, you must specify one DEFVIEW command for each user ID. For example, to add users USER1, USER2, and USER3 into the combined view named MYVIEW, specify three DEFVIEW commands.

```
DEFVIEW NAME(MYVIEW),USER(USER1)
DEFVIEW NAME(MYVIEW),USER(USER2)
DEFVIEW NAME(MYVIEW),USER(USER3)
```

2. The output from a user's console may be added to multiple combined views. The output from a combined view cannot be included in other combined views.
3. Issue the VIEWCON command to view the output from a combined view. To issue a command to a user's console defined in the combined view, you must put the target ID in front of the command. For example, USER1 *command*.

**Related reference:**

“VIEWCON” on page 127



## Return codes

Table 24 lists the return codes for the DEFVIEW command.

Table 24. DEFVIEW return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELDSPS

▶▶—DELDSPS—USER(*user*)—————▶▶

Figure 16. DELDSPS command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DELDSPS command to delete the size definition for a previously defined data space. The default size is used instead. This command does not delete any previously defined data spaces.

### Related reference:

“DEFDSPS” on page 46

## Operands

### USER(*user*)

Specifies a user name or a wildcard pattern to delete.

## Operand value lengths

Table 25 lists the operand value lengths for the DELDSPS command.

Table 25. DELDSPS operand value lengths

| Operand | Length |
|---------|--------|
| USER    | 1-8    |

## Usage notes

1. The value of the USER operand must completely match a previously defined value.

## Return codes

Table 26 lists the return codes for the DELDSPS command.

Table 26. DELDSPS return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |

Table 26. DELDSPS return codes (continued)

| Return code | Description |
|-------------|-------------|
| -100        | Error       |

## DELSRVM

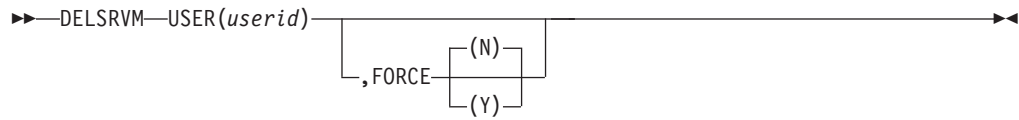


Figure 17. DELSRVM command syntax

### Authorization

The user issuing this command must be authorized for control commands.

### Purpose

Issue the DELSRVM command to delete an action processing service machine from the current Operations Manager configuration.

**Related reference:**

“DEFSRVM” on page 51

### Operands

**USER(userid)**

Specifies the user ID of the action processing server running GOMSVM that you want to delete.

**FORCE(Y|N)**

Specifies if the deletion of the service machine is forced to ensure that the deletion occurs. Forced deletion also issues a CP FORCE command to log off the user ID. The default is FORCE(N).

### Operand value lengths

Table 27 lists the operand value lengths for the DELSRVM command.

Table 27. DELSRVM operand value lengths

| Operand | Length |
|---------|--------|
| FORCE   | 1      |
| USER    | 1-8    |

### Usage notes

1. If FORCE(Y) is specified, the VM user ID running GOMMAIN must have PRIV A to operate correctly. For more information on privilege classes, refer to Chapter 2, “Configuring Operations Manager,” on page 5.
2. If the action processing server is busy, the definition is not deleted unless the FORCE(Y) operand is specified.

3. If FORCE(N) is specified and the action processing server is busy, a message is issued and the command is ignored. The action processing server is not deleted even when it becomes idle.

## Return codes

Table 28 lists the return codes for the DELSRVM command.

Table 28. DELSRVM return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELTCPA

▶▶—DELTCPA—NAME(*tcpaid*)—————▶▶

Figure 18. DELTCPA command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DELTCPA command to terminate a TCP/IP application defined to Operations Manager.

**Related reference:**

“DEFTCPA” on page 53

## Operands

**NAME(*tcpaid*)**

Specifies the name of the TCP/IP application to terminate.

## Operand value lengths

Table 29 lists the operand value lengths for the DELTCPA command.

Table 29. DELTCPA operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 30 lists the return codes for the DELTCPA command.

Table 30. DELTCPA return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELVIEW

▶▶—DELVIEW—NAME(*viewid*)—,USER(*user*)—▶▶

Figure 19. DELVIEW command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the DELVIEW command to delete users from a combined view.

**Related reference:**

“DEFVIEW” on page 55

## Operands

**NAME(*viewid*)**

Specifies the view name.

**USER(*user*)**

Specifies a user name or a wildcard pattern to delete.

## Operand value lengths

Table 31 lists the operand value lengths for the DELVIEW command.

Table 31. DELVIEW operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |
| USER    | 1-8    |

## Usage notes

1. The value of the NAME and USER operand must completely match a previously defined value.
2. Only the unique combination of NAME and USER are removed. Other NAME and USER combinations are not removed.

## Return codes

Table 32 lists the return codes for the DELVIEW command.

Table 32. DELVIEW return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## QUIT

►►—QUIT—◄◄

Figure 20. QUIT command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the QUIT command to stop GOMMAIN. There are no operands and none are allowed.

## Usage notes

1. QUIT, STOP, and SHUTDOWN are equivalent commands. Issue any of these commands to stop GOMMAIN execution.

## Return codes

Table 33 lists the return codes for the QUIT command.

Table 33. QUIT return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## REVOKE

►►—REVOKE—USER(*user*)—  
    ,CONTROL(*Y*)—  
    ,CONFIG(*Y*)—  
    ,CONSOLE(*userid*)—  
    ,SPOOL(*userid*)—  
    ,CMDTCPA(*tcpaid*)—◄◄

Figure 21. REVOKE command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the REVOKE command to remove authorization from a VM user to issue certain commands, view the consoles or spooled output of certain users, and interact with certain TCP/IP applications. If you are using an ESM, do not issue the REVOKE command.

### Related tasks:

“Step 5. (Optional) Configure an external security mechanism” on page 11

## Operands

### USER(*user*)

Identifies the user whose authority is being revoked.

### CONTROL(*Y*)

Specifies that the user’s authority to use Operations Manager control commands is being revoked.

### CONFIG(*Y*)

Specifies that the user’s authority to use Operations Manager configuration commands is being revoked.

### CONSOLE(*userid*)

Specifies that the user's authority to view and interact with the console belonging to *userid* is being revoked.

### SPOOL(*userid*)

Specifies that the user’s authority to view the spooled output belonging to *userid* is being revoked.

### CMDTCPA(*tcpaid*)

Specifies that the user's authority to interact with the TCP/IP application, defined as *tcpaid*, is being revoked.

## Operand value lengths

Table 34 lists the operand value lengths for the REVOKE command.

Table 34. REVOKE operand value lengths

| Operand | Length |
|---------|--------|
| CMDTCPA | 1-8    |
| CONFIG  | 1      |
| CONSOLE | 1-8    |
| CONTROL | 1      |
| SPOOL   | 1-8    |
| USER    | 1-8    |

## Usage notes

1. Each REVOKE command may have only one operand (CONFIG, CONTROL, CONSOLE, SPOOL, or CMDTCPA).
2. The *userid* value of the CONSOLE operand must match the value specified in the AUTH command.

3. The userid value of the SP00L operand must match the value specified in the AUTH command.
4. The tcpaid value of the CMDTCPA operand must match the value specified in the AUTH command.

**Related reference:**

“AUTH” on page 39

## Return codes

Table 35 lists the return codes for the REVOKE command.

*Table 35. REVOKE return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## SHUTDOWN

»—SHUTDOWN—◀◀

*Figure 22. SHUTDOWN command syntax*

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the SHUTDOWN command to stop GOMMAIN. There are no operands and none are allowed.

## Usage notes

1. QUIT, STOP, and SHUTDOWN are equivalent commands. Issue any of these commands to stop GOMMAIN execution.

## Return codes

Table 36 lists the return codes for the SHUTDOWN command.

*Table 36. SHUTDOWN return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

# STATUS

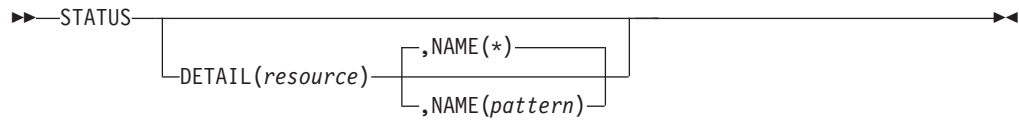


Figure 23. STATUS command syntax

## Authorization

The user issuing this command must be authorized for configuration or control commands.

## Purpose

Issue the STATUS command to show the current status of GOMMAIN processing. If the DETAIL operand is not present a summary is outputted. If the DETAIL operand is present additional information about each specified resource is outputted.

## Operands

### DETAIL(resource)

Requests additional information for the specified resource definitions.

The valid resources are:

- RULE — Displays the status of the defined rules.
- MACHINE — Displays the status of the monitored virtual machines.
- SCHEDULE — Displays the status of the defined schedules.
- ACTION — Displays the status of the defined actions.
- ACTIONQ — Displays the actions queued for execution.
- GROUP — Displays the status of the defined groups.
- DATE — Displays the status of the defined holidays.
- SERVICE — Displays the status of the defined action processing servers.
- SPOOL — Displays the status of the defined spool usage monitors and how many times they were executed.
- SPOOLUSR — Displays the top ten largest spool files and the top ten users with the most spool files.
- AUTH — Displays the status of authorized users.
- IUCV — Displays the status of the active IUCV connections.
- DATASP — Displays the status of the defined data space size creation templates.
- VIEW — Displays the status of the combined views.
- CONSOLE — Displays the status of the recording consoles.
- TCPAPPL — Displays the status of the TCP/IP applications.
- OPTION — Displays the status of the current options settings.
- EVENT — Displays the status of the defined event monitors.
- PAGE — Displays the status of the defined page space monitors.
- IDLE — Displays the status of the defined idle monitors.



**NAME(pattern)**

Lists information for resources names which match the pattern. Wildcard patterns are allowed. The wildcard pattern rules are explained in the DEFULE command.

**Related reference:**

“DEFRULE” on page 94

## Operand value lengths

Table 37 lists the operand value lengths for the STATUS command.

Table 37. STATUS operand value lengths

| Operand | Length |
|---------|--------|
| DETAIL  | 1-8    |
| NAME    | 1-8    |

## Examples

Figure 24 shows summary output from the STATUS command.

```
STATUS
2013-10-24 16:04:59 GOMCMD0899I OPERATIONS MANAGER V1R5 BUILT 2013/10/16 19:29
2013-10-24 16:04:59 GOMCMD0900I 13 RULE(S) DEFINED
2013-10-24 16:04:59 GOMCMD0902I 5 MACHINE MONITOR(S) DEFINED
2013-10-24 16:04:59 GOMCMD0903I 661 SCHEDULE(S) DEFINED
2013-10-24 16:04:59 GOMCMD0904I 52 ACTION(S) DEFINED
2013-10-24 16:04:59 GOMCMD0905I 5 ACTION(S) SCHEDULED FOR PROCESSING
2013-10-24 16:04:59 GOMCMD0906I 3 GROUP(S) DEFINED
2013-10-24 16:04:59 GOMCMD0907I 2 DATES(S) DEFINED
2013-10-24 16:04:59 GOMCMD0908I 105 SPOOL MONITOR(S) DEFINED
2013-10-24 16:04:59 GOMCMD0909I 23 EVENT MONITOR(S) DEFINED
2013-10-24 16:04:59 GOMCMD0910I 105 PAGE MONITOR(S) DEFINED
2013-10-24 16:04:59 GOMCMD0911I 1 IDLE MONITOR(S) DEFINED
2013-10-24 16:04:59 GOMCMD0920I 4 SERVICE MACHINE(S) DEFINED
2013-10-24 16:04:59 GOMCMD0921I 3 USER(S) AUTHORIZED
2013-10-24 16:04:59 GOMCMD0922I 11 IUCV CONNECTION(S)
2013-10-24 16:04:59 GOMCMD0923I 4 DATA SPACE SIZE DEFINITION(S)
2013-10-24 16:04:59 GOMCMD0924I 3 COMBINED VIEW(S)
2013-10-24 16:04:59 GOMCMD0925I 6 USER CONSOLE(S) RECORDING
2013-10-24 16:04:59 GOMCMD0926I 10 TCP/IP APPLICATION(S) DEFINED
```

Figure 24. Summary output from STATUS command

Figure 25 shows details about all monitored virtual machines.

```
STATUS DETAIL(MACHINE)
2013-10-24 16:08:54 GOMCMD0942I MACHINE MONITOR MON1 TRIGGERED 0 TIME(S)
2013-10-24 16:08:54 GOMCMD2977I ACTION MACHAC
2013-10-24 16:08:54 GOMCMD2976I DAYS (ALL)
2013-10-24 16:08:54 GOMCMD2975I HOURS (ALL)
2013-10-24 16:08:54 GOMCMD2973I USER RSCS (RSCS - DSC)
2013-10-24 16:08:54 GOMCMD2972I INTERVAL 1 MINUTE(S)
2013-10-24 16:08:54 GOMCMD2971I DELAY 0 MINUTE(S)
2013-10-24 16:08:54 GOMCMD2970I RETRY INTERVAL 1 MINUTE(S) RETRY COUNT 0
2013-10-24 16:08:54 GOMCMD2968I NEXT CHECK 2013/10/24 16:09
2013-10-24 16:08:54 GOMCMD2999I STATUS DETAIL COMPLETE
```

Figure 25. Details about all monitored virtual machines

Figure 26 on page 66 shows details about all spool usage monitors and how many times they were executed.

```

STATUS DETAIL(SPOOL)
2013-10-24 16:11:50 GOMCMD0948I SPOOL MONITOR WARN TRIGGERED 0 TIME(S)
2013-10-24 16:11:50 GOMCMD2977I ACTION ACTNSP
2013-10-24 16:11:50 GOMCMD2976I DAYS (ALL)
2013-10-24 16:11:50 GOMCMD2975I HOURS (ALL)
2013-10-24 16:11:50 GOMCMD2967I USAGE LOW 80 HIGH 89
2013-10-24 16:11:50 GOMCMD2972I INTERVAL 1 MINUTE(S)
2013-10-24 16:11:50 GOMCMD2968I NEXT CHECK 2013/10/24 16:12
2013-10-24 16:11:50 GOMCMD2999I STATUS DETAIL COMPLETE
2013-10-24 16:12:07 GOMCMD0948I SPOOL MONITOR CRIT TRIGGERED 0 TIME(S)
2013-10-24 16:12:07 GOMCMD2977I ACTION ACTNSP
2013-10-24 16:12:07 GOMCMD2976I DAYS (ALL)
2013-10-24 16:12:07 GOMCMD2975I HOURS (ALL)
2013-10-24 16:12:07 GOMCMD2967I USAGE LOW 90 HIGH 100
2013-10-24 16:12:07 GOMCMD2972I INTERVAL 1 MINUTE(S)
2013-10-24 16:12:07 GOMCMD2968I NEXT CHECK 2013/10/24 16:12
2013-10-24 16:12:07 GOMCMD2999I STATUS DETAIL COMPLETE

```

Figure 26. Details about all spool usage monitors

Figure 27 shows the top ten largest spool files and the top ten users with the most spool files.

```

status detail(spoolusr)
2013-10-24 16:14:19 GOMCMD0970I USER SAMARAK SPOOL FILE ID 0124 IS USING 9672 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER BKRCATLG SPOOL FILE ID 0001 IS USING 8737 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER MAINT620 SPOOL FILE ID 0662 IS USING 6763 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER RVBACKUP SPOOL FILE ID 0001 IS USING 5195 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER SAMARAK SPOOL FILE ID 0037 IS USING 5051 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER LNXSLES SPOOL FILE ID 0004 IS USING 4901 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3319 IS USING 4817 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3325 IS USING 4813 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3320 IS USING 4811 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3321 IS USING 4811 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER BKRADMIN HAS 364 SPOOL FILES USING 9188 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER MAINT620 HAS 312 SPOOL FILES USING 9131 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER MAINT HAS 231 SPOOL FILES USING 1102 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER DMARTIN HAS 220 SPOOL FILES USING 8648 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER TCPMAINT HAS 160 SPOOL FILES USING 1801 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER OPERATOR HAS 131 SPOOL FILES USING 2626 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER IGOR HAS 125 SPOOL FILES USING 933 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER SAMARAK HAS 92 SPOOL FILES USING 21314 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER MDMARA HAS 87 SPOOL FILES USING 2429 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD0971I USER RACFVM HAS 85 SPOOL FILES USING 158 SPOOL BLOCKS
2013-10-24 16:14:19 GOMCMD2999I STATUS DETAIL COMPLETE

```

Figure 27. Details about users' spool files

## Usage notes

None.

## Return codes

Table 38 lists the return codes for the STATUS command.

Table 38. STATUS return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

# STOP

▶—STOP—◀

*Figure 28. STOP command syntax*

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the STOP command to stop GOMMAIN. There are no operands and none are allowed.

## Usage notes

1. QUIT, STOP, and SHUTDOWN are equivalent commands. Issue any of these commands to stop GOMMAIN execution.

## Return codes

Table 39 lists the return codes for the STOP command.

*Table 39. STOP return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |



---

## Chapter 6. Syntax for Configuration Commands

This chapter describes the Operations Manager configuration commands and their syntax.

**Related tasks:**

“Step 5. (Optional) Configure an external security mechanism” on page 11

---

### Purpose of configuration commands

Configuration commands allow you to define specific conditions you want Operations Manager to process and what action is taken when that condition occurs. These conditions include:

- rules
- schedules
- machine monitors
- spool monitors
- page space monitors
- event monitors

You can type a configuration command on the Operations Manager console or issue it using the GOMCMD command interface. Both methods execute the command immediately. You can also put configuration commands into a file so that each time you start Operations Manager, the same rules, schedules, monitors, and actions are in effect and you do not have to re-enter them.

Some of these configuration commands are also valid as control commands.

**Related reference:**

Chapter 3, “Creating a startup configuration file,” on page 21

Chapter 5, “Syntax for Control Commands,” on page 39

---

### Authorization for configuration commands

You must use an ESM or the AUTH command to give a specific user the authority to issue configuration commands. To give users access to specific configuration commands, an ESM is required.

**Related tasks:**

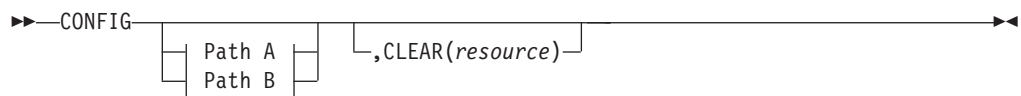
“Step 5. (Optional) Configure an external security mechanism” on page 11

**Related reference:**

“AUTH” on page 39

---

## CONFIG





- IDLE

## Operand value lengths

Table 40 lists the operand value lengths for the CONFIG command.

Table 40. CONFIG operand value lengths

| Operand | Length |
|---------|--------|
| CLEAR   | 1-120  |
| FILE    | 1-20   |
| FM      | 1-2    |
| FN      | 1-8    |
| FT      | 1-8    |

## Usage notes

1. The CONFIG command can be used within a configuration file containing Operations Manager commands, allowing you to embed other configuration files. You can nest the CONFIG command within included files. There is no restriction on the depth of nested files. You can use this feature in the startup configuration file for the GOMMAIN command to organize your configuration data into multiple files.
2. If the FT(ft) operand is entered, the FN operand must be entered.
3. If the FM(fm) operand is entered, the FT operand must be entered.
4. The operands on one path are mutually exclusive with operands on other paths.
5. The value &SYSNAME can be used for either or both of the *fn* or *ft* values in the FN, FT, and FILE operands. The &SYSNAME value is replaced with the system name.

## Return codes

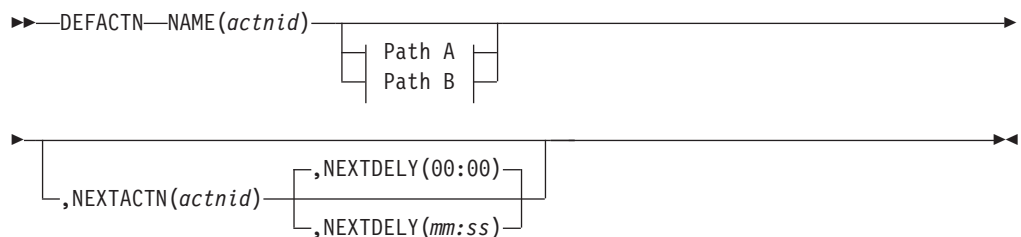
Table 41 lists the return codes for the CONFIG command.

Table 41. CONFIG return codes

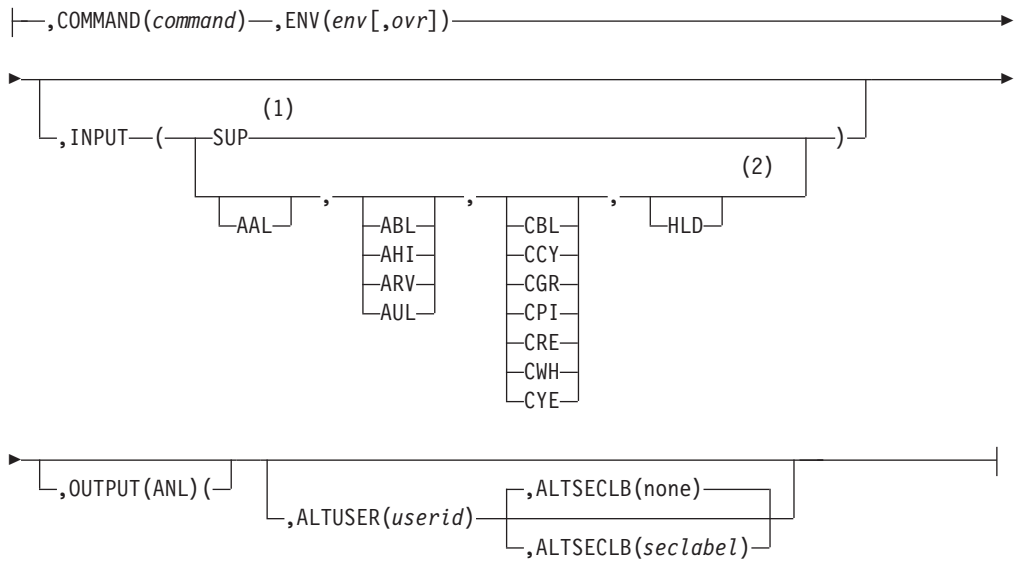
| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

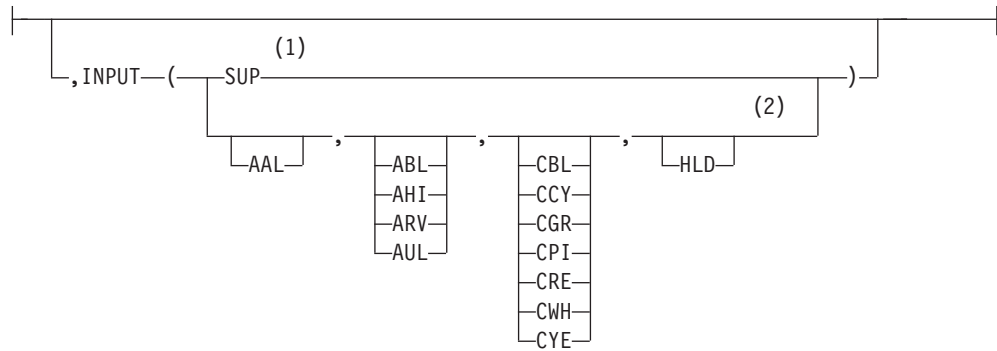
## DEFACTN



### Path A:



### Path B:



### Notes:

- 1 If the SUP option is specified, no other options are allowed.
- 2 Separate multiple options with a comma.

Figure 30. DEFACTN command syntax

### Related reference:

Appendix F, “Application and settings interference,” on page 159

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Use the DEFACTN command to define an action to process as the result of a rule, schedule, or monitor.



## Operands

### **NAME(actnid)**

Specifies a name for the action.

### **COMMAND(command)**

Identifies the command to be processed. This may be any CMS, CP, or Operations Manager command. Operations Manager builds the command allowing variable substitution.

### **ENV(env[,ovr])**

Specifies the environment where the command will be processed and if the processing environment can be overridden by a rule, schedule, or monitor which causes the action's processing.

The valid values for env are:

**GOM** Specifies that this is an Operations Manager command and is processed by the internal Operations Manager command processor.

**LVM** Specifies that the command is processed by the local virtual machine (OPMGRM1). Use caution when specifying LVM. Any command or module that processes in the LVM preempts execution of Operations Manager message processing and can possibly affect the performance and reliability of Operations Manager. It is recommended to use ENV(SVM) for most actions.

**SVM** Specifies that the command is processed by any available action processing server.

**userid** Specifies that the command is processed by only the action processing server indicated.

**wgrpид** Specifies that the command is processed by only action processing servers participating in the work group.

The valid values for ovr are:

**N** Specifies the command processing environment cannot be overridden by a rule, schedule, or monitor. This is the default.

**Y** Specifies the command processing environment can be overridden by a rule, schedule, or monitor.

### **INPUT(inpwka)**

Specifies the well-known action(s) Operations manager takes upon the message input. This is the message that triggered the rule leading to this action.

### **OUTPUT(outwka)**

Specifies the action Operations Manager takes upon the message output of the command specified in the COMMAND operand. This only includes synchronous output. The only available action is ANL, which causes Operations Manager to process the synchronous output of the COMMAND operand through Operations Manager rules.

### **NEXTACTN(actnid)**

Specifies the name of an action to process upon completion of this action.

### **NEXTDELY(mm:ss)**

Specifies an amount of time to delay processing of NEXTACTN. The format is in minutes and seconds.

**ALTUSER(userid)**

Specifies the alternate user ID to be set for the action processing server when processing the action. This operand is *not* valid for GOM or LVM environments. The alternate user ID is set using CP Diagnose '0xD4'. Refer to *z/VM CP Programming Services (SC24-6084)* for more information.

**ALTSECLB(seclabel)**

Specifies the SECLABEL for the service machine when processing the action. This operand is valid *only* if ALTUSER is specified.

**Related reference:**

“DEF SRVM” on page 51

“ALTRCON” on page 124

## Operand value lengths

Table 42 lists the operand value lengths for the DEFACTN command.

Table 42. DEFACTN operand value lengths

| Operand  | Length |
|----------|--------|
| ALTUSER  | 1-8    |
| ALTSECLB | 1-8    |
| COMMAND  | 1-90   |
| ENV      | 1-8    |
| INPUT    | 3-15   |
| NAME     | 1-8    |
| NEXTACTN | 1-8    |
| NEXTDELY | 5      |
| OUTPUT   | 3      |

## Usage notes

1. If the SUP option is specified, no other options are allowed.
2. Separate multiple options with a comma.
3. If OUTPUT is specified, COMMAND must be specified.
4. If COMMAND is specified, ENV must be specified.
5. If ENV is specified, COMMAND must be specified.
6. The following substitution variables are available for all rules, schedules, and monitors in DEFACTN. These substitution variables are always available in DEFACTN.

Table 43. DEFACTN substitution variables

| Variable | Value                                                                                        |
|----------|----------------------------------------------------------------------------------------------|
| &P       | Substitutes the value passed by the PARM operand of a rule, schedule, or monitor.            |
| &R       | Substitutes the ID of the rule, schedule, or monitor that caused the action to be processed. |
| &S       | Substitutes the system name.                                                                 |

The following tables list the substitution variables available when an action is triggered specifically by a rule, schedule, and monitor. These substitution variables are available only when the action is triggered by the specified rule, schedule, or monitor statement.

- DEFEMON allows the following substitution variables:

Table 44. DEFEMON substitution variables

| Variable | Value                                                                                                                                                                                                                                                                                                                                                                      |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| &C       | 0                                                                                                                                                                                                                                                                                                                                                                          |
| &T       | The concatenation of &0 through &n                                                                                                                                                                                                                                                                                                                                         |
| &U       | The constant _GOMEMON                                                                                                                                                                                                                                                                                                                                                      |
| &0-&99   | Specifies information about the event monitor and event. Valid values are:<br><br>&0 - Constant EVENT<br>&1 - The name of the event monitor<br>&2 - The name of the event monitor's action<br>&3 - The user ID which caused the event<br>&4 - The event type<br><br>&5-&n - Any additional information passed from the *VMEVENT system service for a particular event type |

- DEFIMON allows the following substitution variables:

Table 45. DEFIMON substitution variables

| Variable | Value                                                                                                       |
|----------|-------------------------------------------------------------------------------------------------------------|
| &0       | Constant "IDLE"                                                                                             |
| &1       | The name of the idle monitor                                                                                |
| &2       | The name of the idle monitor's action                                                                       |
| &3       | The type of the resource that is monitored: RULE, MACHINE, SCHEDULE, SPOOL, EVENT, PAGE, or IDLE            |
| &4       | The name of the monitored resource                                                                          |
| &5       | The number of times the monitored resource was triggered within the interval specified on the OCCUR operand |

- DEFMMON allows the following substitution variables:

Table 46. DEFMMON substitution variables

| Variable | Value                                    |
|----------|------------------------------------------|
| &0       | Constant "MACHINE"                       |
| &1       | The name of the machine monitor          |
| &2       | The name of the machine monitor's action |
| &3       | The user ID that caused the action       |

- DEFPPMON allows the following substitution variables:

Table 47. DEFPMON substitution variables

| Variable | Value                                                                                                                                                                                                                                                                                                                                                                                              |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| &C       | 0                                                                                                                                                                                                                                                                                                                                                                                                  |
| &T       | The concatenation of &0 &1 &2 &3 &4 &5                                                                                                                                                                                                                                                                                                                                                             |
| &U       | The constant _GOMPMON                                                                                                                                                                                                                                                                                                                                                                              |
| &0-&99   | Specifies information about the page monitor and page space. Valid values are:<br><br>&0 - Constant "PAGE"<br><br>&1 - The name of the page space monitor<br><br>&2 - The name of the page space monitor's action<br><br>&3 - The condition field that is passed to the action (USAGE or INCREASE)<br><br>&4 - The page usage percent<br><br>&5 - The page usage percent increase for this monitor |

- DEFRULE allows the following substitution variables:

Table 48. DEFRULE substitution variables

| Variable | Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| &C       | Indicates the class of the message text. Valid classes are:<br><br>1 - text came from CP MSG or CP MSGNOH<br><br>2 - text came from CP WARNING<br><br>3 - text came from CP console output (includes OBSERVER messages)<br><br>4 - text came from CP SMSG<br><br>5 - text came from VM console output<br><br>6 - text came from CP error message<br><br>7 - text came from CP informational message<br><br>8 - text came from CP single console image facility (includes SECUSER messages)<br><br>22 - text came from Operations Manager TCP/IP Remote Source Input Filter. Refer to GOMRSIF in "Usage notes" on page 54 for more information.<br><br>23 - text came from Operations Manager TCP/IP Remote Syslogd Filter. Refer to GOMRSYL in "Usage notes" on page 54 for more information. |
| &T       | Full message text                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| &U       | User ID of the message text originator                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| &0-&99   | Words of the text                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

- DEFSCHD and RUN allow the following substitution variables:

Table 49. DEFSCHD and RUN substitution variables

| Variable | Value                 |
|----------|-----------------------|
| &C       | 0                     |
| &T       | Null                  |
| &U       | The constant _GOMSCHD |
| &0-&99   | Null                  |

- DEFMON allows the following substitution variables:

Table 50. DEFMON substitution variables

| Variable | Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| &C       | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| &T       | The concatenation of &0 &1 &2 &3 &4 &5 &6 &7                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| &U       | The constant _GOMSMON                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| &0-&99   | Specifies information about the spool monitor, spool usage, and spool file usage. Valid values are:<br><br>&0 - Constant SPOOL<br><br>&1 - The name of the spool monitor<br><br>&2 - The name of the spool monitor's action<br><br>&3 - The condition field passed to the action (USAGE or INCREASE)<br><br>&4 - The spool usage percent<br><br>&5 - The spool usage percent increase for this monitor<br><br>&6 - The spool file usage percent<br><br>&7 - The spool file usage percent increase for this monitor |

7. The well-known input actions are:

- AAL. Activates an audible alarm when the message is displayed.
- ABL. Sets the extended display attribute to blink.
- AHI. Sets the display attribute to high intensity.
- ARV. Sets the extended display attribute to reverse video.
- AUL. Sets the extended display attribute to underline.
- CBL. Sets the extended display color to blue.
- CCY. Sets the extended display color to cyan.
- CGR. Sets the extended display color to green.
- CPI. Sets the extended display color to pink.
- CRE. Sets the extended display color to red.
- CWH. Sets the extended display color to white.
- CYE. Sets the extended display color to yellow.
- HLD. Holds the message on the user's console until it is removed. The HLDATTR operand on DEFOPTN, the ALTRCON command, and the PF05 key in VIEWCON provide additional options for held lines.
- SUP. Suppresses the message from the user's console.

- a. If the SUP option is specified on the INPUT operand, no other options are allowed.
  - b. Only one extended display color is allowed.
  - c. Only one extended attribute is allowed.
8. The only available output action is ANL, which causes Operations Manager to process the synchronous output of the COMMAND operand through Operations Manager rules. If an action is defined with OUTPUT(ANL) the output from the command is captured in the console data space named \_GOMACTN.
  9. Valid well-known output actions from previous releases of Operations Manager are accepted for compatibility, but have no effect on the output.
  10. If ALTUSER is specified, the user ID OPMGRM1, or any user ID running GOMMAIN, must have PRIV B to operate correctly. For more information on privilege classes, refer to Chapter 2, "Configuring Operations Manager," on page 5.
  11. To clear the HLD option, you must press the HOLD function key (PF05 by default) in VIEWCON or issue the ALTRCON command.

**Related reference:**

"DEFOPTN" on page 48

"ALTRCON" on page 124

"VIEWCON" on page 127

## Return codes

Table 51 lists the return codes for the DEFACTN command.

Table 51. DEFACTN return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DEFDATE

▶▶—DEFDATE—NAME(*dateid*)—,HOLIDAY(*yyyy/mm/dd-hh:mm yyyy/mm/dd-hh:mm*)————▶▶

Figure 31. DEFDATE command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFDATE command to define a date as a holiday. These dates may then be included or excluded from schedules and monitors.

## Operands

**NAME(*dateid*)**

Specifies the name of the holiday.

**HOLIDAY (yyyy/mm/dd-hh:mm yyyy/mm/dd-hh:mm)**

Defines the beginning date (year, month, day) and time (hours, minutes) of the holiday, and the ending date/time.

**Related reference:**

“DEFEMON”

“DEFMMON” on page 87

“DEFSCHD” on page 98

“DEFMON” on page 102

**Operand value lengths**

Table 52 lists the operand value lengths for the DEFDATE command.

Table 52. DEFDATE operand value lengths

| Operand | Length |
|---------|--------|
| HOLIDAY | 33     |
| NAME    | 1-8    |

**Usage notes**

None.

**Return codes**

Table 53 lists the return codes for the DEFDATE command.

Table 53. DEFDATE return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

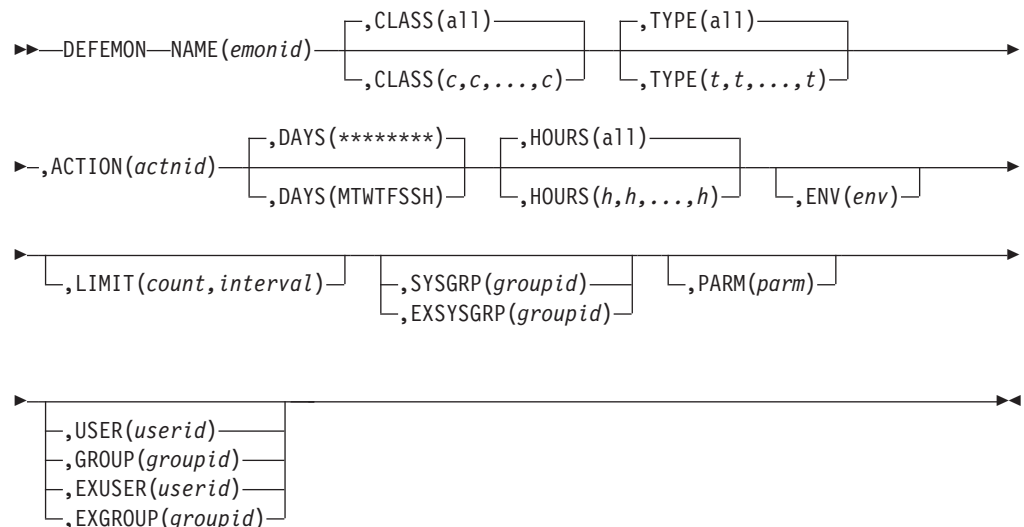
**DEFEMON**

Figure 32. DEFEMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFEMON command to define an event monitor. Events from the CP system service \*VMEVENT are captured and compared against defined event monitors. If conditions for the event monitor are met, the associated action is scheduled for processing.

## Operands

### NAME(*emonid*)

Specifies the name of the event monitor.

### CLASS(*c,c,...,c*)

Specifies the classes of events to monitor. The values for CLASS are listed in the *CP Programming Services* documentation. The default value is all classes.

### TYPE(*t,t,...,t*)

Specifies the types of events to monitor. The values for TYPE are listed in the *CP Programming Services* documentation. The default value is all types.

### ACTION(*actnid*)

Specifies the name of the associated action.

### DAYS(*MTWTFSSH*)

Specifies the days of the week (Monday-Sunday) and if the event is active on holidays. For the event to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

### HOURS(*h,h,...,h*)

Specifies the hours of the day the event is active. Any values from 0 to 23 may appear in the comma-separated list. Up to 24 values may be present. The default is all hours.

### ENV(*env*)

Specifies that the processing environment of the action is overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

### LIMIT(*count, interval*)

Specifies that the processing of the event's action is limited to *count* times within *interval*.

- The value for *count* must be from 1 to 999999.
- The value for *interval* can either be from 1 to 999999 seconds, or in the format of *ddd-hh:mm:ss* which specifies the interval value in days (*ddd*), hours (*hh*), minutes (*mm*), and seconds (*ss*). The values must be between 000-00:00:01 and 999-23:59:59.

For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).



| **SYSGRP(groupid)**

| Specifies that this monitor is only applicable for systems where the local  
| system name is in the group named groupid.

| **EXSYSGRP(groupid)**

| Specifies that this monitor is only applicable for systems where the local  
| system name is *not* in the group named groupid.

| **PARM(parm)**

| Specifies the parameter that is passed to the ACTION when it is scheduled  
| for processing. The DEFACTN command can use the parameter through  
| the &p substitution variable. The parameter is also available to any chained  
| action routine that is defined in the NEXTACTN operand on DEFACTN.

| **USER(userid)**

| Specifies this event is only applicable for events generated for userid.

| **GROUP(groupid)**

| Specifies this event is only applicable for events generated for users in the  
| group named groupid.

| **EXUSER(userid)**

| Specifies this event is only applicable for events not generated for userid.

| **EXGROUP(groupid)**

| Specifies this event is only applicable for events not generated for users in  
| the group named groupid.

| **Related reference:**

| "DEFACTN" on page 71

| "DEFDATE" on page 78

| "DEFGROUP" on page 85

## Operand value lengths

Table 54 lists the operand value lengths for the DEFEMON command.

Table 54. DEFEMON operand value lengths

| Operand  | Length |
|----------|--------|
| ACTION   | 1-8    |
| CLASS    | 1-120  |
| DAYS     | 8      |
| ENV      | 1-8    |
| EXGROUP  | 1-8    |
| EXUSER   | 1-8    |
| EXSYSGRP | 1-8    |
| GROUP    | 1-8    |
| HOURS    | 1-72   |
| LIMIT    | 3-19   |
| NAME     | 1-8    |
| PARM     | 1-120  |
| SYSGRP   | 1-8    |
| TYPE     | 1-120  |
| USER     | 1-8    |

## Usage notes

1. The USER, GROUP, EXUSER, and EXGROUP operands are mutually exclusive.
2. The message field passed to the action contains the following fields:
  - &0 — Constant "EVENT"
  - &1 — The name of the event monitor
  - &2 — The name of the event monitor's action
  - &3 — The userid that caused the event
  - &4 — The event type
  - &5 — The event class
  - &6-&n — Any additional information passed for a particular event class/type
  - &P — Substitutes the value passed by the PARM operand of the event monitor.
  - &R — Substitutes the ID of the event monitor that caused the action to be processed.
  - &S — Substitutes the system name.
3. The SYSGRP operand and the EXSYSGRP operand are mutually exclusive.

## Return codes

Table 55 lists the return codes for the DEFEMON command.

Table 55. DEFEMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

## DEFIMON

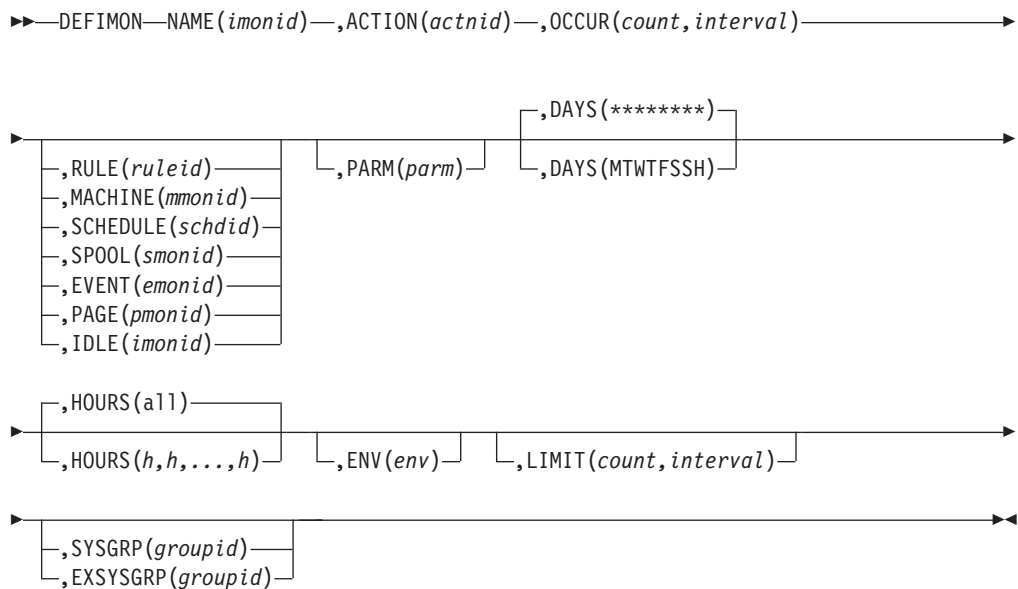


Figure 33. DEFIMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Use the DEFIMON command to define an idle monitor. A check is performed to determine if the RULE, MACHINE, SCHEDULE, SPOOL, EVENT, PAGE, or IDLE resource is triggered a specified number of times within a specified time frame. If a resource is not triggered, the action associated with the idle monitor is scheduled to run.

## Operands

### NAME(*imonid*)

Specifies the name of the idle monitor.

### ACTION(*actnid*)

Specifies the name of the action to process when a resource is not triggered.

### OCCUR(*count*, *interval*)

Specifies the processing of the monitored resource that should happen *count* times, within *interval* minutes. The values for *count* and *interval* must be from 1 to 999999.

### RULE(*ruleid*)

Specifies to monitor the rule *ruleid*.

### MACHINE(*mmonid*)

Specifies to monitor the machine monitor *mmonid*.

### SCHEDULE(*schdid*)

Specifies to monitor the schedule *schdid*.

### SPOOL(*smonid*)

Specifies to monitor the spool monitor *smonid*.

### EVENT(*emonid*)

Specifies to monitor the event monitor *emonid*.

### PAGE(*pmonid*)

Specifies to monitor the page space monitor *pmonid*.

### IDLE(*imonid*)

Specifies to monitor the idle monitor *imonid*.

### PARM(*parm*)

Specifies the parameter that is passed to the action when it is scheduled to run.

### DAYS(MTWFSSH)

Specifies the days of the week (Monday-Sunday) and if the monitor is active on holidays. For the monitor to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

**HOURS(h,h,...,h)**

Specifies the hours of the day the monitor is checked. Any value from 0 to 23 can be used in a comma-separated list. Up to 24 values can be present. The default is all hours.

**ENV(env)**

Specifies that the processing environment of the action is to be overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

**LIMIT(count, interval)**

Specifies that the processing of the monitor's action is limited to *count* times within *interval*.

- The value for *count* must be from 1 to 999999.
- The value for *interval* can either be from 1 to 999999 seconds, or in the format of ddd-hh:mm:ss which specifies the interval value in days (ddd), hours (hh), minutes (mm), and seconds (ss). The value must be between 000-00:00:01 and 999-23:59:59.

For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).

**SYSGRP(groupid)**

Specifies that this monitor is only applicable for systems where the local system name is in the group named *groupid*.

**EXSYSGRP(groupid)**

Specifies that this monitor is only applicable for systems where the local system name is *not* in the group named *groupid*.

## Operand value lengths

Table 56 lists the operand value lengths for the DEFIMON command.

Table 56. DEFIMON operand value lengths

| Operand    | Length   |
|------------|----------|
| ACTION     | 1-8      |
| DAYS       | 8        |
| ENV        | 1-8      |
| EVENT      | 1-8      |
| EXSYSGRP   | 1-8      |
| HOURS      | 1-72     |
| IDLE       | 1-8      |
| LIMIT      | 3-19     |
| MACHINE    | 1-8      |
| NAME       | 1-8      |
| OCCUR(m,n) | 1-6, 1-6 |
| PAGE       | 1-8      |
| PARM       | 1-120    |
| RULE       | 1-8      |
| SCHEDULE   | 1-8      |

Table 56. DEFIMON operand value lengths (continued)

| Operand | Length |
|---------|--------|
| SPOOL   | 1-8    |
| SYSGRP  | 1-8    |

## Usage notes

- Only one of the operands RULE, MACHINE, SCHEDULE, SPOOL, EVENT, PAGE, or IDLE must be specified.
- The message field that is passed to the action contains the following fields:
  - &0 — Constant "IDLE".
  - &1 — The name of the idle monitor.
  - &2 — The name of the idle monitor's action.
  - &3 — The type of the resource that is monitored: RULE, MACHINE, SCHEDULE, SPOOL, EVENT, PAGE, or IDLE
  - &4 — The name of the monitored resource.
  - &5 — The number of times the monitored resource was triggered within the interval specified on the OCCUR operand.
  - &P — Substitutes the value passed by the PARM operand of the idle monitor.
  - &R — Substitutes the ID of the idle monitor that caused the action to be processed.
  - &S — Substitutes the system name
- The SYSGRP operand and the EXSYSGRP operand are mutually exclusive.

## Return codes

Table 57 lists the return codes for the DEFIMON command.

Table 57. DEFIMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

## DEFGROUP

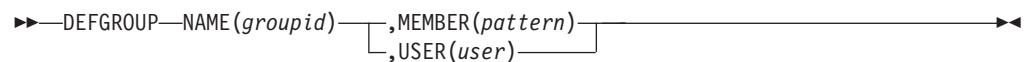


Figure 34. DEFGROUP command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFGROUP command to define a group of members or users to which rules, schedules, and monitors can apply. This command adds members or users, but not both, to a group.

After a group is defined, several commands and operands can refer to the group by its group name. These commands include the DEFEMON, DEFIMON, DEFMMON, DEFPMON, DEFRULE, DEFSCHD, and DEFSMON commands. Depending on how the group is used on these commands, it can refer to groups of users or groups of system names.

## Operands

### NAME(groupid)

Specifies a name for the group

### MEMBER(pattern)

Specifies a member to add to your group. Using the MEMBER operand allows groups to be used as a generalized collection of members and searched for inclusion/exclusion as needed. The pattern may contain wildcard matching information. The wildcard rules are explained in the DEFRULE command.

### USER(user)

Specifies a user ID or wildcard pattern to match against a user ID. The wildcard rules are explained in the DEFRULE command.

**Note:** The USER operand is supported, but is no longer recommended. Use the MEMBER operand instead.

### Related reference:

“DEFSCHD” on page 98

“DEFMMON” on page 87

“DEFSMON” on page 102

“DEFEMON” on page 79

“DEFPMON” on page 90

“DEFIMON” on page 82

“DEFRULE” on page 94

## Operand value lengths

Table 58 lists the operand value lengths for the DEFGROUP command.

Table 58. DEFGROUP operand value lengths

| Operand | Length |
|---------|--------|
| MEMBER  | 1-8    |
| NAME    | 1-8    |
| USER    | 1-8    |

## Usage notes

1. If you want to add multiple member IDs to a group, you must specify one DEFGROUP command for each member ID. For example, to add MEMBER1, MEMBER2, and MEMBER3 into the group GROUP1, specify three DEFGROUP commands.  

```
DEFGROUP NAME(GROUP1),MEMBER(MEMBER1)
DEFGROUP NAME(GROUP1),MEMBER(MEMBER2)
DEFGROUP NAME(GROUP1),MEMBER(MEMBER3)
```
2. Use a wildcard pattern to add multiple, similarly named members to a group . For example, if you have several members whose IDs begin with "ZVMSSI" and you want to add all of those members to the "SSILPARS" group, enter the following command:  

```
DEFGROUP NAME(SSILPARS),MEMBER(ZVMSSI*)
```
3. The MEMBER operand and USER operand are mutually exclusive.

## Return codes

Table 59 lists the return codes for the DEFGROUP command.

Table 59. DEFGROUP return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

## DEFMMON

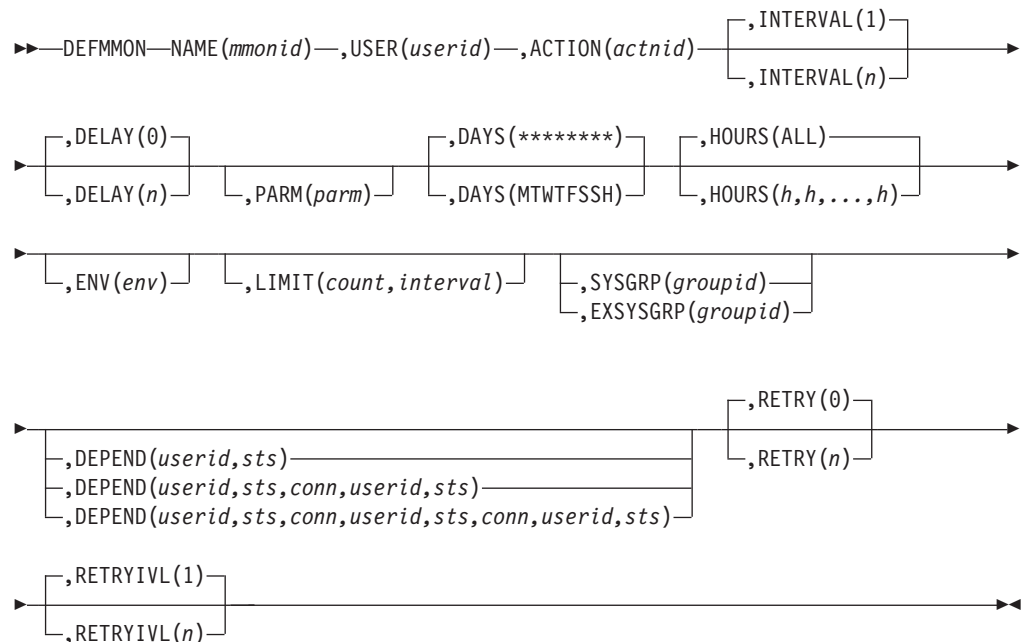


Figure 35. DEFMMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFMMON command to define a virtual machine to monitor. A check is performed to determine if the virtual machine is logged off. If it is, the specified action is scheduled for immediate execution. The action typically issues a command to bring the virtual machine to a logged on status.

## Operands

### NAME(*mmonid*)

Specifies the name of the machine monitor.

### USER(*userid*)

Specifies the user ID of the virtual machine to monitor.

### ACTION(*actnid*)

Specifies the name of the action to process to bring the virtual machine to a "logged on" status. This action must be previously defined using the DEFACTN command.

### INTERVAL(*n*)

Specifies how often, in minutes, to check the virtual machine status. The default is 1.

### DELAY(*n*)

Specifies how long to wait, in minutes, after the monitor is defined to check the virtual machine status. The default is 0 (check immediately).

### PARM(*parms*)

Specifies the parameter that is passed to the action when it is scheduled to run. The DEFACTN command can use the parameter through the &p substitution variable. The parameter is also available to any chained action routine that is defined in the NEXTACTN operand on DEFACTN.

### DAYS(*MTWTFSSH*)

Specifies the days of the week (Monday-Sunday) and if the monitor is checked on holidays. For the monitor to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

### HOURS(*h,h,...,h*)

Specifies the hours of the day the monitor is checked. Any value 0 to 23 may appear in the comma-separated list. Up to 24 values may be present. The default is all hours.

### ENV(*env*)

Specifies that the processing environment of the action is overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

### LIMIT(*count, interval*)

Specifies that the processing of the monitor's action is limited to *count* times within *interval*.

- The value for *count* must be from 1 to 999999.
- The value for *interval* can either be from 1 to 999999 seconds, or in the format of *ddd-hh:mm:ss* which specifies the interval value in days (*ddd*), hours (*hh*), minutes (*mm*), and seconds (*ss*). The value must be between 000-00:00:01 and 999-23:59:59.



For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).

| **SYSGRP(groupid)**

| Specifies that this monitor is only applicable for systems where the local  
| system name is in the group named groupid.

| **EXSYSGRP(groupid)**

| Specifies that this monitor is only applicable for systems where the local  
| system name is *not* in the group named groupid.

| **DEPEND(userid,sts)**

| **DEPEND(userid,sts,conn,userid,sts)**

| **DEPEND(userid,sts,conn,userid,sts,conn,userid,sts)**

| Specifies up to three virtual machines and their status before the monitor is  
| checked.

- **userid** — Specifies the user ID of the virtual machine to check.
- **sts** — Specifies the desired status of the virtual machine:
  - **A** — active. The virtual machine is logged on (or running disconnected).
  - **I** — inactive. The virtual machine is logged off.
- **conn** — Specifies the logical connector:
  - **|** (vertical bar) — or/exclude
  - **&** (ampersand) — and/include

The logical connectors are evaluated left to right and have no hierarchy. The default is no dependencies.

| **RETRY(n)**

| Specifies the number of times to retry the action if the monitored virtual  
| machine is not brought to a logged on status. Each INTERVAL that does not  
| bring the virtual machine to a logged on status is considered a retry. When  
| the number of retries is exceeded, the monitor checking interval is  
| advanced by the number of minutes in the RETRYIVL operand. This is  
| useful if virtual machine does not attain or remain in the logged on status.  
| It can reduce the resource consumption used by a continuously failing  
| virtual machine. The default is 0.

| **RETRYIVL(n)**

| Specifies the number of minutes to elapse before the monitor is checked if  
| the retry is exceeded. If specified, it must be greater than or equal to  
| INTERVAL. The default is the value of INTERVAL.

| **Related reference:**

| “DEFACTN” on page 71

| “DEFDATE” on page 78

## Operand value lengths

Table 60 lists the operand value lengths for the DEFMMON command.

Table 60. DEFMMON operand value lengths

| Operand | Length |
|---------|--------|
| ACTION  | 1-8    |
| DAYS    | 8      |
| DELAY   | 1-6    |

Table 60. DEFMMON operand value lengths (continued)

| Operand  | Length |
|----------|--------|
| DEPEND   | 3-36   |
| ENV      | 1-8    |
| EXSYSGRP | 1-8    |
| HOURS    | 1-72   |
| INTERVAL | 1-6    |
| LIMIT    | 3-19   |
| NAME     | 1-8    |
| PARM     | 1-120  |
| RETRY    | 1-6    |
| RETRYIVL | 1-6    |
| SYSGRP   | 1-8    |
| USER     | 1-8    |

## Usage notes

- The message field passed to the action contains the following fields:
  - &0 — Constant "MACHINE"
  - &1 — The name of the machine monitor
  - &2 — The name of the machine monitor's action
  - &3 — The user ID that caused the action
  - &P — Substitutes the value passed by the PARM operand of the virtual machine monitor.
  - &R — Substitutes the ID of the virtual machine monitor that caused the action to be processed.
  - &S — Substitutes the system name
- The SYSGRP operand and the EXSYSGRP operand are mutually exclusive.

## Return codes

Table 61 lists the return codes for the DEFMMON command.

Table 61. DEFMMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DEFPMON

$\rightarrow$  DEFPMON—NAME(*pmonid*)—,ACTION(*actnid*)—
 
 , INTERVAL(1)  
 , INTERVAL(*n*)
 

 , PARM(*parm*)
 
 $\rightarrow$

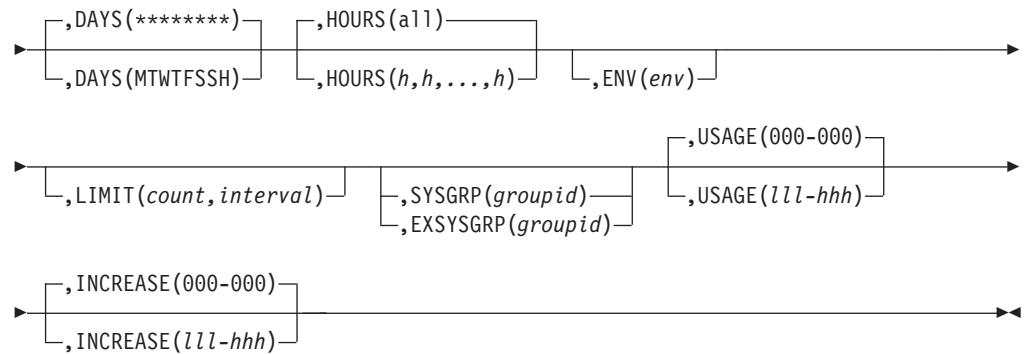


Figure 36. DEFPMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFPMON command to define a page space monitor. A check is performed to determine if the page usage percent or page usage percent increase is within the low and high bounds that are set by the USAGE or INCREASE operand. If so, the action is scheduled for processing. See the usage notes for more information on the substitution variables that are available to the action.

## Operands

### NAME(pmonid)

Specifies the name of the page space monitor.

### ACTION(actnid)

Specifies the name of the action to process when the page usage falls within the bounds for the USAGE or INCREASE operand.

### INTERVAL(n)

Specifies how often to check the page usage for this monitor in minutes. The default is 1.

### PARM(parm)

Specifies the parameter that is passed to the action when it is scheduled to run. The DEFACTN command can use the parameter through the &p substitution variable. The parameter is also available to any chained action routine that is defined in the NEXTACTN operand on DEFACTN.

### DAYS(MTWFSSH)

Specifies the days of the week (Monday-Sunday) and if the monitor is active on holidays. For the monitor to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

### HOURS(h,h,...,h)

Specifies the hours of the day the monitor is active. Any of the values 0 to 23 may appear in the comma-separated list. Up to 24 values may be present. The default is all hours.

**ENV(env)**

Specifies that the processing environment of the action is overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

**LIMIT(count, interval)**

Specifies that the processing of the monitor's action is limited to *count* times within *interval*.

- The value for *count* must be from 1 to 999999.
- The value for *interval* can either be from 1 to 999999 seconds, or in the format of ddd-hh:mm:ss which specifies the interval value in days (ddd), hours (hh), minutes (mm), and seconds (ss). The value must be between 000-00:00:01 and 999-23:59:59.

For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).

**SYSGRP(groupid)**

Specifies that this monitor is only applicable for systems where the local system name is in the group named *groupid*.

**EXSYSGRP(groupid)**

Specifies that this monitor is only applicable for systems where the local system name is *not* in the group named *groupid*.

**USAGE(111-hhh)**

Specifies the low and high usage percent (000-100) bounds to process the action. If USAGE is not specified, no usage percent check is performed. Both low and high bounds must be three digits. The low and high bounds are percentages of the total available page space.

**INCREASE(111-hhh)**

Specifies the low and high usage percent increase (000-100) bounds to process the action. If INCREASE is not specified, no usage percent increase check is performed. The usage percent increase is computed by subtracting the previous page usage from the current page usage. The low and high bounds must be three digits. The low and high bounds are percentages of the total available page space.

**Related reference:**

"DEFACTN" on page 71

"DEFDATE" on page 78

## Operand value lengths

Table 62 lists the operand value lengths for the DEFPMON command.

Table 62. DEFPMON operand value lengths

| Operand  | Length |
|----------|--------|
| ACTION   | 1-8    |
| DAYS     | 8      |
| ENV      | 1-8    |
| EXSYSGRP | 1-8    |
| HOURS    | 1-72   |
| INCREASE | 7      |

Table 62. DEFPMON operand value lengths (continued)

| Operand  | Length |
|----------|--------|
| INTERVAL | 1-6    |
| LIMIT    | 3-19   |
| NAME     | 1-8    |
| PARM     | 1-120  |
| SYSGRP   | 1-8    |
| USAGE    | 7      |

## Usage notes

1. The VM user ID that is running GOMMAIN (OPMGRM1 by default), must have PRIV D to operate correctly. For more information on privilege classes, refer to Chapter 2, "Configuring Operations Manager," on page 5.
2. Either USAGE or INCREASE must be specified.
3. The low and high bound checks are inclusive.
4. Monitor checking is performed in the following order:
  - a. USAGE is checked first. If usage falls within the low and high bounds, the ACTION is scheduled for USAGE condition and no further checks are made.
  - b. INCREASE is checked second. If usage increase occurs within the low and high bounds, the ACTION is scheduled for INCREASE condition.
5. The message field that is passed to the action contains the following fields:
  - &0 — Constant "PAGE"
  - &1 — The name of the page space monitor
  - &2 — The name of the page space monitor's action
  - &3 — The condition field that is passed to the action (USAGE or INCREASE)
  - &4 — The page usage percent
  - &5 — The page usage percent increase for this monitor
  - &P — Substitutes the value passed by the PARM operand of the page space monitor.
  - &R — Substitutes the ID of the page space monitor that caused the action to be processed.
  - &S — Substitutes the system name
6. The value for space utilization percentage is obtained by rounding downward. This value may differ from the space utilization percentage computed from other sources.
7. The SYSGRP operand and the EXSYSGRP operand are mutually exclusive.

## Return codes

Table 63 lists the return codes for the DEFPMON command.

Table 63. DEFPMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

# DEFRULE

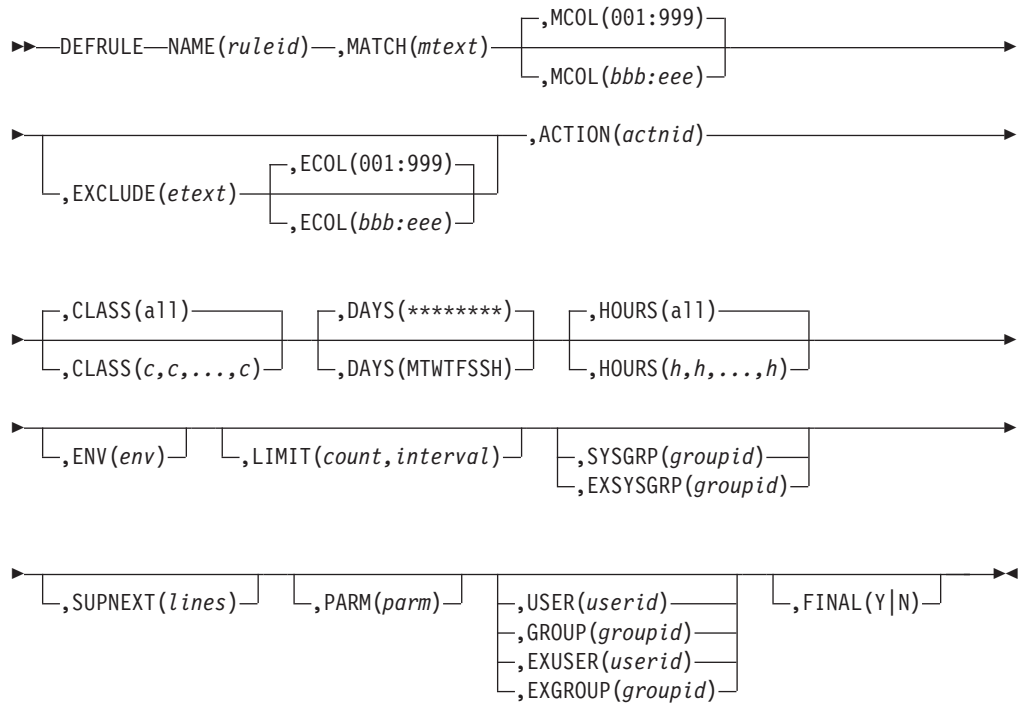


Figure 37. DEFROLE command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFROLE command to setup a rule that defines a pattern that is used for comparing messages against and determining if there is a match. The DEFROLE command also defines an action that is associated with the rule. If conditions for the RULE are met, the associated action is scheduled for processing.

## Operands

### NAME(ruleid)

Specifies the name of the rule.

### MATCH(mtext)

Specifies the pattern used to compare against a message to determine if there is a match. Wildcard pattern matching is allowed. If a match is found, the associated action is scheduled for processing.

### MCOL(bbb:eee)

Specifies the beginning and ending columns to search for the match pattern (inclusive). The valid format is bbb:eee, where bbb is the beginning column and eee is the ending column. The default is 001:999.

**EXCLUDE(etext)**

Specifies the pattern used to compare against a message to determine if there is an exclusion. Wildcard matching is allowed. If an exclusion is found, the associated action is *not* scheduled for processing.

**ECOL(bbb:eee)**

Specifies the beginning and ending columns to search for the exclude pattern (inclusive). The valid format is bbb:eee, where bbb is the beginning column and eee is the ending column. The default is 001:999.

**Note:** The ECOL operand may be specified only if the EXCLUDE operand is specified.

**ACTION(actnid)**

Specifies the name of the associated action.

**CLASS(c,c,...,c)**

Specifies that this rule is only applicable for the listed message classes. Any value from 1 to 255 may appear in the comma-separated list. Up to 32 values may be present. The default is all classes. For a list of valid message classes, see Table 48 on page 76.

**DAYS(MTWFSSH)**

Specifies the days of the week (Monday-Sunday) and if the rule is active on holidays. For the rule to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

**HOURS(h,h,...,h)**

Specifies the hours of the day the rule is active. Any of the values from 0 to 23 may appear in the comma-separated list. Up to 24 values may be present. The default is all hours.

**ENV(env)**

Specifies the processing environment of the action is overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

**LIMIT(count, interval)**

Specifies that the processing of the rule's action is limited to *count* times within *interval*.

- The value for *count* must be from 1 to 999999.
- The value for *interval* can either be from 1 to 999999 seconds, or in the format of ddd-hh:mm:ss which specifies the interval value in days (ddd), hours (hh), minutes (mm), and seconds (ss). The value must be between 000-00:00:01 and 999-23:59:59.

For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).

**SYSGRP(groupid)**

Specifies that this rule is only applicable for systems where the local system name is in the group named groupid.

**EXSYSGRP(groupid)**

Specifies that this rule is only applicable for systems where the local system name is *not* in the group named groupid.

**SUPNEXT(lines)**

Suppresses the number of output lines specified from each user's console

affected by this rule. The suppressed lines begin with the line following the line that triggered this rule. The value for lines can be from 0 to 999999. If lines are already suppressed for the affected user's console, the number of remaining suppressed lines will be changed to *lines*. If SUPNEXT is not specified on a triggered rule, the default is not to change the line suppression count for the affected user's console.

**PARM(parm)**

Specifies the parameter that is passed to the action when it is scheduled for processing. The DEFACTN command can use the parameter through the &p substitution variable. The parameter is also available to any chained action routine that is defined in the NEXTACTN operand on DEFACTN.

**USER(userid)**

Specifies this rule is only applicable for messages from the specified userid.

**GROUP(groupid)**

Specifies this rule is only applicable for messages from users in the group named groupid.

**EXUSER(userid)**

Specifies this rule is only applicable for messages not from userid.

**EXGROUP(groupid)**

Specifies this rule is only applicable for messages not from users in the group named groupid.

**FINAL(Y|N)**

Specifies if this is the final rule to process for the current message if the action was scheduled. The default value is FINAL(N), which specifies rule processing continues for this message. FINAL(Y) specifies no further rule processing for this message.

**Related reference:**

“DEFACTN” on page 71

“DEFDATE” on page 78

“DEFGROUP” on page 85

## Operand value lengths

Table 64 lists the operand value lengths for the DEFRULE command.

Table 64. DEFULE operand value lengths

| Operand  | Length |
|----------|--------|
| ACTION   | 1-8    |
| DAYS     | 8      |
| ECOL     | 7      |
| ENV      | 1-8    |
| EXCLUDE  | 1-60   |
| EXGROUP  | 1-8    |
| EXSYSGRP | 1-8    |
| EXUSER   | 1-8    |
| FINAL    | 1      |
| GROUP    | 1-8    |



Table 64. DEFULE operand value lengths (continued)

| Operand | Length |
|---------|--------|
| HOURS   | 1-72   |
| LIMIT   | 3-19   |
| MATCH   | 1-60   |
| MCOL    | 7      |
| NAME    | 1-8    |
| PARM    | 1-120  |
| SUPNEXT | 1-6    |
| SYSGRP  | 1-8    |
| USER    | 1-8    |

## Usage notes

- Wildcard matching is performed using the following:
  - An asterisk (\*) means zero (0) or more characters of any kind.
  - A percent sign (%) means one (1) required character of any kind.
  - A pound sign (#) means one (1) required numeric character (0-9).
  - An ampersand (&) means one (1) required alphabetic character (A-Z, a-z).
  - An at sign (@) means one (1) required hexadecimal character (A-F, a-f, 0-9).
  - A quote mark (") is the escape character for matching any of the above characters.

Wildcard patterns can be used in any combination.

- USER, GROUP, EXUSER, and EXGROUP are mutually exclusive.
- The text specified on the MATCH and EXCLUDE operands must exactly match the full text found in the columns specified in MCOL and ECOL respectively.
- Use wildcard patterns to look for a substring anywhere in the message.

The following example triggers the action TESTACTN if the text "Testing error" appears anywhere in a message:

```
DEFRULE NAME(TEST),MATCH(*Testing error*),ACTION(TESTACTN)
```

If you do not specify an asterisk at the end of the MATCH operand value, the matching text must end in the column specified in the MCOL operand. Similarly, if you do not specify an asterisk at the beginning of the MATCH operand value, the matching text must begin in the column specified in the MCOL operand.

The example below triggers the action TESTACTN if the text "Testing error" appears in columns 28 through 40 of a message. The matching text must end in column 40 because there is no asterisk at the end of the matching text. Since the matching text is 13 characters, the beginning column must be 28 even though the MCOL beginning column is defined as 001.

```
DEFRULE NAME(TEST),MATCH(*Testing error),MCOL(001:040),ACTION(TESTACTN)
```

- Notes about the SUPNEXT operand:
  - SUPNEXT on a DEFRULE command and INPUT(SUP) on a DEFACTN command are independent. SUPNEXT suppresses the number of lines specified, starting with the line following the line that triggered the rule. INPUT(SUP) suppresses the current line triggered by the rule.
  - All suppressed lines continue through rule processing.

c. Separate values are maintained for SUPNEXT for each console. For example, a rule containing SUPNEXT(10) can affect 3 consoles. For one of those consoles, the third suppressed line can trigger a different rule that uses SUPNEXT(0) to cancel line suppression for that console. The remaining two consoles still suppress the remaining 7 lines.

6. The SYSGRP operand and the EXSYSGRP operand are mutually exclusive.
7. The message field that is passed to the action contains the following fields:
  - &C — Contains the class of the message. For a complete list of classes, refer to Table 48 on page 76.
  - &P — Substitutes the value passed by the PARM operand of the rule.
  - &R — Substitutes the ID of the rule that caused the action to be processed.
  - &S — Substitutes the system name.
  - &T — Full message text
  - &U — User ID of the message text originator
  - &0-&99 — Words of the text

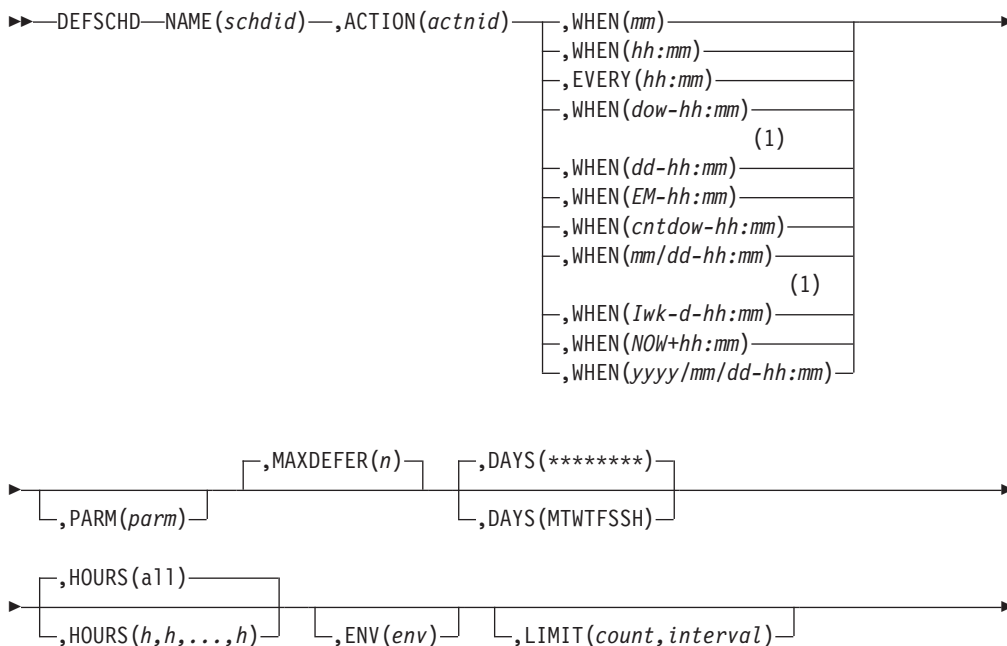
## Return Codes

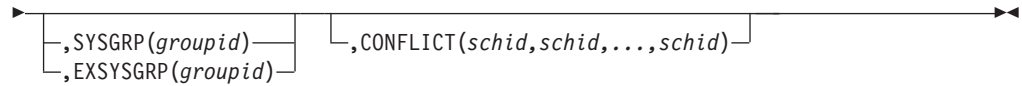
Table 65 lists the return codes for the DEFRULE command.

Table 65. DEFRULE return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

## DEFSCHD





**Notes:**

- 1 If the specified value does not occur within a month or year, the execution time is postponed until the value is reached. For example: WHEN(31-00:00) causes the execution to occur only in months with 31 days.  
WHEN(02/29-00:00) causes the execution to occur only in leap years.

Figure 38. DEFSCHD command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Use the DEFSCHD command to define a schedule and associate an action with the schedule. The schedule indicates whether the action is processed once, hourly or on a specific hour, daily or on a specific day, on a holiday, monthly, or yearly. It also indicates which days, hours, and holidays are used for processing.

## Operands

**NAME(schdid)**

Specifies the name of the schedule.

**ACTION(actnid)**

Specifies the name of the action to associate with the schedule.

**WHEN(mm)**

Specifies the processing time is hourly. The action is scheduled at mm minutes of the hour.

**WHEN(hh:mm)**

Specifies the processing time is daily. The action is scheduled at hh:mm time of the day. For example, 01:30 refers to 1:30 A.M., and 13:30 refers to 1:30 P.M.

**EVERY(hh:mm)**

Specifies the processing time is based on an interval. The interval is every hh:mm of the day starting with Operations Manager startup.

**WHEN(dow-hh:mm)**

Specifies the processing time is weekly. The action is scheduled on a specific *dow* or day of the week (MON, TUE, WED, THU, FRI, SAT, or SUN) at hh:mm time of the day.

**WHEN(dd-hh:mm)**

Specifies the processing time is monthly. The action is scheduled on dd day of the month at hh:mm time of the day.

**WHEN(EM-hh:mm)**

Specifies the processing time is monthly. The action is scheduled on the last day of the month at hh:mm time of the day.

**WHEN(cntdow-hh:mm)**

Specifies the processing time is monthly. The action is scheduled on the specified day at hh:mm time of the day. The cnt can be 1ST, 2ND, 3RD, 4TH, 5TH, or LST day of the month. The dow can be MON, TUES, WED, THU, FRI, SAT, SUN, WKD (week day), or WKE (weekend day).

**WHEN(mm/dd-hh:mm)**

Specifies the processing time is yearly. The action is scheduled on mm/dd of the year at hh:mm time of the day.

**WHEN(Iwk-d-hh:mm)**

Specifies the processing time is based upon the ISO week number. The wk value can be a week number (01-53), the first week (FW), last week (LW), even week (EV), or odd week (OD). The d value is the ISO "day of the week" number (1-7). The days are numbered from 1 (Monday) through 7 (Sunday). The action is scheduled at hh:mm time of the day.

**WHEN(NOW+hh:mm)**

Specifies the processing time to be once, after a delay of hh:mm from now. The hh:mm may range from +00:00 to +23:59. If omitted, the default is +00:00.

**WHEN(yyyy/mm/dd-hh:mm)**

Specifies the processing time to be once on the specified date and time. The action is not scheduled if the date entered has already occurred.

**PARM(parm)**

Specifies the parameter that is passed to the action when it is scheduled for processing. The DEFACTN command can use the parameter through the & substitution variable. The parameter is also available to any chained action routine that is defined in the NEXTACTN operand on DEFACTN.

**MAXDEFER(n)**

Specifies the number of minutes the schedule may be delayed from the beginning of the processing due to CONFLICT. If a schedule is not able to begin before the MAXDEFER value is reached, the schedule is advanced to the next processing time. The default is 0.

**DAYS(MTWFSSH)**

Specifies the days of the week (Monday-Sunday) and if the schedule is active on holidays. For the schedule to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

**HOURS(h,h,...,h)**

Specifies the hours of the day the schedule is active. Any value from 0 to 23 may appear in the comma separated list. Up to 24 values may be present. The default is all hours.

**ENV(env)**

Specifies that the processing environment of the action is overridden with the env supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for env.

**LIMIT(count, interval)**

Specifies that the processing of the schedule's action is limited to count times within interval.

- The value for count must be from 1 to 999999.

- The value for *interval* can either be from 1 to 999999 seconds, or in the format of ddd-hh:mm:ss which specifies the interval value in days (ddd hours (hh), minutes (mm), and seconds (ss). The value must be between 000-00:00:01 and 999-23:59:59.

For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).

**SYSGRP(groupid)**

Specifies that this schedule is only applicable for systems where the local system name is in the group named groupid.

**EXSYSGRP(groupid)**

Specifies that this schedule is only applicable for systems where the local system name is *not* in the group named groupid.

**CONFLICT(schid,schid,...,schid)**

Specifies up to 16 conflicting schedules. If any of the specified schedules are processing, this schedule is deferred from processing. The schedule is then checked every minute until MAXDEFER is reached.

**Related reference:**

“DEFACTN” on page 71

“DEFDATE” on page 78

## Operand value lengths

Table 66 lists the operand value lengths for the DEFSCHD command.

Table 66. DEFSCHD operand value lengths

| Operand  | Length |
|----------|--------|
| ACTION   | 1-8    |
| CONFLICT | 1-120  |
| DAYS     | 8      |
| ENV      | 1-8    |
| EVERY    | 5      |
| EXSYSGRP | 1-8    |
| HOURS    | 1-72   |
| LIMIT    | 3-19   |
| MAXDEFER | 1-6    |
| NAME     | 1-8    |
| PARM     | 1-120  |
| SYSGRP   | 1-8    |
| WHEN     | 2-16   |

## Usage notes

1. Any changes made to the hardware TOD clock do not automatically adjust existing schedules. Stop Operations Manager before making changes to the TOD clock.
2. CONFLICT describes a one-way conflict. If schedules A and B should not execute at the same time, schedule A must name schedule B as a conflict *and* schedule B must name schedule A as a conflict.

3. The processing time for any schedule that specifies *mm* for minutes is not necessarily at exactly *mm:00*. The processing begins within 30 seconds of the start of the minute specified.
4. The `SYSGRP` operand and the `EXSYSGRP` operand are mutually exclusive.
5. The message field passed to the action contains the following fields:
  - `&C` — 0
  - `&P` — Substitutes the value passed by the `PARM` operand of the schedule.
  - `&R` — Substitutes the ID of the schedule that caused the action to be processed.
  - `&S` — Substitutes the system name.
  - `&T` — Null
  - `&U` — The constant `_GOMSCHD`
  - `&0-&99` — Null

## Return codes

Table 67 lists the return codes for the `DEFSCHD` command.

Table 67. *DEFSCHD* return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

## DEFSMON

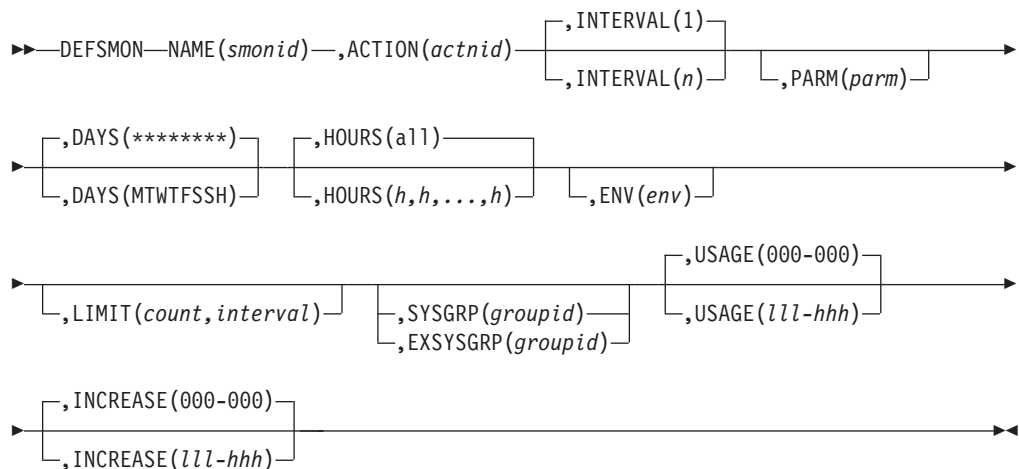


Figure 39. *DEFSMON* command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the `DEFSMON` command to define a spool monitor. A check is performed to determine if the spool usage percent or spool usage percent increase is within the

low and high bounds set by USAGE or INCREASE. If so, the specified action is scheduled for processing. See “Usage notes” on page 104 for details of the message passed to the action.

## Operands

### NAME(*smonid*)

Specifies the name of the spool monitor.

### ACTION(*actnid*)

Specifies the name of the action to process when spool usage falls within the bounds for USAGE or INCREASE.

### INTERVAL(*n*)

Specifies how often, in minutes, to check the spool usage for this monitor. The default is 1.

### PARM(*parm*)

Specifies the parameter that is passed to the action when it is scheduled for processing. The DEFACTN command can use the parameter through the &p substitution variable. The parameter is also available to any chained action routine that is defined in the NEXTACTN operand on DEFACTN.

### DAYS(MTWFSSH)

Specifies the days of the week (Monday-Sunday) and if the monitor is checked on holidays. For the monitor to be active on a day of the week and holiday, the corresponding location must contain the asterisk (\*) symbol. Issue the DEFDATE command to define holidays. The default is all days and holidays (\*\*\*\*\*).

### HOURS(*h,h,...,h*)

Specifies the hours of the day the monitor is checked. Any of the values from 0 to 23 can be used in a comma-separated list. Up to 24 values can be present. The default is all hours.

### ENV(*env*)

Specifies that the processing environment of the action is overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

### LIMIT(*count, interval*)

Specifies that the processing of the monitor's action is limited to *count* times within *interval*.

- The value for *count* must be from 1 to 999999.
- The value for *interval* can either be from 1 to 999999 seconds, or in the format of ddd-hh:mm:ss which specifies the interval value in days (ddd), hours (hh), minutes (mm), and seconds (ss). The value must be between 000-00:00:01 and 999-23:59:59.

For example, a LIMIT of 4 times in 10 seconds could be specified as either LIMIT(4,10) or LIMIT(4,000-00:00:10).

### SYSGRP(*groupid*)

Specifies that this monitor is only applicable for systems where the local system name is in the group named *groupid*.

### EXSYSGRP(*groupid*)

Specifies that this monitor is only applicable for systems where the local system name is *not* in the group named *groupid*.

**USAGE(111-hhh)**

Specifies the low and high usage percent (000-100) bounds to process the ACTION. If USAGE is not specified, no usage percent check is performed. Both low and high bounds must be three digits. The low and high bounds are percentages of the total available spool space.

**INCREASE(111-hhh)**

Specifies the low and high usage percent increase (000-100) bounds to process the ACTION. If INCREASE is not specified, no usage percent increase check is performed. The usage percent increase is computed by subtracting the spool usage percent at the previous interval from the current spool usage percent. The low and high bounds must be three digits. The low and high bounds are percentages of the total available spooling space.

**Related reference:**

“DEFACTN” on page 71

“DEFDATE” on page 78

## Operand value lengths

Table 68 lists the operand value lengths for the DEFMON command.

Table 68. DEFMON operand value lengths

| Operand  | Length |
|----------|--------|
| ACTION   | 1-8    |
| DAYS     | 8      |
| ENV      | 1-8    |
| EXSYSGRP | 1-8    |
| HOURS    | 1-72   |
| INCREASE | 7      |
| INTERVAL | 1-6    |
| LIMIT    | 3-19   |
| NAME     | 1-8    |
| PARM     | 1-120  |
| SYSGRP   | 1-8    |
| USAGE    | 7      |

## Usage notes

1. The VM user ID running GOMMAIN must have PRIV D to operate correctly. For more information on privilege classes, refer to Chapter 2, “Configuring Operations Manager,” on page 5.
2. Either USAGE or INCREASE must be specified.
3. The low and high bound checks are inclusive.
4. Monitor checking is performed in the following order:
  - a. USAGE is checked first. If usage falls within the low and high bounds, the ACTION is scheduled for USAGE condition and no further checks are made.
  - b. INCREASE is checked second. If usage increase occurs within the low and high bounds, the ACTION is scheduled for INCREASE condition.
5. The message field passed to the action contains the following fields:
  - &0 — Constant "SPOOL".



- &1 — The name of the spool monitor.
  - &2 — The name of the spool monitor’s action.
  - &3 — The condition field passed to the action. The valid conditions are USAGE or INCREASE.
  - &4 — The spool usage percentage.
  - &5 — The spool usage percent increase for this monitor.
  - &6 — The spool file usage percentage.
  - &7 — The spool file usage percentage increase for this monitor.
  - &P — Substitutes the value passed by the PARM operand of the spool monitor.
  - &R — Substitutes the ID of the spool monitor that caused the action to be processed.
  - &S — Substitutes the system name
6. The value for space utilization percentage is obtained by rounding downward. This value may differ from the space utilization percentage computed from other sources.
  7. The SYSGRP operand and the EXSYSGRP operand are mutually exclusive.

## Return codes

Table 69 lists the return codes for the DEFMON command.

Table 69. DEFMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DEFACTN

▶▶—DEFACTN—NAME(*actnid*)—————▶▶

Figure 40. DEFACTN command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DEFACTN command to delete an action.

**Related reference:**

“DEFACTN” on page 71

## Operands

**NAME(*actnid*)**

Specifies the name of the action to delete.

## Operand value lengths

Table 70 lists the operand value lengths for the DELACTN command.

Table 70. DELACTN operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 71 lists the return codes for the DELACTN command.

Table 71. DELACTN return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELDATE

►►—DELDATE—NAME(*dateid*)—————►►

Figure 41. DELDATE command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELDATE command to delete a holiday.

**Related reference:**

“DEFDATE” on page 78

## Operands

**NAME(*dateid*)**

Specifies the name of the holiday to delete.

## Operand value lengths

Table 72 lists the operand value lengths for the DELDATE command.

Table 72. DELDATE operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 73 lists the return codes for the DELDATE command.

Table 73. DELDATE return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELEMON

►►—DELEMON—NAME (*emonid*)—◄◄

Figure 42. DELEMON command syntax

**Related reference:**

“DEFEMON” on page 79

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELEMON command to delete an event monitor.

## Operands

**NAME (*emonid*)**

Specifies the name of the event monitor to delete.

## Operand value lengths

Table 74 lists the operand value lengths for the DELEMON command.

Table 74. DELEMON operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 75 lists the return codes for the DELEMON command.

Table 75. DELEMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELGROUP

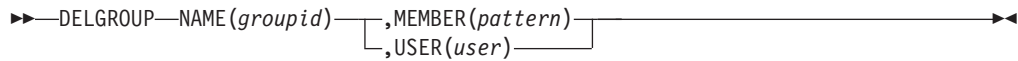


Figure 43. DELGROUP command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELGROUP command to delete members or users from a group.

Before deleting a member or user from a group, keep in mind that several commands and operands may refer to the group. These commands include the DEFEMON, DEFIMON, DEFMMON, DEFPMON, DEFRULE, DEFSCHD, and DEFMON commands.

### Related reference:

“DEFGROUP” on page 85

## Operands

### NAME(groupid)

Specifies the group name.

### MEMBER(pattern)

Specifies a member to delete from your group. Using the MEMBER operand allows groups to be used as a generalized collection of members and searched for inclusion/exclusion as needed. The pattern may contain wildcard matching information. The wildcard rules are explained in the DEFRULE command.

### USER(user)

Specifies a user ID or a wildcard pattern to delete from the group.

**Note:** The USER operand is supported, but is no longer recommended. Use the MEMBER operand instead.

## Operand value lengths

Table 76 on page 109 lists the operand value lengths for the DELGROUP command.

Table 76. DELGROUP operand value lengths

| Operand | Length |
|---------|--------|
| MEMBER  | 1-8    |
| NAME    | 1-8    |
| USER    | 1-8    |

## Usage notes

1. The value of the NAME and of the MEMBER (or USER) operands must completely match a previously defined value.
2. Only the unique combination of NAME and MEMBER (or USER) is removed. Other NAME and MEMBER (or USER) combinations are not removed.
3. The MEMBER operand and USER operand are mutually exclusive.

## Return codes

Table 77 lists the return codes for the DELGROUP command.

Table 77. DELGROUP return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELIMON

▶▶—DELIMON—NAME(*imonid*)————▶▶

Figure 44. DELIMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELIMON command to delete an idle monitor.

## Operands

**NAME(*imonid*)**

Specifies the name of the idle monitor.

## Operand value lengths

Table 78 lists the operand value lengths for the DELIMON command.

Table 78. DELIMON operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 79 lists the return codes for the DELIMON command.

Table 79. DELIMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELMMON

▶▶—DELMMON—NAME (*mmonid*)—————▶▶

Figure 45. DELMMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELMMON command to delete a virtual machine monitor.

**Related reference:**

“DEFMMON” on page 87

## Operands

**NAME (*mmonid*)**

Specifies the name of the machine monitor to delete.

## Operand value lengths

Table 80 lists the operand value lengths for the DELMMON command.

Table 80. DELMMON operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

1. The status of the monitored virtual machine is not changed.

## Return codes

Table 81 lists the return codes for the DELMMON command.

Table 81. DELMMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELPMON

►►—DELPMON—NAME (*pmonid*)—————►►

Figure 46. DELPMON command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELPMON command to delete a page space monitor.

## Operands

**NAME (*pmonid*)**

Specifies the name of the page space monitor to delete.

## Operand value lengths

Table 82 lists the operand value lengths for the DELPMON command.

Table 82. DELPMON operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 83 lists the return codes for the DELPMON command.

Table 83. DELPMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELRULE

▶▶—DELRULE—NAME(*ruleid*)—————▶▶

Figure 47. DELRULE command syntax

### Authorization

The user issuing this command must be authorized for configuration commands.

### Purpose

Issue the DELRULE command to delete a rule.

**Related reference:**

“DEFRULE” on page 94

### Operands

**NAME(*ruleid*)**

Specifies the name of the rule to delete.

### Operand value lengths

Table 84 lists the operand value lengths for the DELRULE command.

Table 84. DELRULE operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

### Usage notes

None.

### Return Codes

Table 85 lists the return codes for the DELRULE command.

Table 85. DELRULE return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELSCHD

▶▶—DELSCHD—NAME(*schdid*)—————▶▶

Figure 48. DELSCHD command syntax



## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELSCHD command to delete a schedule.

### Related reference:

“DEFSCHD” on page 98

## Operands

### NAME(schdid)

Specifies the name of the schedule to delete.

## Operand value lengths

Table 86 lists the operand value lengths for the DELSCHD command.

Table 86. DELSCHD operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 87 lists the return codes for the DELSCHD command.

Table 87. DELSCHD return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## DELSMON

▶▶—DELSMON—NAME(smonid)—▶▶

Figure 49. DELSMON command syntax

### Related reference:

“DEFMON” on page 102

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the DELSMON command to delete a spool usage monitor.

## Operands

### NAME(*smonid*)

Specifies the name of the spool usage monitor to delete.

## Operand value lengths

Table 88 lists the operand value lengths for the DELSMON command.

Table 88. DELSMON operand value lengths

| Operand | Length |
|---------|--------|
| NAME    | 1-8    |

## Usage notes

None.

## Return codes

Table 89 lists the return codes for the DELSMON command.

Table 89. DELSMON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## RESUME

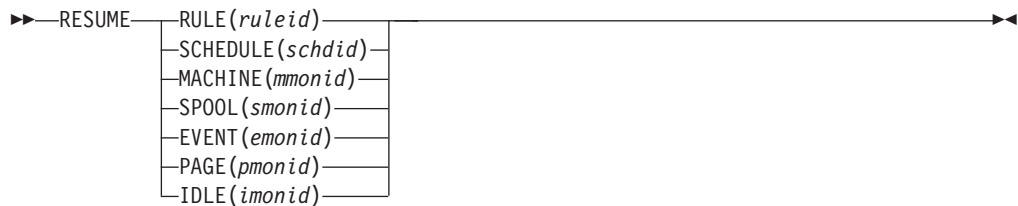


Figure 50. RESUME command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the RESUME command to return a resource from the suspended state.

The valid resources are:

- RULE
- SCHEDULE
- MACHINE
- SPOOL
- EVENT

- PAGE
- IDLE

**Related reference:**

“SUSPEND” on page 121

## Operands

**RULE** Specifies the ruleid from DEFRULE.

**SCHEDULE**

Specifies the schdid from DEFSCHD.

**MACHINE**

Specifies the mmonid from DEFMMON.

**SPOOL** Specifies the smonid from DEFSMON.

**EVENT** Specifies the emonid from DEFEMON.

**PAGE** Specifies the pmonid from DEFPMON.

**IDLE** Specifies the imonid from DEFIMON.

**Related reference:**

“DEFRULE” on page 94

“DEFSCHD” on page 98

“DEFMMON” on page 87

“DEFSMON” on page 102

“DEFEMON” on page 79

“DEFPMON” on page 90

“DEFIMON” on page 82

## Operand value lengths

Table 90 lists the operand value lengths for the RESUME command.

*Table 90. RESUME operand value lengths*

| Operand  | Length |
|----------|--------|
| EVENT    | 1-8    |
| IDLE     | 1-8    |
| MACHINE  | 1-8    |
| PAGE     | 1-8    |
| RULE     | 1-8    |
| SCHEDULE | 1-8    |
| SPOOL    | 1-8    |

## Usage notes

1. One of the resources must be specified.
2. Use ALL in place of ruleid, mmonid, schdid, smonid, emonid, pmonid, or imonid to resume all definitions for the specified resource type. For example, the following command resumes all rules that are in the suspended state:  
RESUME RULE(ALL)

## Return codes

Table 91 lists the return codes for the RESUME command.

Table 91. RESUME return codes

| Return Code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## RUN

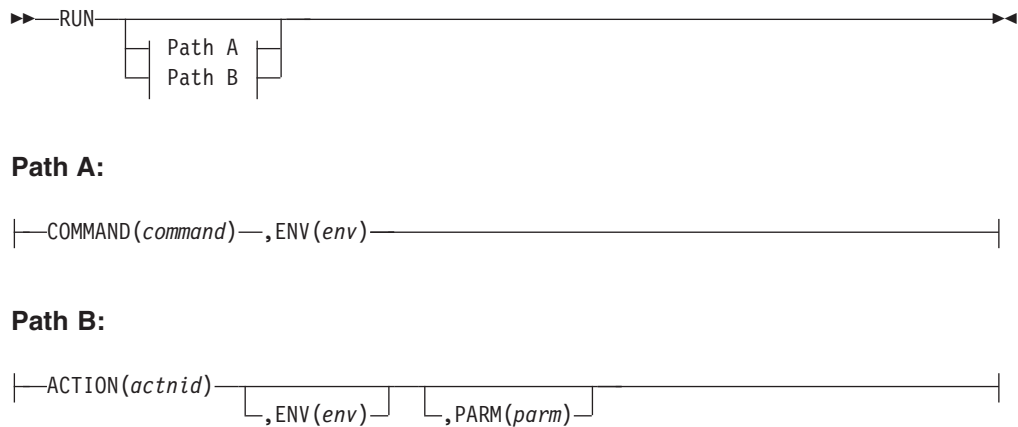


Figure 51. RUN command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the RUN command to run a command or action.

## Operands

### Path A options

#### COMMAND(command)

Specifies the command to be processed. This can be any CMS or CP command.

#### ENV(env)

Specifies the environment where the command or actions must be processed. The valid values for *env* are:

- LVM** Specifies that the command must be processed by the local virtual machine. Use caution when specifying LVM. Any command or module that processes in the LVM preempts processing of Operations manager messages and can possibly affect performance and reliability of Operations Manager.

- SVM** Specifies that the command must be processed by any available action processing server.
- userid** Specifies that the command must be processed by only the indicated action processing server.
- wgrpuid** Specifies that the command must be processed by only the action processing servers participating in the work group.

## Path B options

### **ACTION(actnid)**

Specifies the action to be processed.

### **ENV(env)**

Specifies the processing environment of the action must be overridden with the *env* supplied. The DEFACTN command provides additional information on overriding the processing environment and valid values for *env*.

### **PARM(parm)**

Specifies the parameter passed to the action when it is scheduled for processing. See the DEFACTN command for parameter values.

### **Related reference:**

“DEFACTN” on page 71

## Operand value lengths

Table 92 lists the operand value lengths for the RUN command.

*Table 92. RUN operand value lengths*

| Operand | Length |
|---------|--------|
| ACTION  | 1-8    |
| COMMAND | 1-90   |
| ENV     | 1-8    |
| PARM    | 1-120  |

## Usage notes

None.

## Return codes

Table 93 lists the return codes for the RUN command.

*Table 93. RUN return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

# STATUS

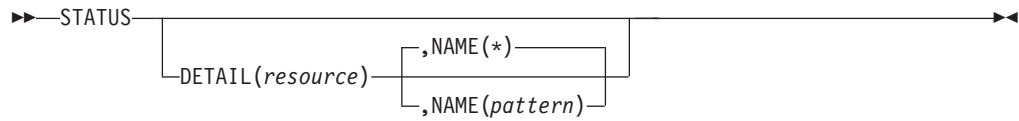


Figure 52. STATUS command syntax

## Authorization

The user issuing this command must be authorized for configuration or control commands.

## Purpose

Issue the STATUS command to show the current status of GOMMAIN processing. If the DETAIL operand is not present a summary is outputted. If the DETAIL operand is present additional information about each specified resource is outputted.

## Operands

### DETAIL(resource)

Requests additional information for the specified resource definitions.

The valid resources are:

- RULE — Displays the status of the defined rules.
- MACHINE — Displays the status of the monitored virtual machines.
- SCHEDULE — Displays the status of the defined schedules.
- ACTION — Displays the status of the defined actions.
- ACTIONQ — Displays the actions queued for execution.
- GROUP — Displays the status of the defined groups.
- DATE — Displays the status of the defined holidays.
- SERVICE — Displays the status of the defined action processing servers.
- SPOOL — Displays the status of the defined spool usage monitors and how many times they were executed.
- SPOOLUSR — Displays the top ten largest spool files and the top ten users with the most spool files.
- AUTH — Displays the status of authorized users.
- IUCV — Displays the status of the active IUCV connections.
- DATASP — Displays the status of the defined data space size creation templates.
- VIEW — Displays the status of the combined views.
- CONSOLE — Displays the status of the recording consoles.
- TCPAPPL — Displays the status of the TCP/IP applications.
- OPTION — Displays the status of the current options settings.
- EVENT — Displays the status of the defined event monitors.
- PAGE — Displays the status of the defined page space monitors.
- IDLE — Displays the status of the defined idle monitors.

### NAME(pattern)

Lists information for resources names which match the pattern. Wildcard patterns are allowed. The wildcard pattern rules are explained in the DEFULE command.

### Related reference:

“DEFRULE” on page 94

## Operand value lengths

Table 94 lists the operand value lengths for the STATUS command.

Table 94. STATUS operand value lengths

| Operand | Length |
|---------|--------|
| DETAIL  | 1-8    |
| NAME    | 1-8    |

## Examples

Figure 53 shows summary output from the STATUS command.

```
STATUS
2014-10-24 16:04:59 GOMCMD0899I OPERATIONS MANAGER V1R5 BUILT 2014/10/16 19:29
2014-10-24 16:04:59 GOMCMD0900I 13 RULE(S) DEFINED
2014-10-24 16:04:59 GOMCMD0902I 5 MACHINE MONITOR(S) DEFINED
2014-10-24 16:04:59 GOMCMD0903I 661 SCHEDULE(S) DEFINED
2014-10-24 16:04:59 GOMCMD0904I 52 ACTION(S) DEFINED
2014-10-24 16:04:59 GOMCMD0905I 5 ACTION(S) SCHEDULED FOR PROCESSING
2014-10-24 16:04:59 GOMCMD0906I 3 GROUP(S) DEFINED
2014-10-24 16:04:59 GOMCMD0907I 2 DATES(S) DEFINED
2014-10-24 16:04:59 GOMCMD0908I 105 SPOOL MONITOR(S) DEFINED
2014-10-24 16:04:59 GOMCMD0909I 3 EVENT MONITOR(S) DEFINED
2014-10-24 16:04:59 GOMCMD0910I 105 PAGE MONITOR(S) DEFINED
2014-10-24 16:04:59 GOMCMD0911I 1 IDLE MONITOR(S) DEFINED
2014-10-24 16:04:59 GOMCMD0920I 4 SERVICE MACHINE(S) DEFINED
2014-10-24 16:04:59 GOMCMD0921I 3 USER(S) AUTHORIZED
2014-10-24 16:04:59 GOMCMD0922I 11 IUCV CONNECTION(S)
2014-10-24 16:04:59 GOMCMD0923I 4 DATA SPACE SIZE DEFINITION(S)
2014-10-24 16:04:59 GOMCMD0924I 3 COMBINED VIEW(S)
2014-10-24 16:04:59 GOMCMD0925I 6 USER CONSOLE(S) RECORDING
2014-10-24 16:04:59 GOMCMD0926I 10 TCP/IP APPLICATION(S) DEFINED
```

Figure 53. Summary output from STATUS command

Figure 54 shows details about all monitored virtual machines.

```
STATUS DETAIL(MACHINE)
2014-10-24 16:08:54 GOMCMD0942I MACHINE MONITOR MONI TRIGGERED 0 TIME(S)
2014-10-24 16:08:54 GOMCMD2977I ACTION MACHAC
2014-10-24 16:08:54 GOMCMD2976I DAYS (ALL)
2014-10-24 16:08:54 GOMCMD2975I HOURS (ALL)
2014-10-24 16:08:54 GOMCMD2973I USER RSCS (RSCS - DSC)
2014-10-24 16:08:54 GOMCMD2972I INTERVAL 1 MINUTE(S)
2014-10-24 16:08:54 GOMCMD2971I DELAY 0 MINUTE(S)
2014-10-24 16:08:54 GOMCMD2970I RETRY INTERVAL 1 MINUTE(S) RETRY COUNT 0
2014-10-24 16:08:54 GOMCMD2968I NEXT CHECK 2014/10/24 16:09
2014-10-24 16:08:54 GOMCMD2999I STATUS DETAIL COMPLETE
```

Figure 54. Details about all monitored virtual machines

Figure 55 on page 120 shows details about all spool usage monitors and how many times they were executed.

```

STATUS DETAIL(SPOOL)
2014-10-24 16:11:50 GOMCMD0948I SPOOL MONITOR WARN TRIGGERED 0 TIME(S)
2014-10-24 16:11:50 GOMCMD2977I ACTION ACTNSP
2014-10-24 16:11:50 GOMCMD2976I DAYS (ALL)
2014-10-24 16:11:50 GOMCMD2975I HOURS (ALL)
2014-10-24 16:11:50 GOMCMD2967I USAGE LOW 80 HIGH 89
2014-10-24 16:11:50 GOMCMD2972I INTERVAL 1 MINUTE(S)
2014-10-24 16:11:50 GOMCMD2968I NEXT CHECK 2014/10/24 16:12
2014-10-24 16:11:50 GOMCMD2999I STATUS DETAIL COMPLETE
2014-10-24 16:12:07 GOMCMD0948I SPOOL MONITOR CRIT TRIGGERED 0 TIME(S)
2014-10-24 16:12:07 GOMCMD2977I ACTION ACTNSP
2014-10-24 16:12:07 GOMCMD2976I DAYS (ALL)
2014-10-24 16:12:07 GOMCMD2975I HOURS (ALL)
2014-10-24 16:12:07 GOMCMD2967I USAGE LOW 90 HIGH 100
2014-10-24 16:12:07 GOMCMD2972I INTERVAL 1 MINUTE(S)
2014-10-24 16:12:07 GOMCMD2968I NEXT CHECK 2014/10/24 16:12
2014-10-24 16:12:07 GOMCMD2999I STATUS DETAIL COMPLETE

```

Figure 55. Details about all spool usage monitors

Figure 56 shows the top ten largest spool files and the top ten users with the most spool files.

```

status detail(spoolusr)
2014-10-24 16:14:19 GOMCMD0970I USER SAMARAK SPOOL FILE ID 0124 IS USING 9672 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER BKRCATLG SPOOL FILE ID 0001 IS USING 8737 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER MAINT620 SPOOL FILE ID 0662 IS USING 6763 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER RVBACKUP SPOOL FILE ID 0001 IS USING 5195 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER SAMARAK SPOOL FILE ID 0037 IS USING 5051 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER LNXSLES SPOOL FILE ID 0004 IS USING 4901 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3319 IS USING 4817 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3325 IS USING 4813 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3320 IS USING 4811 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0970I USER RWHITE SPOOL FILE ID 3321 IS USING 4811 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER BKRADMIN HAS 364 SPOOL FILES USING 9188 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER MAINT620 HAS 312 SPOOL FILES USING 9131 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER MAINT HAS 231 SPOOL FILES USING 1102 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER DMARTIN HAS 220 SPOOL FILES USING 8648 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER TCPMAINT HAS 160 SPOOL FILES USING 1801 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER OPERATOR HAS 131 SPOOL FILES USING 2626 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER IGOR HAS 125 SPOOL FILES USING 933 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER SAMARAK HAS 92 SPOOL FILES USING 21314 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER MDMARA HAS 87 SPOOL FILES USING 2429 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD0971I USER RACFVM HAS 85 SPOOL FILES USING 158 SPOOL BLOCKS
2014-10-24 16:14:19 GOMCMD2999I STATUS DETAIL COMPLETE

```

Figure 56. Details about users' spool files

## Usage notes

None.

## Return codes

Table 95 lists the return codes for the STATUS command.

Table 95. STATUS return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |



---

# SUSPEND

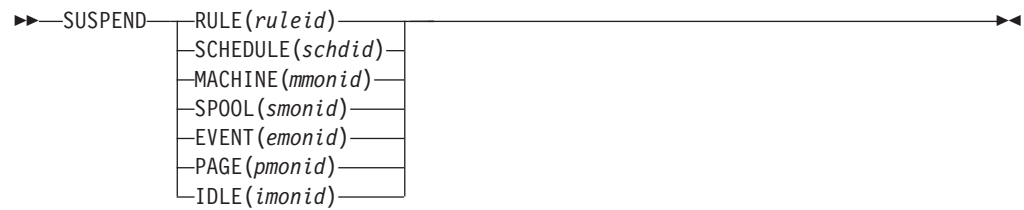


Figure 57. SUSPEND command syntax

## Authorization

The user issuing this command must be authorized for configuration commands.

## Purpose

Issue the SUSPEND command to place a resource in the suspended state. A resource in the suspended state is ignored for processing.

The valid resources are:

- RULE
- SCHEDULE
- MACHINE
- SPOOL
- EVENT
- PAGE
- IDLE

**Related reference:**

“RESUME” on page 114

## Operands

**RULE** Specifies the *ruleid* from DEFRULE.

**SCHEDULE** Specifies the *schdid* from DEFSCHD.

**MACHINE** Specifies the *mmonid* from DEFMMON.

**SPOOL** Specifies the *smonid* from DEFSMON.

**EVENT** Specifies the *emonid* from DEFEMON.

**PAGE** Specifies the *pmonid* from DEFPMON.

**IDLE** Specifies the *imonid* from DEFIMON.

**Related reference:**

“DEFRULE” on page 94

“DEFSCHD” on page 98

“DEFMMON” on page 87

“DEFSMON” on page 102

“DEFEMON” on page 79

“DEFPMON” on page 90

“DEFIMON” on page 82

## Operand value lengths

Table 96 lists the operand value lengths for the SUSPEND command.

Table 96. SUSPEND operand value lengths

| Operand  | Length |
|----------|--------|
| EVENT    | 1-8    |
| IDLE     | 1-8    |
| MACHINE  | 1-8    |
| PAGE     | 1-8    |
| RULE     | 1-8    |
| SCHEDULE | 1-8    |
| SPOOL    | 1-8    |

## Usage notes

1. One of the resources must be specified.
2. Use ALL in place of ruleid, mmonid, schdid, smonid, emonid, pmonid, or imonid to suspend all definitions for the specified resource type. For example, the following command suspends all defined rules:

```
SUSPEND RULE(ALL)
```

## Return codes

Table 97 lists the return codes for the RESUME command.

Table 97. SUSPEND return codes

| Return Code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## Chapter 7. Syntax for Console, Log, and Spool Access Commands

This chapter describes the Operations Manager console, log, and spool access commands and their syntax.

---

### Purpose of console, log, and spool access commands

Console, log, and spool access commands allow you to view and modify a user's console display, the Operations Manager log, and a user's spool files. With a console, log, or spool access command, you can perform various tasks such as issue commands on a user's console, display the console or log in full-screen mode, modify the attributes in a console, add comments in a log, and view a spool list. You can also create a combined view.

Use GOMCMD to issue console, log, and spool access commands.

**Related reference:**

"GOMCMD" on page 32

---

### Authorization for console, log, and spool access commands

Authorization varies for console, log and spool access commands. Refer to the specific command for the required authorization.

---

### Console data spaces

Operations Manager uses z/VM data spaces to contain the following types of output:

- Console output from a monitored user ID (for example, output sent to Operations Manager via SECUSER or OBSERVER).
- Combined views of console output defined in the DEFVIEW command.
- The Operations Manager log.
- Output and information related to the execution of schedules, machine monitors, spool monitors, event monitors, and the output from actions.

Use the VIEWCON command to look at any of these data spaces except the logging data space. You must use the VIEWLOG command to look at the Operations Manager log data space.

Table 98 lists reserved data space names and the corresponding execution type.

*Table 98. Reserved data space names*

| Execution type      | Data space name |
|---------------------|-----------------|
| Event monitors      | _GOMEMON        |
| Idle monitors       | _GOMIMON        |
| Log                 | _GOMALOG        |
| Machine monitors    | _GOMMMON        |
| Output from actions | _GOMACTN        |

Table 98. Reserved data space names (continued)

| Execution type      | Data space name |
|---------------------|-----------------|
| Page space monitors | _GOMPMON        |
| Schedules           | _GOMSCHD        |
| Spool monitors      | _GOMSMON        |

**Related reference:**

“VIEWCON” on page 127

“VIEWLOG” on page 131

## ALTRCON

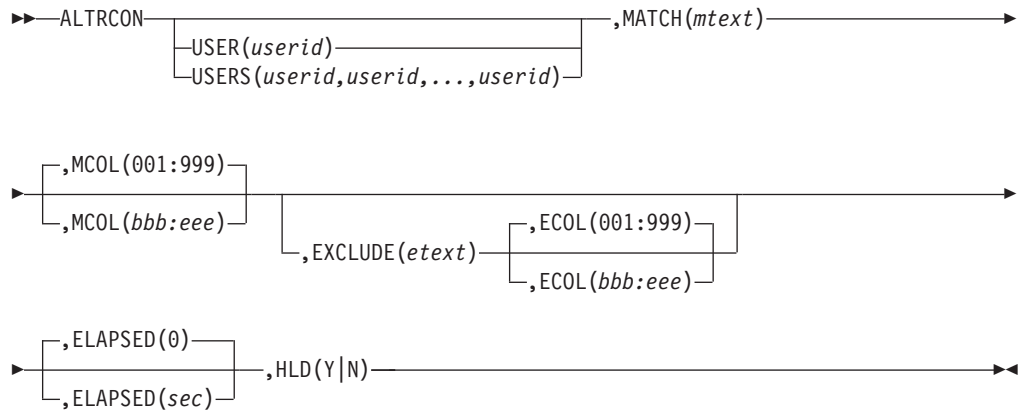


Figure 58. ALTRCON command syntax

### Authority

The user issuing this command must be authorized for console access for the specified *userid* or view.

### Purpose

Issue the ALTRCON command to modify the attributes of a line in a user's console when it is displayed using VIEWCON.

The ALTRCON command can also be issued to modify the attributes of a line in a combined view created with a DEFVIEW command.

**Related reference:**

“DEFVIEW” on page 55

“VIEWCON” on page 127

### Operands

**USER(userid)**

Specifies the *userid* or view name for the console you want to modify.

**USERS(userid,userid,...,userid)**

Specifies up to 16 userids (consoles) to modify. The userid can be a wildcard pattern. The wildcard rules are explained in the DEFRULE command.

**MATCH(mtext)**

Specifies the pattern used to compare against the lines to determine if there is a match. Wildcard pattern matching is allowed. The wildcard rules are explained in the DEFRULE command.

**MCOL(bbb:eee)**

Specifies the beginning and ending columns to search for the match pattern (inclusive). The format is bbb:eee, where bbb is the beginning column and eee is the ending column.

**EXCLUDE(etext)**

Specifies the pattern used to compare against the lines to determine if there is an exclusion. Wildcard pattern matching is allowed. The wildcard rules are explained in the DEFRULE command.

**ECOL(bbb:eee)**

Specifies the beginning and ending columns to search for the exclude pattern (inclusive). The format is bbb:eee, where bbb is the beginning column and eee is the ending column.

**ELAPSED(sec)**

Specifies the minimum amount of elapsed time since a line was logged for a line to be eligible for modification. Only lines which have been in the console longer than the number of elapsed seconds are eligible for modification. The value must be between 0 and 999999.

**HLD(Y|N)**

Specifies the setting of the HLD attribute.

**Related reference:**

“DEFRULE” on page 94

## Operand value lengths

Table 99 lists the operand value lengths for the ALTRCON command.

*Table 99. ALTRCON operand value lengths*

| Operand | Length |
|---------|--------|
| ECOL    | 7      |
| ELAPSED | 1-6    |
| EXCLUDE | 1-60   |
| HLD     | 1      |
| MATCH   | 1-60   |
| MCOL    | 7      |
| USER    | 1-8    |
| USERS   | 1-120  |

## Usage notes

Either USER or USERS must be specified.

## Return codes

Table 100 lists the return codes for the ALTRCON command.

Table 100. ALTRCON return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## LOGTEXT

►►—LOGTEXT—TEXT(*comment*)—◄◄

Figure 59. LOGTEXT command syntax

## Authorization

No specific authority is required for this command.

## Purpose

Issue the LOGTEXT command to add a comment to the Operations Manager log file.

**Related reference:**

“VIEWLOG” on page 131

## Operands

**TEXT(*comment*)**

Specifies the comment text that appears in the log.

## Operand value lengths

Table 101 lists the operand value lengths for the LOGTEXT command.

Table 101. LOGTEXT operand value lengths

| Operand | Length |
|---------|--------|
| TEXT    | 1-120  |

## Usage notes

None.

## Return codes

Table 102 lists the return codes for the LOGTEXT command.

Table 102. LOGTEXT return codes

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

# VIEWCON

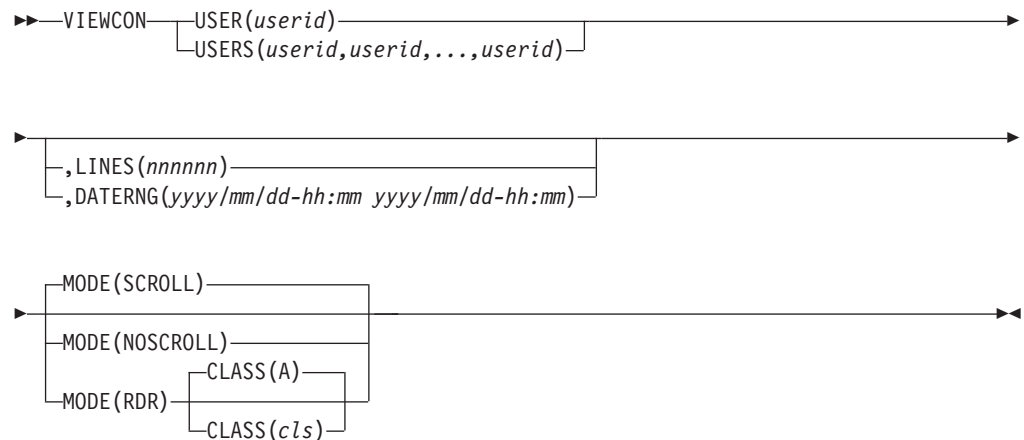


Figure 60. VIEWCON command syntax

## Authorization

The user issuing this command must be authorized for console access for the specified `userid(s)` or view.

### Related tasks:

“Step 5. (Optional) Configure an external security mechanism” on page 11

### Related reference:

“AUTH” on page 39

## Purpose

Issue the VIEWCON command to display the current console content for a monitored user in full-screen mode and to issue commands on that user's console or place a copy of the current console in your virtual reader.

The VIEWCON command may also be issued to display a combined view created with a DEFVIEW command. Commands may be issued to any monitored user combined into the view. The target `userid` and a space must prefix the command.

### Related reference:

“DEFVIEW” on page 55

## Operands

### USER(userid)

Specifies the user ID or view name for the console you want to display.

There are five reserved console data spaces available to use with VIEWCON USER. Refer to “Console data spaces” on page 123 for more information on the reserved console data space names.

**Note:** You cannot use VIEWCON USER to view the logging data space. You must use the VIEWLOG command to view the logging console.

### USERS(userid,userid,...,userid)

Specifies the userids for up to sixteen (16) consoles to view. The `userid`

can be a wildcard pattern. The `userid` is not matched against views. The console information is presented by creating a combined view for the `userids`. The wildcard rules are explained in the `DEFRULE` command.

**LINES(nnnnnn)**

Specifies the number of lines you wish to view. The last *nnnnnn* lines are displayed. The default is to view all lines.

**DATERNG(yyyy/mm/dd-hh:mm yyyy/mm/dd-hh:mm)**

Specifies the beginning and ending date and time range of lines to display within the current console content. The default is to view all lines of the current console content. If the ending date and time specified is earlier than the current date and time, no new messages will be displayed.

**MODE(SCROLL)**

Specifies the initial view is autoscrolling. `SCROLL` is the default.

**MODE(NOScroll)**

Specifies the initial view is not autoscrolling.

**MODE(RDR)**

Specifies that a copy of the current console is to be placed in your virtual reader.

**CLASS(c1s)**

Specifies the reader file class. The default is class A.

**Related reference:**

“`DEFRULE`” on page 94

## Operand value lengths

Table 103 lists the operand value lengths for the `VIEWCON` command.

*Table 103. VIEWCON operand value lengths*

| Operand | Length |
|---------|--------|
| CLASS   | 1      |
| LINES   | 1-6    |
| MODE    | 3-8    |
| USER    | 1-8    |
| USERS   | 1-120  |

## Usage notes

1. Either `USER` or `USERS` must be specified.
2. The date format displayed in `VIEWCON` will be based on the `CP DATEFORMAT` setting of the user issuing the `VIEWCON` command. If `MODE(RDR)` is specified, the date format will be provided in `yyyy/mm/dd` format.
3. `LINES` and `DATERNG` are mutually exclusive.
4. The name of the combined view created when using the `USERS` operand begins with `$TMP`. Issue the `DEFDSPS` command to control the data space size for the combined view.
5. You can use the following default PF keys to move through the console display:
  - PF01 — Show the help screen



- PF03 — End the VIEWCON session
- PF05 — Toggle the acknowledgment for a line in the console.  
To clear an existing hold status, press the PF05 key again.
- PF06 — Toggle the date/time format displayed
- PF07 — Scroll up
- PF08 — Scroll down
- PF10 — Scroll left
- PF11 — Scroll right
- PF12 — Retrieve previous command
- PF19 — Scroll to the top
- PF20 — Scroll to the bottom
- PF21 — Process CMS command
- PF22 — Display the users viewing the console
- PF23 — Toggle autoscrolling
- PF24 — Display the PF key assignments
- PA02 — Toggle autoscrolling
- ENTER — Process the command on the command line

**Note:** For the scrolling PF keys, you can enter a scrolling amount on the command line. If no amount is specified, the scroll amount is equal to the size of the viewing area.

A command may be typed on the command line for the process CMS command action.

6. If an external security manager is being used, for example DEFOPTN ESM(Y),AUTHCMD(N) or DEFOPTN ESM(Y),AUTHCMD(Y), the user must have READ access to the profile ESM.PREF.ALTRCON.*userid* for authorization to use the HOLD/UNHOLD toggle function. Refer to “FACILITY class profiles” on page 13 for more information. If an external security manager is not being used, no authorization checking is done for the HOLD/UNHOLD toggle function.
7. The ALTRON command and the VIEWCON HOLD/UNHOLD toggle perform the same function.
8. The amount of data displayed or sent to your virtual reader is set by the user's console or view data space size.
9. To search through the display, use the forward slash (/) to search forward and the back slash (\) to search backward. Use the equal sign (=) to repeat the previous command. Issue the DEFOPTN command to change these characters.
10. The actions for PF01–PF24, initial value for the date/time format display, and display of the PF key information lines are configurable through the VIEWCON PROFILE file.
  - a. Issue the following command in the VIEWCON PROFILE file to customize the actions for PF01-PF24:

*Pfnn=action*

Where *nn* is 01 through 24 and *action* is a valid PF key action.

The valid PF key actions are:

- HELP — Show the help screen
- END — End the VIEWCON session
- HOLD — Toggle the acknowledgment for a line in the console
- FORMAT — Toggle the date/time format displayed

- UP — Scroll up
- DOWN — Scroll down
- LEFT — Scroll left
- RIGHT — Scroll right
- RECALL — Retrieve previous command
- TOP — Scroll to top
- BOT — Scroll to bottom
- USERS — Display the users viewing the console
- SCROLL — Toggle autoscrolling
- VIEWPF — Display the PF key assignments
- EXCMD — Process CMS command (the command to process may be included)

Refer to usage note 9 for more information.

- b. Issue one of the following commands in the VIEWCON PROFILE file to customize the initial date/time format display:
- FORMAT=0 — Displays the console data with no date and no time. This is the default.
  - FORMAT=1 — Displays the console data with time.
  - FORMAT=2 — Displays the console data with date and time.
  - FORMAT=3 — Displays the console data with date, time and message class.
  - FORMAT=4 — Displays the console data with time and message class.
  - FORMAT=5 — Displays the console data with message class.
- c. Issue the following command in the VIEWCON PROFILE file to control the display of the PF key information lines:

```
SHOWPF=c[,h,xat,clr]
```

Where *c* is one of the following options:

- N — Do not display the PF key assignments on the screen.
- Y — Display PF01–PF12 assignments on the screen.
- A — Display PF01–PF24 assignments on the screen.

The values for *h,xat,clr* are optional. If one of the values is specified, they must all be specified.

- *h* — Specifies if the line is highlighted. Values are Y or N.
- *xat* — Specifies the extended attributes applied to the line. Values are:
  - AAI — Leave as is
  - ABL — Blinking
  - ARV — Reverse video
  - AUL — Underline
- *clr* — Specifies the color applied to the line. Values are:
  - CAI — Leave as is
  - CBL — Blue
  - CRE — Red
  - CPI — Pink
  - CGR — Green
  - CCY — Cyan
  - CYE — Yellow

- CWH — White

The following shows an example of what might be in a VIEWCON PROFILE file after customizing a PF key assignment, the initial date/time format, and the display of the PF keys:

```
PF13=HELP
FORMAT=1
SHOWPF=Y,Y,AAI,CAI
```

11. While using VIEWCON, you can also execute a z/VM or Operations Manager command by using EXCMD. Execute the EXCMD command with one of the following methods:

- Press a function key that is assigned to EXCMD without any command on the command line. This invokes CMS SUBSET. To return to VIEWCON, use the CMS command return.
- Press a function key that is assigned to EXCMD with a CMS command on the command line.

When the command completes, if the command on the command line is not an Operations Manager command, you return to the originating VIEWCON command.

When the command completes, if the command on the command line is an Operations Manager command, you return to the point that originated the VIEWCON command.

- Assign EXCMD with a z/VM or Operations Manager command to a PF key in VIEWCON PROFILE. For example, PF19=EXCMD LISTFILE \* \* A or PF20=EXCMD GOMCMD OPMGRM1 VIEWLOG.

When the command completes, if the command assigned to the PF key is not an Operations Manager command, you return to the originating VIEWCON command.

When the command completes, if the command assigned to the PF key is an Operations Manager command, you return to the point that originated the VIEWCON command.

**Related reference:**

“DEFDPS” on page 46

“DEFOPTN” on page 48

## Return codes

Table 104 lists the return codes for the VIEWCON command.

Table 104. VIEWCON return codes

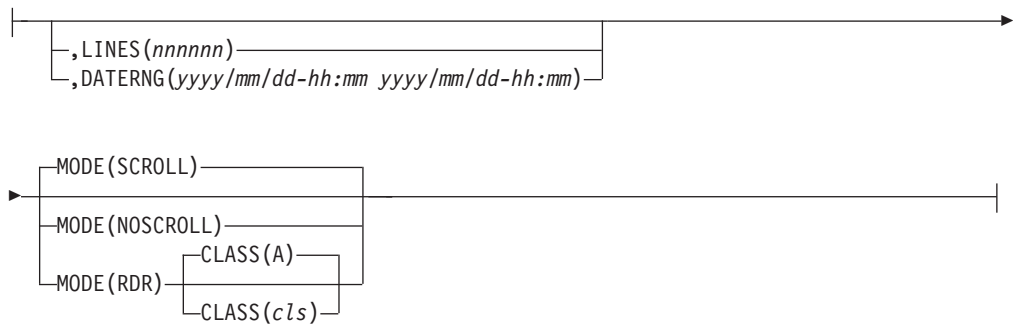
| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## VIEWLOG



### PATH A:



### PATH B:

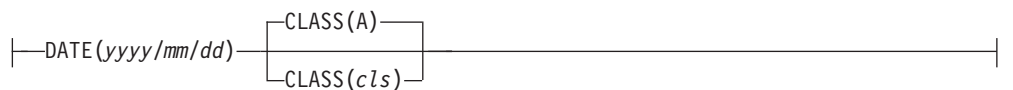


Figure 61. VIEWLOG command syntax

## Authorization

The user issuing this command must be authorized for control commands.

## Purpose

Issue the VIEWLOG command to display the current Operations Manager log in full-screen mode and to issue Operations Manager commands or place a copy of the log in your virtual reader.

You can also issue the VIEWLOG command to obtain a DISK DUMP of the log for a specific day.

## Operands

### LINES (nnnnn)

Specifies the number of lines you wish to view. The last *nnnnn* lines are displayed. The default is to view all lines.

### DATERNG (yyyy/mm/dd-hh:mm yyyy/mm/dd-hh:mm)

Specifies the beginning and ending date and time range of lines to display within the current console content. The default is to view all lines of the current console content. If the ending date and time are earlier than the current date and time, no new lines will be displayed.

### MODE (SCROLL)

Specifies the initial view is autoscrolling. SCROLL is the default.

### MODE (NOSCROLL)

Specifies the initial view is not autoscrolling.

### MODE (RDR)

Specifies that a copy of the active VIEWLOG log is to be placed in your virtual reader.

**DATE(yyyy/mm/dd)**

Specifies that a copy of the log for an entire day is to be placed in your virtual reader. Use the format yyyy/mm/dd, where:

- yyyy. The year, which must be specified with four digits.
- mm. The month (01-12).
- dd. The day of the month (01-31).

**CLASS(c1s)**

Specifies the reader file class. Class A is the default.

## Operand value lengths

Table 105 lists the operand value lengths for the VIEWLOG command.

*Table 105. VIEWLOG operand value lengths*

| Operand | Length |
|---------|--------|
| CLASS   | 1      |
| DATE    | 10     |
| LINES   | 1-6    |
| MODE    | 3-8    |

## Usage notes

1. LINES and DATERNG are mutually exclusive.
2. The date format displayed in VIEWLOG will be based on the CP DATEFORMAT setting of the user issuing VIEWLOG. If MODE(RDR) or DATE is specified, the date format will be provided in yyyy/mm/dd format.
3. You can use the following PF keys to move through the log display:
  - PF01 — Show the help screen
  - PF03 — End the VIEWLOG session
  - PF05 — Toggle the acknowledgment for a line in the log
  - PF07 — Scroll up
  - PF08 — Scroll down
  - PF10 — Scroll left
  - PF11 — Scroll right
  - PF12 — Retrieve previous command
  - PF19 — Scroll to the top
  - PF20 — Scroll to the bottom
  - PF21 — Process CMS command
  - PF22 — Display the users viewing the log
  - PF23 — Toggle autoscrolling
  - PF24 — Display the PF key assignments
  - PA02 — Toggle autoscrolling
  - ENTER — Process the command on the command line

**Note:** For the scrolling PF keys, you can enter a scrolling amount on the command line. If no amount is specified, the scroll amount is equal to the size of the viewing area.

A command may be typed on the command line for the process CMS command action.

4. The amount of data displayed or sent to your virtual reader from the log is limited to the size of the logging data space. The `logsize` parameter on `GOMMAIN` determines this limit. If you want to view the entire log use the `DATE` operand to obtain a log of the entire day.
5. To search through the display, use the forward slash (/) to search forward and the back slash (\) to search backward. Use the equal sign (=) to repeat the previous command. Issue the `DEFOPTN` command to change these characters.
6. The actions for PF01 - PF24 and initial value for the PF key display are configurable through the `VIEWLOG PROFILE` file.
  - a. Issue the following command in the `VIEWLOG PROFILE` file to set the PF key assignments:

`PFnn=action`

Where *nn* is 01 through 24 and *action* is a valid PF key action.

The valid PF key actions are:

- HELP — Show the help screen
- END — End the `VIEWLOG` session
- HOLD — Toggle the acknowledgment for a line in the log
- UP — Scroll up
- DOWN — Scroll down
- LEFT — Scroll left
- RIGHT — Scroll right
- RECALL — Retrieve previous command
- TOP — Scroll to top
- BOT — Scroll to bottom
- USERS — Display the users viewing the log
- SCROLL — Toggle autoscrolling
- VIEWPF — Display the PF key assignments
- EXCMD — Process a CMS command (the command to process may be included)

Refer to usage note 5 for more information.

- b. Issue the following command in the `VIEWLOG PROFILE` file to control the display of the PF key information lines:

`SHOWPF=c[,h,xat,clr]`

Where *c* is one of the following options:

- N — Do not display the PF key assignments on the screen.
- Y — Display PF01–PF12 assignments on the screen.
- A — Display PF01–PF24 assignments on the screen.

The values for *h,xat,clr* are optional. If one of the values is specified, they must all be specified.

- h — Specifies if the line is highlighted. Values are Y or N.
- xat — Specifies the extended attributes applied to the line. Values are:
  - AAI — Leave as is
  - ABL — Blinking
  - ARV — Reverse video
  - AUL — Underline

- `clr` — Specifies the color applied to the line. Values are:
  - CAI — Leave as is
  - CBL — Blue
  - CRE — Red
  - CPI — Pink
  - CGR — Green
  - CCY — Cyan
  - CYE — Yellow
  - CWH — White

The following shows an example of what might be in a VIEWLOG PROFILE file after customizing a PF key assignment and the display of the PF keys:

```
PF13=HELP
SHOWPF=Y,Y,AAI,CAI
```

7. While using VIEWLOG, you can also execute a z/VM or Operations Manager command by using EXCMD. Execute the EXCMD command with one of the following methods:
  - Press a function key that is assigned to EXCMD without any command on the command line. This invokes CMS SUBSET. To return to VIEWLOG, use the CMS command return.
  - Press a function key that is assigned to EXCMD with a CMS command on the command line.

When the command completes, if the command on the command line is not an Operations Manager command, you return to the originating VIEWLOG command.

When the command completes, if the command on the command line is an Operations Manager command, you return to the point that originated the VIEWLOG command.

- Assign EXCMD with a z/VM or Operations Manager command to a PF key in VIEWLOG PROFILE. For example, `PF19=EXCMD LISTFILE * * A` or `PF20=EXCMD GOMCMD OPMGRM1 VIEWSP`.

When the command completes, if the command assigned to the PF key is not an Operations Manager command, you return to the originating VIEWLOG command.

When the command completes, if the command assigned to the PF key is an Operations Manager command, you return to the point that originated the VIEWLOG command.

**Related reference:**

“DEFOPTN” on page 48

“GOMMAIN” on page 31

## Return codes

Table 106 lists the return codes for the VIEWLOG command.

*Table 106. VIEWLOG return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

# VIEWSPL

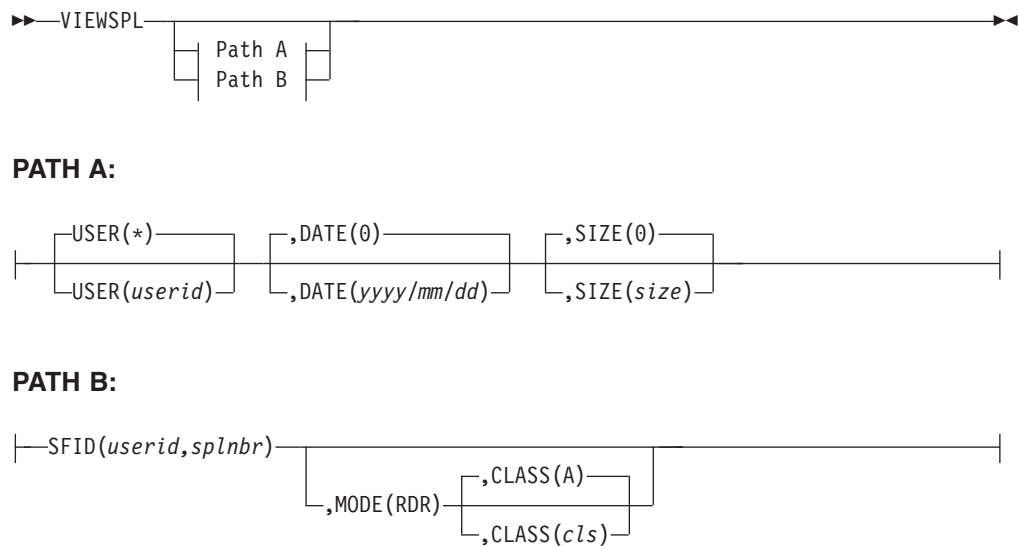


Figure 62. VIEWSPL command syntax

## Authorization

The user issuing this command must have authorization to access the spool of the specified `userid`.

### Related tasks:

“Step 5. (Optional) Configure an external security mechanism” on page 11

### Related reference:

“AUTH” on page 39

## Purpose

Issue the VIEWSPL command to display a list of spool files, view the contents of a spool file in full-screen mode, or place a copy of the contents of a spool file in your virtual reader. .

## Operands

### USER(userid)

Specifies the `userid` whose spool files you want to display. The `userid` can be a wildcard pattern. The wildcard pattern rules are explained in the DEFRULE command.

### DATE(yyyy/mm/dd)

Specifies the earliest creation date of the spool files you want to display. Use the format `yyyy/mm/dd`, where:

- `yyyy`. The year, which must be specified with four digits.
- `mm`. The month (01-12).
- `dd`. The day of the month (01-31).



**SIZE(size)**

Specifies the minimum size of the spool files to display. The size must be a number in bytes or suffixed with K for kilobytes, M for megabytes, or G for gigabytes.

**SFID(userid,sp1nbr)**

Specifies that you want to view the spool file contents in full-screen mode and skip viewing the list of spool files. If SFID is used with the MODE(RDR) operand, a copy of the spool file contents are placed in your virtual reader. The userid cannot contain wildcards. The sp1nbr is the 4-digit spool file number. If the spool file number contains leading zeros, you can abbreviate the number. For example, if the spool file number is 0003, you can specify 3.

**MODE(RDR)**

Specifies that a copy of the current spool file is to be placed on your virtual reader.

**CLASS(c1s)**

Specifies the reader file class. The default is class A.

**Related reference:**

“DEFRULE” on page 94

## Operand value lengths

Table 107 lists the operand value lengths for the VIEWSPPL command.

Table 107. VIEWSPPL operand value lengths

| Operand | Length |
|---------|--------|
| CLASS   | 1      |
| DATE    | 10     |
| MODE    | 3-8    |
| SFID    | 3-13   |
| SIZE    | 1-10   |
| USER    | 1-8    |

## Usage notes

### General usage notes

The date format displayed in VIEWSPPL will be based on the CP DATEFORMAT setting of the user that issued VIEWSPPL. If MODE(RDR) is specified, the date format will be provided in yyyy/mm/dd format.

### Viewing spool file list

- You can use the following default PF keys to move through the displayed spool file list:
  - PF01 — Show the help screen
  - PF02 — View the spool file contents
  - PF03 — End the VIEWSPPL session
  - PF05 — Sort the files in ascending order
  - PF06 — Sort the files in descending order
  - PF07 — Scroll up

- PF08 — Scroll down
  - PF10 — Scroll left
  - PF11 — Scroll right
  - PF19 — Scroll to top
  - PF20 — Scroll to bottom
  - PF21 — Process CMS command
  - PF24 — Display the PF key assignments
  - ENTER — Process the command in the command field
2. The list of spool files includes:
    - files allowed by the \*SPL GENERIC interface
    - PRT files
    - RDR files
    - PUN files
    - DMP files
    - HELD files (USER or SYS HOLD)
    - OPEN files
  3. The list of spool files does not include:
    - HELP files
    - SYS files (DMP, NSS, NLS, and so on)
  4. Available CP commands in the spool file list:
    - a. Issue the PURGE, CHANGE, and TRANSFER commands to manipulate the spool files. Enter these commands followed by a forward slash (/) and the command operands. Refer to the *CP Commands and Utilities Reference (SC24-6081)* for more information and the valid abbreviations for these commands.
    - b. To repeat the previous command, type the equal sign (=) in the command field.
    - c. To sort a column, place the cursor on the column heading and use the PF05 or PF06 key.
    - d. The number of spool files selected is limited to 25,000 files.
    - e. The value for space utilization percentage is obtained by rounding downward. This value may differ from the space utilization percentage computed from other sources.
  5. The spool file list can contain the following indicators before the spool file number:

Table 108. Spool file list indicators

| Indicator | Description                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------|
| ' '       | (no indicator) Local file that can be viewed                                                                                    |
| ' *'      | Local file that cannot be viewed using the PF02 key, or whichever key is assigned the VIEW command from the VIEWSPL PROFILE     |
| ' +'      | Non-local file that can be viewed                                                                                               |
| '*+'      | Non local file that cannot be viewed using the PF02 key, or whichever key is assigned the VIEW command from the VIEWSPL PROFILE |

A file cannot be viewed for one of the following reasons:

- the file is HELD (SYS and/or USER)
- the file was created by CP (DMP, and so on)

- the file is currently open and in use

The following screen is an example of a spool file list, where file 0145 is local and viewable, file 0029 is local and non-viewable, file 0142 is non-local and viewable, and file 0082 is non-local and non-viewable:

| System: RS54 |          | Spool: 55% Used |     | Files: 0% Used |     | 196 of 202 |           |            |       |      |
|--------------|----------|-----------------|-----|----------------|-----|------------|-----------|------------|-------|------|
|              |          | Max: 2.4G       |     | Max: 1655640   |     |            |           |            |       |      |
| Cmd          | Owner    | File            | CLS | QUE            | TYP | Size       | Hold Date | Time       | Name  | Type |
|              | MAINT620 | 0145            | T   | RDR            | CON | 4K         | NONE      | 10/26/2012 | 17:52 |      |
|              | OPERATOR | *0029           | T   | PRT            | CON | 88K        | NONE      | 12/05/2012 | 18:41 |      |
|              | MAINT620 | +0142           | A   | PRT            | PRT | 4K         | NONE      | 05/08/2013 | 13:20 |      |
|              | RVBACKUP | **0082          | T   | PRT            | CON | 1M         | USER      | 05/24/2013 | 13:32 |      |

6. While using VIEWSP, you can also execute a z/VM or Operations Manager command by using EXCMD. Execute the EXCMD command with one of the following methods:

- Press a function key that is assigned to EXCMD without any command on the command line. This invokes CMS SUBSET. To return to VIEWSP, use the CMS command return.
- Press a function key that is assigned to EXCMD with a CMS command on the command line.

When the command completes, if the command on the command line is not an Operations Manager command, you return to the originating VIEWSP command.

When the command completes, if the command on the command line is an Operations Manager command, you return to the point that originated the VIEWSP command.

- Assign EXCMD with a z/VM or Operations Manager command to a PF key in VIEWSP PROFILE. For example, PF19=EXCMD LISTFILE \* \* A or PF20=EXCMD GOMCMD OPMGRM1 VIEWCON USER(OPERATOR).

When the command completes, if the command assigned to the PF key is not an Operations Manager command, you return to the originating VIEWSP command.

When the command completes, if the command assigned to the PF key is an Operations Manager command, you return to the point that originated the VIEWSP command.

7. The value for space utilization percentage is obtained by rounding downward. This value may differ from the space utilization percentage computed from other sources.

## Viewing spool file contents

You can use the following default PF keys to move through the displayed spool file contents:

- PF01 — Show the help screen
- PF03 — Return to the spool file list
- PF07 — Scroll up
- PF08 — Scroll down
- PF10 — Scroll left
- PF11 — Scroll right
- PF19 — Scroll to top

- PF20 — Scroll to bottom
- PF21 — Process CMS command
- PF24 — Display the PF key assignments
- ENTER — Process the command in the command field

Consider the following when using these PF keys:

- A command may be typed on the command line for the process CMS command action.
- When viewing a spool file, you can search through the display. Use the forward slash (/) to search forward and the back slash (\) to search backward. Use the equal sign (=) to repeat the previous command. Issue the DEFOPTN command to change these characters. These characters are only valid while viewing the contents of a file. They are not valid and will not work while viewing a list of spool files.
- When viewing the contents of a spool file, the VIEW, SORTA, and SORTD action keys have no effect. They are considered null PF keys.

### Setting values through the profile

- The actions for PF01-PF24, initial value for the sort order, and display of the PF key information lines are configurable through the VIEWSPL PROFILE file.
  1. Issue the following command in the VIEWSPL PROFILE file to customize the actions for PF01 - PF24:

*PFnn=action*

Where *nn* is 01 through 24 and *action* is a valid PF key action.

The valid PF key actions while viewing the spool file list are:

- HELP — Show the help screen
  - VIEW — View the spool file contents
  - END — End the VIEWSPL session
  - SORTA — Sort the files in ascending order
  - SORTD — Sort the files in descending order
  - UP — Scroll up
  - DOWN — Scroll down
  - LEFT — Scroll left
  - RIGHT — Scroll right
  - TOP — Scroll to top
  - BOT — Scroll to bottom
  - VIEWPF — Display the PF key assignments
  - EXCMD — Execute a z/VM or Operations Manager command
- Refer to usage note 10 for more information.

The valid PF key actions while viewing the spool file content are:

- HELP — Show the help screen
- END — Return to the spool file list
- UP — Scroll up
- DOWN — Scroll down
- LEFT — Scroll left
- RIGHT — Scroll right
- TOP — Scroll to top

- BOT — Scroll to bottom
  - VIEWPF — Display the PF key assignments
  - EXCMD — Execute a z/VM or Operations Manager command
- Refer to usage note 10 for more information.

2. Issue the SORT command in the VIEWSPL PROFILE file to customize the initial sort order of the spool files. Only one column can be specified. You can sort a column in ascending (A) or descending (D) order. If no sort order is provided, the default is ascending (A). Use the following format:

`SORT=column[,A|D]`

The columns you can sort are:

- OWNER
- FILE
- CLS
- QUE
- TYP
- SIZE
- HOLD
- DATE
- NAME
- TYPE
- DIST
- DEST
- COPY
- KEEP
- MSG
- USERFORM
- OPERFORM
- FCB

3. Issue the following command in the VIEWSPL PROFILE file to control the display of the PF key information lines:

`SHOWPF=c[,h,xat,clr]`

Where *c* is one of the following options:

- N — Do not display the PF key assignments on the screen.
- Y — Display PF01–PF12 assignments on the screen.
- A — Display PF01–PF24 assignments on the screen.

The values for *h,xat,clr* are optional. If one of the values is specified, they must all be specified.

- *h* — Specifies if the line is highlighted. Values are Y or N.
- *xat* — Specifies the extended attributes applied to the line. Values are:
  - AAI — Leave as is
  - ABL — Blinking
  - ARV — Reverse video
  - AUL — Underline
- *clr* — Specifies the color applied to the line. Values are:
  - CAI — Leave as is
  - CBL — Blue

- CRE — Red
- CPI — Pink
- CGR — Green
- CCY — Cyan
- CYE — Yellow
- CWH — White

The following shows an example of what might be in a VIEWSPL PROFILE file after customizing the initial sort order of the spool files and the display of the PF keys:

```
PF13=HELP
SORT=OWNER,A
SHOWPF=Y,Y,AAI,CAI
```

## Return codes

Table 109 lists the return codes for the VIEWSPL command.

*Table 109. VIEWSPL return codes*

| Return code | Description        |
|-------------|--------------------|
| 0           | Finished correctly |
| -100        | Error              |

---

## Appendix A. Sample files

Several sample files are shipped with Operations Manager. These samples help you create customized statements or EXEC files for your site. The files are located on the 5697J10E 2C2 minidisk or associated SFS directory.

### **OPMGRM1 CONFSAMP**

Contains sample configuration statements for defining the Operations Manager environment and who is authorized to use it. Refer to Chapter 2, “Configuring Operations Manager,” on page 5 for more information on how to use this sample to create the Operations Manager startup configuration file.

### **EXAMPLES CONFSAMP**

Contains sample Operations Manager configuration statements to do the following:

- Perform some initial monitoring of spool space, page space, console messages, minidisk full conditions, and logoff status of critical user IDs. Alerts can be sent via email or to a z/VM user ID that is used as a central operations console.
- Schedule spool cleanup.
- Schedule expiration processing in IBM Backup and Restore Manager for z/VM and IBM Archive Manager for z/VM.
- Request that all Operations Manager servers reaccess the OPMGRM1 198 disk where configuration information and, possibly, your REXX EXECs are stored.

To use the samples in EXAMPLES CONFSAMP, you must do one of the following:

- Copy the example(s) to your startup configuration file for Operations Manager, or
- Copy the file to the OPMGRM1 198 disk, rename the file, and add a CONFIG statement to the startup configuration file that references this additional configuration file.

In addition, the examples must be modified to work in your environment. It is required to update the email address and the user ID for the central operations console. Refer to the comments in EXAMPLES CONFSAMP for more details.

### **SMTMSG SAMPEXEC**

Contains a sample REXX EXEC for sending an email notification when an alert condition occurs. Refer to the comments in the sample EXEC for details.

To use this sample EXEC, copy the file to the OPMGRM1 198 disk (or other minidisk accessible to all Operations Manager service machines) and rename it to SMTMSG EXEC. Be sure to review this EXEC and modify it, if necessary, to work in your environment.

### **CHKDISK SAMPEXEC**

Contains a sample REXX EXEC for checking the percent full of a specified minidisk. If the percent full is above a specified threshold, the EXEC will call SMTMSG EXEC to send an email alert to the specified user. Refer to the comments in the sample EXEC for details.

To use this sample EXEC, copy the file to the OPMGRM1 198 disk (or other minidisk accessible to all Operations Manager service machines) and rename it to CHKDISK EXEC. Be sure to review this EXEC and modify it, if necessary, to work in your environment.

#### **EBC2ASC SAMPEXEC**

Contains a sample REXX EXEC for translating from EBCDIC to ASCII using the specified code pages. Refer to the comments in the sample EXEC for details.

To use this sample EXEC, copy the file to the OPMGRM1 198 disk (or other minidisk accessible to all Operations Manager service machines) and rename it to EBC2ASC EXEC. Be sure to review this EXEC and modify it, if necessary, to work in your environment.



---

## Appendix B. Accounting and Monitoring Information

This appendix describes the accounting and monitoring records generated by Operations Manager.

---

### Accounting record format

Operations Manager optionally generates accounting records that contain useful information about its processing. An accounting record is generated every hour and when you shutdown GOMMAIN. To retrieve this information, you can write a program to read the accounting data and generate any reports you may need. See *IBM z/VM CP Programming Service (SC24-6001)* for more information.

To generate accounting records, you must update the CP directory for user ID OPMGRM1, and any other user ID running GOMMAIN.

Each accounting record consists of 80 characters. Each column or group of columns in the record contains specific information about Operations Manager. Table 110 describes the format of each accounting record.

Table 110. Accounting record format

| Column | Contents                                                                                              |
|--------|-------------------------------------------------------------------------------------------------------|
| 01-08  | User ID. (Character)                                                                                  |
| 09-12  | GOM is displayed to indicate that the record is being generated by Operations Manager. (Character)    |
| 13-16  | The length of the Operations Manager data. (Binary)                                                   |
| 17-24  | The beginning of the accounting interval. (Clock format)                                              |
| 25-32  | The ending of the accounting interval. (Clock format)                                                 |
| 33-36  | The count of IUCV traffic. This includes all messages, messages between consoles, and so on. (Binary) |
| 37-40  | The number of rules triggered from pattern matches. (Binary)                                          |
| 41-44  | The number of monitored machines triggered. (Binary)                                                  |
| 45-48  | The number of schedules triggered. (Binary)                                                           |
| 49-52  | The number of spool usage monitors triggered. (Binary)                                                |
| 53-56  | The number of space page monitors triggered. (Binary)                                                 |
| 57-60  | The number of idle monitors triggered. (Binary)                                                       |
| 61-64  | The number of event monitors triggered. (Binary)                                                      |
| 65-68  | The number of actions executed. This includes actions waiting to be executed. (Binary)                |
| 69-76  | The amount of wall clock time consumed executing actions. (Clock format)                              |
| 77-78  | Unused. (Blank)                                                                                       |
| 79-79  | Accounting record identification code (C0).                                                           |
| 80-80  | Accounting record identification code (C0).                                                           |

#### Related reference:

“Updates to CP directory” on page 6

---

## Monitoring record format

Operations Manager optionally generates monitoring records that contain useful information about its processing. A monitoring record is generated every hour and when you shutdown GOMMAIN. To retrieve this information, you can write a program to read the accounting data and generate any reports you may need. See *IBM z/VM CP Programming Service (SC24-6001)* for more information.

To generate monitoring records, you must update the CP directory for user ID OPMGRM1, and any other user ID running GOMMAIN.

Monitoring records do not have a size limit. Each column or group of columns in the record contains specific information about Operations Manager. Table 111 describes the format of each monitoring record.

*Table 111. Monitoring record format*

| Column | Contents                                                                                              |
|--------|-------------------------------------------------------------------------------------------------------|
| 01-08  | User ID. (Character)                                                                                  |
| 09-12  | GOM is displayed to indicate that the record is being generated by Operations Manager. (Character)    |
| 13-16  | The length of the Operations Manager data. (Binary)                                                   |
| 17-24  | The beginning of the monitor interval. (Clock format)                                                 |
| 25-32  | The ending of the monitor interval. (Clock format)                                                    |
| 33-36  | The count of IUCV traffic. This includes all messages, messages between consoles, and so on. (Binary) |
| 37-40  | The number of rules triggered from pattern matches. (Binary)                                          |
| 41-44  | The number of monitored machines triggered. (Binary)                                                  |
| 45-48  | The number of schedules triggered. (Binary)                                                           |
| 49-52  | The number of spool usage monitors triggered. (Binary)                                                |
| 53-56  | The number of page space monitors triggered. (Binary)                                                 |
| 57-60  | The number of idle monitors triggered. (Binary)                                                       |
| 61-64  | The number of event monitors triggered. (Binary)                                                      |
| 65-68  | The number of actions executed. This includes actions waiting to be executed. (Binary)                |
| 69-76  | The amount of wall clock time consumed executing actions. (Clock format)                              |

**Related reference:**

“Updates to CP directory” on page 6

---

## Appendix C. Log File Management

Operations Manager writes all of its messages to a log. The log provides a chronological audit trail of Operations Manager processing. It records settings, inbound and outbound message traffic, messages from users who trigger rules, when schedules occur, machine monitors, any commands issued, and so on. The information in the log can help with problem identification and resolution.

A new log is created when Operations Manager is started or when a message is logged for a new date. Because a log is created for each new day, the log file name is MASALOG *mddyyyy*, where *mm* is the month, *dd* is the day, and *yyyy* is the year. If a log file for a specific day already exists when the log is started, the log output will be appended to that day's log file.

Operations Manager places the log files on DASD accessed as file mode D. By default this is the OPMGRM1 194 minidisk or the VMSYS:OPMGRM1.LOGS directory in SFS. If the D disk does not exist, no log files will be created. If the D disk location changes while Operations Manager is running, any new log files are written to the current D disk.

Before a new log file is created for a new date, Operations Manager runs GOMCLG EXEC to manage the space containing the log files. Before starting a new log file, GOMCLG removes the oldest log files on the D disk until the space of the D disk is less than 50%. If the log files are in a shared file system, Operations Manager cannot manage the log files because it cannot determine the amount of space within the shared file system to be allocated for log files. In this case, you can write your own space management routines to manage the log files within the shared space.

**Note:** It is recommended that you retain old log files for several days. You can do this by using the DEFSCHD command to automatically archive older log files in *IBM Archive Manager for z/VM* on a regular basis.

---

### Customize log file management

#### About this task

You can customize the log file management cleanup routine to suit your requirements. To customize the cleanup routine, you must copy and change the GOMCLG EXEC distributed with Operations Manager. The following steps provide a guideline to customize your cleanup routine:

1. Log on to the VM account your installation uses to modify Operations Manager configuration files. Do *not* use OPMGRM1 or GOMSVM 1–4.
2. Link to the minidisk that contains the original GOMCLG EXEC:  
CP LINK OPMGRM1 400 400 RR
3. Access the original minidisk:  
ACCESS 400 I
4. Link to the minidisk containing the configuration files:  
CP LINK OPMGRM1 198 198 MR
5. Access the configuration minidisk:  
ACCESS 198 E

6. Copy the original GOMCLG EXEC file:  
COPYFILE GOMCLG EXEC G = = E
7. Modify the log cleanup EXEC:  
XEDIT GOMCLG EXEC E

---

## Appendix D. Operations Manager and TCP/IP interaction

This appendix describes how to configure Operations Manager to interact with TCP/IP using the DEFTCPA configuration statement. This may be used to process syslog data, or file input from local or remote z/VM systems.

---

### Collection and processing of syslog data

Operations Manager may be configured to accept and process syslog data from any machine that supports forwarding syslog data. When Operations Manager receives syslog data, the data is processed in the same manner as SECUSER or OBSERVER data from a monitored user console; the data is filtered through the defined rules and triggers actions if any matching criteria is found. The collected syslog data may be viewed using the VIEWCON command.

To configure Linux, or other syslog capable machines, to forward syslog data to another system, please consult your Linux documentation. The data must be sent to the TCP/IP address of the system running Operations Manager.

To configure Operations Manager to process syslog data, issue the DEFTCPA command. The following command is an example:

```
DEFTCPA NAME(RMTSYLGD),TCPAPPL(GOMRSYL),+  
TCPUSER(TCPIP),TCPADDR(000.000.000.000),TCPPOINT(00514),+  
PARM(SYSLOG 03330417UTF8)
```

The command above starts the syslog processing program provided by Operations Manager (TCPAPPL(GOMRSYL)) and instructs the program to listen on UDP port 514 (TCPPOINT(00514)) of any available IP address (TCPADDR(000.000.000.000)) assigned to the TCP/IP service machine named TCPIP (TCPUSER(TCPIP)). The PARM field establishes the following information:

- the user ID the syslog data is collected under (SYSLOG)
- the ASCII to EBCDIC translation table used (03330417)
- the ASCII encoding scheme (UTF8)

#### Notes:

- The PARM value is column specific. The DEFTCPA command provides more information on the available PARM values.
- Depending on your TCP/IP configuration, you may need to authorize OPMGRM1 (and any user ID running GOMMAIN) to use the ports specified in your DEFTCPA statements. Refer to the PORT or OBEY statements in *z/VM TCP/IP Planning and Customization (SC24-6125)* for information on enabling an application to use specific ports.

To view the collected syslog data, issue the VIEWCON command:

```
GOMCMD OPMGRM1 VIEWCON USER(SYSLOG)
```

where OPMGRM1 is the user ID of the Operations Manager server running GOMMAIN.

#### Related reference:

“DEFTCPA” on page 53

“VIEWCON” on page 127

## Translation tables

The syslog data must be converted from ASCII to EBCDIC. Three sample ASCII to EBCDIC translation tables are provided on the sample installation disk and may be copied to the OPMGRM1 198 minidisk or any other disk accessed by user ID OPMGRM1, or any other user ID running GOMMAIN. Operations Manager allows the use of any input and output code page.

Table 112 lists the sample tables included in the Operations Manager samples disk 5697J10E 2C2.

*Table 112. Sample translation table files*

| Table file name   | Description                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
| 03330025 SAMPTABL | Translation from ASCII code page 00819 to EBCDIC code page 00037                                                |
| 03330417 SAMPTABL | Translation from ASCII code page 00819 to EBCDIC code page 01047                                                |
| 0333047C SAMPTABL | Translation from ASCII code page 00819 to EBCDIC code page 01148                                                |
| 00000000 SAMPTABL | No translation is performed. This sample table can be used when data is expected to be in ECBDIC when received. |

IBM also provides many translation tables at <http://www.ibm.com/software/globalization/cdra/index.jsp>. Refer to Appendix J for a link to downloadable translation tables. When the download is complete, the file must have fixed records and a blocksize of 256 for Operations Manager to accept the translation table.

The syslog data must be encoded in ISO8 or UTF8 format. If the syslog data is encoded in UTF8, the input is limited to plane0 and plane1 of Unicode.

### Copying the sample Operations Manager tables

To copy the sample translation tables provided with Operations Manager, follow the steps below.

1. Log on to the VM account your installation uses to modify Operations Manager configuration files. Do *not* use OPMGRM1 or OPMGRS1–4.
2. Link to the minidisk that contains the samples:  
CP LINK 5697J10E 2C2 2C2 RR
3. Access the samples minidisk:  
ACCESS 2C2 I
4. Link to the minidisk that contains the configuration files:  
CP LINK OPMGRM1 198 198 MR
5. Access the configuration minidisk:  
ACCESS 198 E
6. Copy the translation tables:  
COPYFILE \* SAMPTABL I = TABLE E

---

## Collection and processing of remote file input

Operations Manager may be configured to accept and process the content of a file sent from a CMS guest on the local or a remote z/VM system. When Operations Manager receives the file content, the data is processed in the same manner as SECUSER or OBSERVER data from a monitored user console; the file content is filtered through the defined rules and triggers actions if any matching criteria is found. The collected file content may be viewed using the **VIEWCON** command.

To configure Operations Manager to process file content data, issue the **DEFTCPA** command. The following command is an example:

```
DEFTCPA NAME(RMTFILES),TCPAPPL(GOMRSIF),+
TCPUSER(TCPIP),TCPADDR(192.168.100.200),TCPPOINT(63000)
```

The command above starts the file content processing program provided by Operations Manager (TCPAPPL(GOMRSIF)) and instructs the program to listen on TCP port 63000 (TCPPOINT(63000)) of IP address 192.168.100.200 (TCPADDR(192.168.100.200)) using the TCP/IP service machine named TCPIP (TCPUSER(TCPIP)).

### Notes:

1. The TCP/IP address specified on the TCPADDR operand of the **DEFTCPA** command must match the address specified on the **GOMRSIF** command, or the TCP/IP address specified in the **DEFTCPA** command must be 000.000.000.000.
2. The port specified on the TCPPOINT operand of the **DEFTCPA** command must match the port specified on the **GOMRSIF** command.
3. The port specified on the TCPPOINT operand of the **DEFTCPA** command must be enabled in TCP/IP for the TCP protocol. The **GOMRSIF** command only supports the TCP protocol.

### Related reference:

“DEFTCPA” on page 53

“VIEWCON” on page 127

## Send and view file content

### About this task

To send file content from a CMS guest, issue the GOMRSIF EXEC:

```
GOMRSIF TESTCONS DATA A 192.168.100.7 63000 TERRY
```

The command above transfers the content of TESTCONS DATA A to Operations Manager at the TCP/IP address 192.168.100.7 on port 63000 and the data is collected under user name TERRY.

To view the collected file content, issue the VIEWCON command:

```
GOMCMD OPMGRM1 VIEWCON USER(TERRY)
```

Where OPMGRM1 is the user ID of the Operations Manager server running GOMMAIN.

## Send and view multiple files

### About this task

To send multiple files to Operations Manager and view them all in one VIEWCON session, follow the steps below.

1. Issue a DEFTCPA statement in Operations Manager. For example:  

```
DEFTCPA NAME(RMTFILES),TCPAPPL(GOMRSIF),TCPUSER(TCPIP),+
TCPADDR(000.000.000.000),TCPPOINT(63000)
```
2. From a VM user ID that has access to the files, issue the necessary GOMRSIF commands. The port specified in the DEFTCPA statement must match the port specified in the GOMRSIF command. The following example sends the content of TEST DATA A and SAMPLE DATA A to Operations Manager and specifies that the data is collected under the user name TERRY:  

```
GOMRSIF TEST DATA A 192.168.100.200 63000 TERRY
GOMRSIF SAMPLE DATA A 192.168.100.200 63000 TERRY
```
3. This allows you to view all of the remote file input in one VIEWCON session. For this example, the following command is issued to display the data:  

```
GOMCMD OPMGRM1 VIEWCON USER(TERRY)
```

To send multiple files to Operations Manager and view them in separate VIEWCON sessions, follow the steps below.

**Note:** Only one DEFTCPA statement is required.

1. Issue a DEFTCPA statement in Operations Manager. For example:  

```
DEFTCPA NAME(RMTFILES),TCPAPPL(GOMRSIF),TCPUSER(TCPIP),+
TCPADDR(000.000.000.000),TCPPOINT(63000)
```
2. From a VM user ID that has access to the files, issue the necessary GOMRSIF commands, specifying different user names. The following example sends the content of TEST DATA A to Operations Manager and specifies that the data is collected under the user name TERRY, then sends the content of SAMPLE DATA A and specifies that the data is collected under the user name JOE.  

```
GOMRSIF TEST DATA A 192.168.100.200 63000 TERRY
GOMRSIF SAMPLE DATA A 192.168.100.200 63000 JOE
```
3. This allows you to view the remote file input in separate VIEWCON sessions. For this example, the following commands are issued to display the data:  

```
GOMCMD OPMGRM1 VIEWCON USER(TERRY)
GOMCMD OPMGRM1 VIEWCON USER(JOE)
```

The following section describes the GOMRSIF EXEC in more detail.

---

## GOMRSIF

►►—GOMRSIF—filename—filetype—filemode—ipaddr—ipport—username—————►►

*Figure 63. GOMRSIF command syntax*

## Authorization

No specific Operations Manager authority is required for the GOMRSIF command.

The GOMRSIF command may be run on any system where the EXEC is installed; Operations Manager does not need to be installed to issue this command.



## Purpose

Issue the GOMRSIF command to send a file to Operations Manager and collect the data under a specified user name.

## Operands

**filename**

specifies the file name.

**filetype**

specifies the file type.

**filemode**

specifies the file mode.

**ipaddr** specifies the target VM TCP/IP address to send the data. Format is xxx.xxx.xxx.xxx. Leading zeros are not required. This must be the TCP/IP address of the system running Operations Manager.

**ipport** specifies the target VM TCP/IP port to send the data. Format is nnnnn. Leading zeros are not required. This must match the port number specified on the corresponding DEFTCPA statement.

**username**

specifies the user name the data is collected under.

## Usage notes

None.

## Return codes

Table 113 lists the return codes for the GOMRSIF command.

*Table 113. GOMRSIF return codes*

| Return code | Description                                  |
|-------------|----------------------------------------------|
| 0           | Successful                                   |
| 4           | No data in file                              |
| 8           | Error reading file                           |
| 12          | File not found or TCP/IP error               |
| 16          | Command parameter missing or parameter error |



---

## Appendix E. Testing Operations Manager installation and configuration

This appendix describes the steps required to test various components of Operations Manager's installation and configuration.

If any of the tests described in this appendix fail you must update the Operations Manager configuration file.

**Related tasks:**

"Managing the configuration file" on page 28

---

### Checking the status of OPMGRM1

#### About this task

To verify that the Operations Manager server (OPMGRM1) has been successfully started and is able to process requests, follow the steps below.

#### Procedure

1. Log on to an authorized user ID. This user ID must be authorized to issue the Operations Manager STATUS command. This can be done by giving the user configuration authority, control authority, or authorizing the user to the STATUS command.
2. Issue the following command to instruct Operations Manager to output status information:  
`GOMCMD OPMGRM1 STATUS`
3. Several messages from OPMGRM1 appear indicating the server's status.

---

### Verifying consoles are monitored and viewable

#### About this task

To verify that the console of a user ID is being monitored by Operations Manager and that authorized users can view and interact with the console follow the steps below.

#### Procedure

1. Log on to an authorized user ID. This user ID must be authorized to view a monitored console using the VIEWCON command.
2. Issue the following command to view another user's console:  
`GOMCMD OPMGRM1 VIEWCON USER(userid)`

Where *userid* is a monitored console that has or is generating console messages. *userid* must have SECUSER or OBSERVER set to OPMGRM1. In addition, if you are using SECUSER, *userid* must be disconnected.

3. A full-screen view of the console of *userid* appears. Use PF07 and PF08 to scroll up and down in the console, PF10 and PF11 to scroll left and right, and PA02 to toggle autoscrolling.

**Note:** Any commands entered on the VIEWCON command line are issued on *userid*.

4. Press PF03 to exit the VIEWCON session.
5. Issue the following command to view the Operations Manager log:  
GOMCMD OPMGRM1 VIEWLOG
6. A full-screen view of the console of the Operations Manager server (OPMGRM1) appears. This is also known as the Operations Manager log. The log includes all consoles being monitored by Operations Manager as well as Operations Manager messages concerning rules, schedules, monitors, and actions. You can also use the PF keys to scroll up, down, left, right, and toggle autoscrolling.
7. Press PF03 to exit the VIEWLOG session.

**Related tasks:**

“Step 4. Update IDs monitored by Operations Manager” on page 9

**Related reference:**

“VIEWCON” on page 127

“VIEWLOG” on page 131

---

## Creating a schedule and corresponding action

### About this task

To create a schedule and corresponding action to send a message to your user ID five minutes from now, follow the steps below. In this example, you are defining the schedule and action dynamically. Typically, schedules and actions are defined in the configuration file that is loaded when Operations Manager starts.

### Procedure

1. Log on to an authorized user ID. This user ID must be authorized to define schedules and define actions. This can be done by giving the user ID configuration authority or authorization to the DEFSCHD and DEFACTN commands.
2. To define an action to send you a message, issue the following command:  
GOMCMD OPMGRM1 DEFACTN NAME(TSTACTN1),  
COMMAND(CP MSG *userid* Test Operations Mgr),ENV(LVM)  
Replace *userid* with your current user ID.
3. To define a schedule that triggers in five minutes from now, issue the following command :  
GOMCMD OPMGRM1 DEFSCHD NAME(TESTSCHD),ACTION(TSTACTN1),WHEN(NOW+00:05)
4. In five minutes the following message from OPMGRM1 appears:  
Test Operations Mgr

---

## Verifying spool files are viewable

### About this task

To verify that authorized users can view and purge spool files, follow the steps below.

## Procedure

1. Log on to an authorized user ID. This user ID must be authorized to view spool files using the VIEWSPPL command.
2. Issue the following command to view the list of spool files on the system:  
GOMCMD OPMGRM1 VIEWSPPL
3. A full-screen view of the spool files on the system appears. Use PF07 and PF08 to scroll up and down in the list. Use PF10 and PF11 to scroll left and right.
  - To sort the list, place the cursor on a column heading and use PF05 or PF06 to sort ascending or descending.
  - To view the contents of a file, place the cursor on the correct row, and use PF02. When viewing the contents of a file, use PF03 to exit the file and return to the list view. Files noted with an asterisk (\*) are still open and not viewable.
  - To purge a file, place the cursor in the command area next to the file and type PURGE. Files noted with an asterisk (\*) are still open and can not be purged.
4. Press PF03 to exit the VIEWSPPL session.



---

## Appendix F. Application and settings interference

### Applications

Certain applications and settings can interfere with Operations Manager. The use of any application or setting that interferes with Operations Manager is strongly discouraged in any virtual machine while any Operations Manager component is active.

Applications that do any of the following can interfere with Operations Manager:

- Manipulate the system clock comparator directly. The applications should use STIMER or STIMERM macros to manipulate the system clock comparator and system clock comparator interrupts.
- Utilize the \*MSG or \*MSGALL CP IUCV system services. CP system services restrict the number of IUCV connections to these services to one connection. Operations Manager requires the connection to function correctly.
- Modify the settings for CPCONIO, VMCONIO, MSG, IMSG, WNG, EMSG, or SMSG. Operations Manager establishes the required settings at initialization, and does not expect the settings to changes.

The WAKEUP program is known to violate one or more of the above rules and interfere with Operations Manager.

The following user IDs run Operation Manager components and should not run an application that will interfere with it:

- OPMGRM1 and any other user ID running GOMMAIN
- Any action processing server (OPMGRS1 through OPMGRS4) and any other user ID running GOMSVM

### Settings

Operations Manager utilizes the \*MSG system service. The output messages generated from action processing are captured by the \*MSG system service.

The following settings are known to interfere with Operations Manager's ability to capture the output messages generated from action processing using the \*MSG system service.

- Setting OBSERVER to any value other than OFF
- Setting SECUSER to any value other than OFF

The following user IDs run Operation Manager components and should not use settings that will interfere with it:

- OPMGRM1 and any other user ID running GOMMAIN
- Any action processing server (OPMGRS1 through OPMGRS4) and any other user ID running GOMSVM





---

## Appendix G. GOMGLBL facility

GOMGLBL is an optional facility intended to help share information between REXX EXECs that run as Operations Manager actions that execute on different Operations Manager servers. The servers can include the main Operations Manager server (OPMGRM1 by default) and any of the action processing servers (OPMGRS*n* by default).

Although GOMGLBL is intended to be used with Operations Manager, it is not required to run on or use Operations Manager and does not need to be configured unless you intend to use it. GOMGLBL is somewhat similar to the CMS GLOBALV facility, which allows you to define names, assign values to those names, and retrieve the values at a later time. Whereas CMS GLOBALV is restricted to use within a single CMS guest, GOMGLBL provides a similar function across several CMS guests.

GOMGLBL may not be suitable for every environment or you may prefer other techniques for sharing data between Operations Manager servers (such as access to a Shared File System file).

GOMGLBL requires a writable shared segment, named GOMSEG, which all of the GOM workers sharing data must be able to access. The size of the segment depends on how much data you want to share; each megabyte in the shared segment can contain more than 2000 name/value pairs.

Similar to CMS GLOBALV, the names and the values assigned to them can be up to 255 characters in length. The names must be valid as REXX variable names (no embedded spaces, and so on), but the value can contain any character.

Any user ID that will be issuing the GOMGLBL command must have authority to issue a CP diagnostic code `DIAG x'0064'`. By default, this authority is provided in the NAMESEG GOMSEG option in the user's CP directory entry.

---

### Preparing GOMGLBL

The GOMSEGDF EXEC is provided to help in defining the GOMSEG writable shared segment. As provided, GOMSEGDF will define a 2MB segment at location 0x6000000 (96MB), which will provide room for 4095 shared variables. The location of the shared segment must be beyond the defined storage of any Operations Manager server (or other CMS guest) that will use it.

#### About this task

As shipped, the GOMSEGDF EXEC defines GOMSEG as shared, writable, and restricted (RSTD), so that only guests which have a NAMESAVE statement for GOMSEG in their directory entry can use it.

#### Procedure

1. Customize GOMSEGDF EXEC with the appropriate values for your site:

**Note:** If you intend to skip a large amount between the end of the defined storage for the Operations Manager servers and the start of the shared

segment, review the usage notes about the effects on real storage in the DEFSEG section in the *CP Commands and Utilities Reference* for your level of z/VM.

If you require more than 4095 shared variables, each additional megabyte provides for 2048 more. For example, available shared variables = (shared segment size in MB\*2048) -1.

a. From user ID 5697J10E, type:

```
'ACC 400 M'
'XEDIT GOMSEGDF EXEC M'
```

b. Find the following statement:

```
Address 'COMMAND' 'CP DEFSEG GOMSEG 6000-61FF SW RSTD'
```

Change any of the following items as needed:

- Change the address range of the segment location if you want the segment larger than 2MB or located someplace other than 96MB.
- Remove the RSTD operand if you do not want the segment to be restricted.

c. File your changes.

2. Run GOMSEGDF EXEC from a user ID with the necessary privilege class to issue the CP DEFSEG command. By default, this is class E. For example, issue the following command:

```
GOMSEGDF DEFINESEG
```

For more information on privilege classes, refer to Chapter 2, “Configuring Operations Manager,” on page 5.

3. Run the GOMSEGIN program (GOMSEGIN MODULE) to initialize the segment.
4. Ensure that all user IDs that will use GOMGLBL have the NAMESAVE GOMSEG statement in their directory entry. GOMGLBL is now ready to use.
5. Log off and restart any user IDs that were updated in step 4.

---

## GOMGLBL

```
▶▶ GOMGLBL FROM REXXvarname [TRUNCate] INTO REXXvarname VAR REXXvarname ▶▶
▶ VALUE newvalue NAME sharedname QUIET ▶▶
```

Figure 64. GOMGLBL command syntax

### Operands

#### FROM REXXvarname

Sets the value of the shared variable to the contents of the REXX variable REXXvarname. REXX must be active.

#### TRUNCate

If the value obtained for the FROM operand exceeds 255 bytes, truncate it to 255 bytes and perform the set operation anyway.

#### INTO REXvarname

Place the value of the shared variable into the REXX variable REXXvarname. REXX must be active.

**VAR** *REXXvarname*

The name of the shared variable is to be obtained from the REXX variable *REXXvarname*. REXX must be active.

**NAME** *sharedvarname*

*sharedvarname* is the name of the shared variable whose value is to be set and/or retrieved.

**QUIET** Does not write error messages to the console.

## Usage notes

1. Shared variable names and values are limited to 255 bytes.
2. Variables may be retrieved and set in the same operations. For example, the old value, if any, will be retrieved, and then the new value set.
3. If you attempt to retrieve a shared variable which has never been set, a null value will be returned.
4. If a REXX variable is specified on the FROM operand whose contents exceed 255 bytes in length, the operation will terminate with an error message unless the TRUNCate operand is also specified.
5. The VALUE operand may be used to provide data to be stored into a shared variable, but data can only be retrieved into a REXX variable. Data specified on the VALUE operand cannot contain embedded blanks. To store data containing embedded blanks, the FROM operand must be set.
6. The values of shared variables are not saved across IPLs of the z/VM system, or when all machines using the DCSS log off.
7. You must not use stem variables whose names contain lower-case characters with GOMGLBL.

## Restrictions

Only user IDs permitted to access the GOMSEG DCSS will be able to use GOMGLBL.

## Return codes

Table 114 lists the return codes for the GOMGLBL command.

Table 114. GOMGLBL return codes

| Return code | Description                                                                                     |
|-------------|-------------------------------------------------------------------------------------------------|
| 8           | No parameters supplied.                                                                         |
| 12          | A required operand is missing.                                                                  |
|             | A required parameter is missing.                                                                |
|             | Conflicting options selected.                                                                   |
|             | Duplicate options entered.                                                                      |
|             | Operand exceeds maximum length.                                                                 |
|             | A specified REXX variable contains no data. The REXX variable specified for VAR is set to null. |

Table 114. GOMGLBL return codes (continued)

| Return code | Description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16          | GOMGLBL requires the extended parameter list. GOMGLBL can only be called from environments, such as CMS or REXX, that support the CMS extended parameter list.                                                                                                                                                                                                                                                                         |
|             | CSL services are not available.                                                                                                                                                                                                                                                                                                                                                                                                        |
|             | VARNAME, FROM, and INTO are specified but not running under REXX. The REXX environment must be active for GOMGLBL to access REXX variables.                                                                                                                                                                                                                                                                                            |
|             | GOMSEG DCSS is not initialized or is at the wrong level.                                                                                                                                                                                                                                                                                                                                                                               |
|             | The user is not authorized to access the GOMSEG DCSS. The GOMSEG DCSS by default is restricted; users must have a NAMESAVE GOMSEG statement in their directory entry.                                                                                                                                                                                                                                                                  |
|             | DCSS GOMSEG is not defined.                                                                                                                                                                                                                                                                                                                                                                                                            |
|             | CC=3 from diag 0x64 (FINDSEG).                                                                                                                                                                                                                                                                                                                                                                                                         |
|             | Error code <rc> (hex) from FINDSEG.                                                                                                                                                                                                                                                                                                                                                                                                    |
|             | Error code <rc> (hex) from LOADSHR.                                                                                                                                                                                                                                                                                                                                                                                                    |
| 104         | Cannot retrieve value-spin.<br><br>The value of the variable specified could not be retrieved, probably due to concurrent activity from other users. GOMGLBL retries the operation several times before issuing this message, but you may also reissue the command if desired.                                                                                                                                                         |
| 108         | Error code <rc> (hex) from DMSCSR.<br><br>The value of the variable specified was retrieved, but could not be stored into a REXX variable for the caller. However, if a new value was specified, it was successfully stored.                                                                                                                                                                                                           |
| 112         | Cannot set new value-spin.<br><br>The value of the variable specified could not be set, probably due to concurrent activity from other users. GOMGLBL retries the operation several times before issuing this message, but you may also reissue the command if desired.                                                                                                                                                                |
| 116         | Indicates that while the specified variable exists, and was retrieved, it could not be placed in a REXX variable for the caller, nor could the new value be stored. One or both of the messages associated with return codes 108 and 112 may be issued.                                                                                                                                                                                |
| 120         | Out of room in the GOMSEG saved segment.                                                                                                                                                                                                                                                                                                                                                                                               |
| 124         | Data exceeds 255 bytes and TRUNC was not specified.<br><br>The FROM operand was used to direct GOMGLBL to obtain the value to which to set a shared variable, but the contents of that REXX variable were more than 255 bytes in length, and the TRUNC operand was not specified. The maximum length of the value of a shared variable is 255 bytes. If you want GOMGLBL to truncate strings when necessary, use the TRUNCate operand. |
| 128         | VAR variable exceeds 255 bytes.<br><br>The VAR operand was used to direct GOMGLBL to obtain the name of the shared variable to set and/or retrieve from a REXX variable, but the contents of that REXX variable were more than 255 bytes in length. The maximum length of a shared variable name is 255 bytes.                                                                                                                         |

---

## Appendix H. Messages and Codes

This appendix lists the messages generated by Operations Manager. A description of each message and suggestions on how you can resolve the problem are provided, where possible.

Many of the messages issued by Operations Manager include a return code. For a description of these return codes, refer to *z/VM CP Programming Services* or *z/VM CMS Macros and Functions Reference*.

All messages generated by Operations Manager have a severity code as the last character of the message ID. Severity codes are described in Table 115.

Table 115. Severity codes

| Code | Description                                                                                                              |
|------|--------------------------------------------------------------------------------------------------------------------------|
| E    | Error. Some errors might be user-correctable, read the User Response text to determine the appropriate course of action. |
| I    | Information only. No user action is required.                                                                            |
| L    | Log message.                                                                                                             |
| S    | Severe message.                                                                                                          |
| W    | Warning message. Results might not be as expected.                                                                       |

---

**GOM0001I** STARTED V1R5 BUILT *yy/mm/dd hh:mm*

**Explanation:** An Operations Manager component has started.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0002E** STIMERM SET FAILED *nn*

**Explanation:** A call to the STIMERM system function failed with code *nn*.

**System action:** Operations Manager terminates.

**User response:** See STIMERM in the *z/OS MVS Assm Services Reference* for a list of code descriptions.

---

**GOM0003E** REQUIRED PARAMETERS NOT PRESENT

**Explanation:** Required parameters for startup were not present.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

**GOM0004I** EP= *xxxxxxx*

**Explanation:** Records the entry point of startup, where *xxxxxxx* is the actual entry point.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0005E** HNDEXT SET FAILED *nn*

**Explanation:** A call to the HNDEXT system function failed with code *nn*.

**System action:** Operations Manager terminates.

**User response:** See HNDEXT in the *z/VM CMS Macros and Functions Reference* for a list of code descriptions.

---

**GOM0006W** ACCOUNTING INFORMATION NOT GENERATED *nn*

**Explanation:** An accounting information record was unable to be generated via diagnose 4C. The condition code from the operation is indicated in *nn*.

**System action:** The program continues.

**User response:** See *z/VM CP Programming Services* for an explanation for *nn*.

---

**GOM0007W** MONITOR SAMPLE INTERVAL APPLDATA NOT GENERATED *cc rc*

**Explanation:** An attempt was made to collect sample monitoring records.

**System action:** The program continues.

**User response:** Add OPTION APPLMON to the CP directory so Operations Manager can collect MONITOR SAMPLE INTERVAL records. See *z/VM CP Programming Services* DIAGNOSE x'DC' for an explanation for the condition code *cc* and the return code *rc*.

---

**GOM0008E CONNECTION TO *userid* NOT ACCEPTED**

**Explanation:** The VM *userid* did not accept the connection request.

**System action:** The Operations Manager command terminates.

**User response:** A connection request was not accepted due to a version/service level mismatch between the client and server. Ensure the version/service level of GOMMAIN, GOMCMD, and GOMSVM are the same.

---

**GOM0009E SUBCOM PROCESSING LIMITED TO *userid***

**Explanation:** When GOMCMD is processed from an ACTION, the *userid* is restricted to the *userid* running GOMMAIN which initiated the ACTION.

**System action:** The program continues.

**User response:** Ensure that the *userid* specified on the GOMCMD command is correct.

---

**GOM0010E GOM ENVIRONMENT ALREADY ACTIVATED**

**Explanation:** When an Operations Manager command is started it attempts to establish a GOM environment. The Operations Manager command detected the GOM environment is currently active from another Operations Manager command.

**System action:** The Operations Manager command terminates.

**User response:** Issue the Operations Manager command on another user ID or terminate the previous Operations Manager command.

---

**GOM0011E GOM ENVIRONMENT NOT ACTIVATED**

**Explanation:** When an Operations Manager command is started it attempts to establish a GOM environment. The Operations Manager command was unable to establish the GOM environment.

**System action:** The Operations Manager command terminates.

**User response:** Determine why the GOM environment cannot be activated.

---

**GOM0012E EXTENDED PARAMETER LIST NOT AVAILABLE**

**Explanation:** When an Operations Manager command is started, it attempts to obtain the extended parameter list. The Operations Manager command was unable to obtain the extended parameter list.

**System action:** The Operations Manager command terminates.

**User response:** Determine why the Operations Manager command is unable to obtain an extended parameter list.

---

**GOM0099S UNRECOVERABLE ERROR ENCOUNTERED**

**PSW=xxxxxxxxxxxxxxxx EP=xxxxxx  
MCB=xxxxxx**

**Explanation:** An unrecoverable error condition was encountered.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

**GOM0100I ENDED**

**Explanation:** An Operations Manager component has ended.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

**GOM0101E ABNEXIT SET FAILED, RC=*nn***

**Explanation:** A call to the ABNEXIT system function failed with code *nn*.

**System action:** Operations Manager terminates.

**User response:** See ABNEXIT in the z/VM documentation for a list of return code descriptions.

---

**GOM0102E ABNEXIT CLR FAILED, RC=*nn***

**Explanation:** A call to the ABNEXIT system function failed with code *nn*.

**System action:** Operations Manager terminates.

**User response:** See ABNEXIT in the z/VM documentation for a list of return code descriptions.

---

**GOM0111L COMMAND *cmd* RC=*rc* (*cmd response*)**

**Explanation:** The CP command *cmd* returned a non-zero return code. The return code and command output are listed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0112L** COMMAND *cmd* OUTPUT TRUNCATED, AVAIL=*avail* USED=*used* TRUNC=*trunc*

**Explanation:** The CP command *cmd* response was truncated. The number of bytes available, used, and truncated are listed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0121I** MCB IS LOCATED AT *xxxx*

**Explanation:** Records the address, represented by *xxxx*, of the MCB for this execution of Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0122I** ASSIZE=*nn* ASIT=*xxxxxxxx*

**Explanation:** Records the address space size in pages, represented by *nn*, and the address space token, represented by *xxxxxxxx*, for this execution of Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0123I** PRIVILEGES:

**Explanation:** This message displays in combination with message 0124I. The result is a message that records the CP PRIV classes available for this execution of Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0124I** (list of privileges)

**Explanation:** This message displays in combination with 0123I. The result is a message that records the CP PRIV classes available for this execution of Operations Manager.

**System action:** The program continues.

**User response:** Verify that all installation or maintenance procedures completed successfully.

---

**GOM0125I** SETTINGS:

**Explanation:** This message displays in combination with message 0126I. The result is a message that records the CP SETTINGS in effect at the beginning of this execution of Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0126I** (list of settings)

**Explanation:** This message displays in combination with 0125I. The result is a message that records the CP SETTINGS in effect at the beginning of this execution of Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0127E** SET PHASE 1 FAILED: error-message

**Explanation:** This message displays in combination with 0128E. The required settings for CP MSG processing failed.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

**GOM0128E** SET PHASE 2 FAILED: error-message

**Explanation:** This message displays in combination with 0127E. The required settings for CP MSG processing failed.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

**GOM0129I** STARTUP COMMAND *cmd*

**Explanation:** Records the startup command and parameters entered to begin this execution of Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0130W** DIAGNOSE CODE *hhhh* NOT AVAILABLE

**Explanation:** The privileged diagnose code *hhhh* is not available to Operations Manager. You may be unable to use one or more features of Operations Manager.

**System action:** The program continues. One or more features of Operations Manager are unavailable.

**User response:** Change the privilege class to allow the diagnose code listed if the particular feature of Operations Manager is required. See "Step 1. Update OPMGRM1" on page 5 for a list of privileged diagnose codes used by Operations Manager.

---

**GOM0161E** HNDIUCV SET FAILED *nn*

**Explanation:** A call to HNDIUCV failed with code *nn*.

**System action:** Operations Manager terminates.

**User response:** See HNDIUCV in the z/VM documentation for a list of code descriptions.

---

---

**GOM0162E HNDIUCV CLR FAILED *nn*****Explanation:** A call to HNDIUCV failed with code *nn*.**System action:** Operations Manager terminates.**User response:** See HNDIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0163E IUCV CONNECT *userid* FAILED *nn*****Explanation:** A call to CMSIUCV to connect to *userid* failed with code *nn*.**System action:** Operations Manager terminates.**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0164E IUCV SEVER *userid* FAILED *nn*****Explanation:** A call to CMSIUCV to sever the connection to *userid* failed with code *nn*.**System action:** The program continues.**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0165E IUCV SEVER *userid* FAILED, NAME NOT FOUND****Explanation:** An internal lookup for *userid* failed. This is an internal logic error.**System action:** The program continues.**User response:** Contact IBM Technical Support for assistance.

---

**GOM0166E EXTERNAL INTERRUPT TYPE *nn* IS NOT VALID****Explanation:** An external interrupt with an incorrect type *nn* has been presented for processing.**System action:** The program continues.**User response:** No action is required.

---

**GOM0167E IUCV ACCEPT *userid* FAILED *nn*****Explanation:** A call to CMSIUCV to accept a connect from *userid* failed with code *nn*.**System action:** The program continues.**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0168E CONNECTION COMPLETE FOR PATHID *pppp* FAILED, PATH NOT FOUND****Explanation:** An internal lookup for IUCV PATHID,represented by *pppp*, failed. This is an internal logic error.**System action:** The program continues.**User response:** Contact IBM Technical Support for assistance.

---

**GOM0169E IUCV SEVER PATHID *pppp* FAILED *nn*****Explanation:** A call to CMSIUCV to sever the path, which is represented by *pppp*, failed with code *nn*.**System action:** The program continues.**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0170E IUCV SEVER PATHID *pppp* FAILED, PATH NOT FOUND****Explanation:** An internal lookup for the IUCV PATHID, which is represented by *pppp*, failed. This is an internal logic error.**System action:** The program continues.**User response:** Contact IBM Technical Support for assistance.

---

**GOM0171E IUCV RECEIVE FOR PATHID *pppp* FAILED *nn*****Explanation:** A call to IUCV to receive a message from the path represented by *pppp* failed with code *nn*.**System action:** The program continues.**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0172E IUCV RECEIVE FOR PATHID *pppp* FAILED, PATH NOT FOUND****Explanation:** An internal lookup for IUCV PATHID, which is represented by *pppp*, failed. This is an internal logic error.**System action:** The program continues.**User response:** Contact IBM Technical Support for assistance.

---

**GOM0173E IUCV REPLY FOR PATHID *pppp* FAILED *nn*****Explanation:** A call to IUCV to send a reply on path *pppp* failed with code *nn*.**System action:** The program continues.**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.



---

**GOM0174E IUCV REPLY FOR *userid* FAILED, PATH NOT FOUND**

**Explanation:** An internal lookup for *userid* failed. This resulted in an internal logic error.

**System action:** The program continues.

**User response:** Contact IBM Technical Support for assistance.

---

**GOM0175E IUCV SEND FOR PATHID *pppp* FAILED *nn***

**Explanation:** A call to IUCV to send a message on path *pppp* failed with code *nn*.

**System action:** The program continues.

**User response:** See CMSIUCV in z/VM documentation call for return codes.

---

**GOM0176E IUCV SEND FOR *userid* FAILED, PATH NOT FOUND**

**Explanation:** An internal lookup for *userid* failed. This resulted in an internal logic error.

**System action:** The program continues.

**User response:** Contact IBM Technical Support for assistance.

---

**GOM0177E IUCV SEND COMPLETE FOR PATHID *pppp* FAILED, PATH NOT FOUND**

**Explanation:** An internal lookup for the IUCV PATHID, represented by *pppp*, failed. This is an internal logic error.

**System action:** The program continues.

**User response:** Contact IBM Technical Support for assistance.

---

**GOM0178E IUCV SEND COMPLETE FOR PATHID *pppp* MSGID *nnnnnnnnnn* FAILED, MSG NOT FOUND**

**Explanation:** An internal lookup for a message, represented by *nnnnnnnnnn*, failed. This is an internal logic error.

**System action:** The program continues.

**User response:** Contact IBM Technical Support for assistance.

---

**GOM0179E IUCV SEND COMPLETE FOR PATHID *pppp* FAILED *xxxxxxx***

**Explanation:** The completion of an IUCV message send command failed on the PATHID *pppp* with notification code *xxxxxxx*.

**System action:** The program continues.

---

**User response:** See CMSIUCV in the z/VM documentation for a list of code descriptions.

---

**GOM0180E IUCV CONNECT FOR *userid* FAILED, ALREADY ASSIGNED PATHID *pppp***

**Explanation:** An attempt to establish multiple IUCV connections to the same machine has occurred. This resulted in an internal logic error.

**System action:** The program continues.

**User response:** Contact IBM Technical Support for assistance.

---

**GOM0181E IUCV REPLY FOR *userid* FAILED, NO REPLY EXPECTED**

**Explanation:** An attempt was made to send a reply to an IUCV message that did not expect a reply. This is an internal logic error.

**System action:** The program continues.

**User response:** Contact IBM Technical Support for assistance.

---

**GOM0182E VERSION/SERVICE LEVEL MISMATCH ON IUCV CONNECT ATTEMPT BY *userid***

**Explanation:** An attempt to connect to Operations Manager using IUCV from a previous or later version/service level was detected.

**System action:** The IUCV connection is terminated and the program continues.

**User response:** Use only GOM commands from a matching version/service level.

---

**GOM0183E IUCV CONNECTION TO SYSTEM SERVICE *service-name* SEVERED**

**Explanation:** An IUCV connection to a system service was severed. System services are listed in z/VM CP Programming Services.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0201L *userid* "cmd" VID=*vmid* SRC=*source-id* CLS=*c***

**Explanation:** The command *cmd* was issued by *userid*. VID is the *vmid* originating the command. The internal source path is identified by *source-id*. CLS is the message class.

**System action:** The program continues.

**User response:** No action is required.

---

---

**GOM0202E UNKNOWN COMMAND *cmd***

**Explanation:** An unknown command, represented by *cmd*, was encountered by Operations Manager.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0203E *userid* IS NOT AUTHORIZED FOR *cmd***

**Explanation:** The *userid* has not been authorized to execute the *cmd* command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0204I *text***

**Explanation:** A user has issued the LOGTEXT command to log an entry in the Operations Manager log.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0205E COMMAND *cmd* SYNTAX NOT VALID**

**Explanation:** An error exists in the syntax of the input command *cmd*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0206L *output***

**Explanation:** This message is used to log the output from the CP CMD()Operations Manager command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0207E MAXIMUM CONTROLLED USERIDS REACHED FOR *userid***

**Explanation:** An attempt was made to authorize too many CONSOLES for *userid*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0208E SCAN FAILED *nn***

**Explanation:** A call to the SCAN function failed with code *nn*.

**System action:** The program continues.

**User response:** See SCAN in the z/VM documentation for a list of return codes.

---

---

**GOM0209E COMMAND RC= *nn***

**Explanation:** CMS or CP processing of a command returned a non-zero return code.

**System action:** The program continues.

**User response:** See z/VM documentation of returns codes.

---

**GOM0210L *output***

**Explanation:** This message is used to log the output from the CMS CMD() Operations Manager command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0211E command name ALREADY DEFINED**

**Explanation:** An attempt was made to define a rule, action, schedule, machine monitor, spool monitor, group, or date that already exists. The name of the duplicate is *name*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0212E command name NOT FOUND**

**Explanation:** An attempt was made to delete a rule, action, schedule, machine monitor, spool monitor, group, or date that does not exist. The name you were attempting to delete is *name*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0213E CONFIGURATION FILE "*filename filetype filemode*" NOT FOUND**

**Explanation:** The CONFIG command is unable to find the file specified.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0214E SERVICE MACHINE *svmid* IS PROCESSING**

**Explanation:** An attempt was made to delete a busy service machine. The name of the service machine is *svmid*. If required, use FORCE(Y) to force deletion.

**System action:** The program continues.

**User response:** No action is required.

---

---

**GOM0215E PROCESSING COMPLETION  
RESPONSE TO *userid* FAILED**

**Explanation:** An attempt was made to send the output of a remotely processed command to GOMMAIN, but it failed. The service machine will be marked busy until DELSRVM with FORCE is processed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0216L *userid "msg" VID=vmid SRC=source-id  
CLS=c***

**Explanation:** The message *msg* was originated from *userid* and received by Operations Manager. The message was forwarded by the CP system service or VM identified by *vmid*. The internal source path is identified by *source-id*. CLS is the message class.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0217E COMMAND *cmd* NOT VALID FROM  
*src***

**Explanation:** An attempt was made to execute an Operations Manager command from an invalid source, represented by *src*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0218L USER *monitoring-user* ISSUED  
COMMAND *cmd* TO *monitored-user***

**Explanation:** The user *monitoring-user* is monitoring the console of *monitored-user* and has issued the command *cmd*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0219E COMBINED VIEW COMMAND  
FORMAT ERROR (FROM *userid*)**

**Explanation:** The user *userid* entered a command incorrectly from a combined view. The command syntax should be *targetuid command*.

**System action:** The program continues.

**User response:** Correct the syntax and retry the command.

---

**GOM0220E COMBINED VIEW COMMAND  
TARGET *targetuid* NOT ELIGIBLE  
(FROM *userid*)**

**Explanation:** The user *userid* entered a command from

a combined view for *targetuid* which is not in the combined view.

**System action:** The program continues.

**User response:** Specify a *targetuid* that is in the combined view.

---

**GOM0221E COMBINED VIEW COMMAND  
TARGET *targetuid* NOT MONITORED  
(FROM *userid*)**

**Explanation:** The user *userid* entered a command from a combined view for *targetuid* which is not monitored by Operations Manager.

**System action:** The program continues.

**User response:** Setup *targetuid* as an Operations Manager monitored user using the SET SECUSER or SET OBSERVER command.

---

**GOM0222E ESM CALL RETURNED SAF=*xx*  
RACF=*xxxxxx***

**Explanation:** A call failed or ESM (External Security Manager) could not determine if the requesting user has authority to execute an Operations Manager command. The SAF return code *xx* and the RACF three-digit (hex) return code and three-digit (hex) reason code can be found in the *RACF External Security Interface (RACROUTE) Macro Reference*.

**System action:** The request to execute the Operations Manager command is denied.

**User response:** Resolve the SAF and RACF return codes.

---

**GOM0224L *userid* EVENT TYPE *nn* VID=*vmid*  
SRC=*source-id* CLS=*c***

**Explanation:** An event for *userid* of type *nn* has occurred. The event notification was forwarded by the CP system service or VM identified by *vmid*. The internal source path is identified by *source-id*. CLS is the message class.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0225E Operand *operand* SYNTAX NOT VALID**

**Explanation:** The operand listed could not be parsed.

**System action:** The program continues.

**User response:** No action is required.

---

---

**GOM0226E OPERAND *operand* IS NOT VALID FOR COMMAND *command***

**Explanation:** The operand listed is not valid for the command entered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0227E OPERAND *operand* SPECIFIED MULTIPLE TIMES**

**Explanation:** The operand listed was entered multiple times.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0228I RULE *ruleid* PROCESSING LIMIT EXCEEDED**

**Explanation:** The rule *ruleid* was defined with a LIMIT operand. The start of the rule's ACTION operand is being suppressed because the rule's LIMIT operand count per interval is exceeded. The rule's ACTION operand will start when the LIMIT operand interval expires.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0229I EVENT *emonid* PROCESSING LIMIT EXCEEDED**

**Explanation:** The event *emonid* was defined with a LIMIT operand. The start of the event's ACTION operand is being suppressed because the event's LIMIT operand count per interval is exceeded. The event's ACTION operand will start when the LIMIT operand interval expires.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0230E OPERAND VALUE FOR OPERAND *operand* SYNTAX NOT VALID**

**Explanation:** The operand value entered could not be processed. Possible reasons are the operand value is too long or short, the value is out of range, or improperly formatted.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0231E REQUIRED OPERAND *operand* OMITTED**

**Explanation:** The operand indicated is required to process the command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0232E OPERAND *operand* MUTUALLY EXCLUSIVE WITH ANOTHER OPERAND**

**Explanation:** An operand entered is mutually exclusive with another operand entered for the command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0233E AN OPTIONAL OPERAND IS REQUIRED**

**Explanation:** One of the optional operands for the command is required.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0234E LOG FILE FOR DATE *yyyy/mm/dd* NOT FOUND**

**Explanation:** The log file for the requested date was not found.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0235L *userid data* VID=*vmid* SRC=*source-id* CLS=*cc* TYP=*tt***

**Explanation:** An event for *userid* of the class *cc* and type *tt* has occurred.

**System action:** The event notification was forwarded by the CP system service or VM identified by *vmid*. The internal source path is identified by *source-id*. The data shown was provided in the event notification.

**User response:** No action is required.

---

**GOM0236L *userid data* VID=*vmid* SRC=*source-id* CLS=*cc* TYP=*tt***

**Explanation:** An event for *userid* of the class *cc* and type *tt* has occurred.

**System action:** The event notification was forwarded by the CP system service or VM identified by *vmid*. The internal source path is identified by *source-id*. The data shown was provided in the event notification.

**User response:** No action is required.

---

---

**GOM0237E TCPAPPL *tcpappl* DOES NOT SUPPORT COMMANDS**

**Explanation:** The TCP/IP application named *tcpappl* does not support interaction using the CMDTCPA command.

**System action:** The program continues.

**User response:** Determine if the TCP/IP application is defined with a TCPAPPL operand which supports command interaction.

---

**GOM0238E REXX VARIABLE *varname* NOT LOCATED RC=*rc***

**Explanation:** The CMDTCPA command specified a REXXVAR(*varname*) operand which could not be located in the REXX environment.

**System action:** The program continues.

**User response:** Determine why the REXX variable could not be located.

---

**GOM0251E CONFIGURATION FILE "*filename filetype filemode*" OPEN FAILED *rc***

**Explanation:** The CONFIG command is unable to open the specified file, resulting in a return code of *rc*.

**System action:** The program continues.

**User response:** See FSOPEN in the z/VM documentation for a list of return code descriptions.

---

**GOM0252E CONFIGURATION FILE "*filename filetype filemode*" READ FAILED *rc***

**Explanation:** The CONFIG command is unable to read the specified file, resulting in a return code of *rc*.

**System action:** The program continues.

**User response:** See FSREAD in the z/VM documentation for a list of return code descriptions.

---

**GOM0253E CONFIGURATION FILE "*filename filetype filemode*" CLOSE FAILED *nn***

**Explanation:** The CONFIG command is unable to close the specified file, resulting in a return code of *nn*.

**System action:** The program continues.

**User response:** See FSCLOSE in the z/VM documentation for a list of return code descriptions.

---

**GOM0254L BEGIN FILE *fn ft fm* LAST UPDATE *date***

**Explanation:** This message is issued before the contents of a file that contains Operations Manager commands is processed. The file name, file type, and file mode are listed.

**System action:** The program continues.

---

**User response:** No action is required.

---

**GOM0255L END FILE *fn ft fm***

**Explanation:** This message is issued after the contents of a file that contains Operations Manager commands is processed. The file name, file type, and file mode are listed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0260I type *name* ACTION *actnid* TRIGGERED BY *userid***

**Explanation:** An action is triggered by a RULE, SCHEDULE, or MONITOR. *name* is the name of the RULE, SCHEDULE, or MONITOR triggering the action. *actnid* is the name of the action triggered. *userid* is the name of the console recording the triggering.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0261E ACTION *actnid* NOT FOUND FOR RULE | SCHEDULE | MONITOR**

**Explanation:** An attempt to locate the action *actnid* that was specified for the rule *ruleid*, schedule *schdid*, machine monitor *mmonid*, spool monitor *smonid*, or event monitor *emonid* was not successful.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0262I ACTION *action-name* BEGIN FOR *userid* SERVER *service-machine***

**Explanation:** The action, listed as *action-name*, is beginning execution on *server-machine*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0265E UNABLE TO SEND WORK UNIT TO SERVER *service-machine***

**Explanation:** An attempt was made to send a request to execute a command on a service machine. However, this attempt failed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0266E SERVICE MACHINE *userid* NOT LOCATED**

**Explanation:** The user specified by *userid* attempted to execute a GOMSVMS command without the corresponding DEF SRVM configuration command.

---

## GOM0267I • GOM0301E

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0267I ACTION** *action-name* **END RC=nn**  
**SERVER** *service-machine*

**Explanation:** The *action-name* ended with return code *nn* on *service-machine*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0268E COMMAND BUILD EXCEEDED**  
**AVAILABLE SPACE**

**Explanation:** While dynamically building a command for execution, all of the available build space was exhausted.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0269L COMMAND** *cmd*

**Explanation:** This message is used to log the dynamically constructed command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0270L** *output-text*

**Explanation:** This message is used to log the command output.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0271E SET ALTUSER FOR** *userid alternate-userid*  
**FAILED** *rc*

**Explanation:** A diagnose D4 to set the SECUSER to *alternate-userid* for *userid* failed *rc*.

**System action:** The program continues.

**User response:** Determine the reason the diagnose D4 failed and correct the error.

---

**GOM0281I SECUSER OR OBSERVER NOT SET**  
**FOR** *userid*

**Explanation:** A VIEWCON command requested data for a user that is not being monitored.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0282I MAXIMUM OBSERVERS FOR** *userid* **IN**  
**USE**

**Explanation:** A VIEWCON command requested data for a user that is being monitored by the maximum number of observers (other users).

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0283I UNABLE TO DELIVER DATA TO** *userid*

**Explanation:** An attempt to send data to a user executing the VIEWLOG or VIEWCON command failed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0284E ADDRESS SPACE CREATE FOR**  
*data-space* **FAILED** *rc*

**Explanation:** An attempt to create a data space with the name *data-space* failed *rc*.

**System action:** The program continues.

**User response:** Determine the reason the data space create failed and correct the error.

---

**GOM0285E UNRECOVERABLE ERROR ON**  
**VIRTUAL PRINTER***device-number*

**Explanation:** An error occurred while processing a VIEWCON, VIEWLOG, or VIEWSPPL command with the MODE(RDR) operand.

**System action:** The program continues.

**User response:** Determine why the command encountered an error and correct the problem.

---

**GOM0286E NO DATA TO DISPLAY**

**Explanation:** The current console or current log does not contain any lines within the DATERNNG operand specified on the VIEWCON or VIEWLOG command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0301E CONSOLE OPEN FAILED** *rc*

**Explanation:** An attempt was made to open a console, however, this attempt failed with return code *rc*.

**System action:** Operation Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM0302E CONSOLE QUERY FAILED** *rc*

**Explanation:** An attempt was made to query a console, however, this attempt failed with return code *rc*.

**System action:** Operation Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM0303E CONSOLE DEVICE NOT SUPPORTED**  
(*hex-data*)

**Explanation:** The device being used when executing VIEWCON or VIEWLOG is not supported. The *hex-data* describes the console attributes.

**System action:** Operation Manager terminates.

**User response:** No action is required.

---

**GOM0304E CONSOLE WRITE FAILED** *rc*

**Explanation:** An attempt was made to write to a console, however, this attempt failed with return code *rc*.

**System action:** Operation Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of code descriptions.

---

**GOM0305E CONSOLE READ FAILED** *rc*

**Explanation:** An attempt was made to read a console, however, this attempt failed with return code *rc*.

**System action:** Operation Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM0306E RETURN CODE** *rc* **FROM** *cmd*

**Explanation:** The command *cmd* returned a non-zero return code *rc*.

**System action:** Processing continues.

**User response:** No action is required.

---

**GOM0351E CONSOLE OPEN FAILED** *rc*

**Explanation:** An attempt was made to open a console, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

---

**GOM0352E CONSOLE QUERY FAILED** *rc*

**Explanation:** An attempt was made to query a console, however, this attempt failed with return code *rc*.

**System action:** Operation Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM0353E CONSOLE DEVICE NOT SUPPORTED**  
*hex-data*

**Explanation:** The device being used when executing VIEWSPPL is not supported. The *hex-data* describes the console attributes.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

**GOM0354E CONSOLE WRITE FAILED** *rc*

**Explanation:** An attempt was made to write to a console, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM0355E CONSOLE READ FAILED** *rc*

**Explanation:** An attempt was made to read a console, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM0356E SPOOL FILE NOT FOUND OR NOT VIEWABLE FOR USER** *userid* **FILE** *fileid*

**Explanation:** The spool file requested was not found or is not a format viewable by VIEWSPPL.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0358W SPOOL ACCESS FOR** *data-1* **CLASS**  
*data-2* **FAILED** *data-3*

**Explanation:** An attempt to access spool data failed. The data items list the error information.

**System action:** The program continues.

**User response:** No action is required.

---

---

**GOM0359E** RETURN CODE *rc* FROM *cmd*

**Explanation:** The command *cmd* returned a non-zero return code *rc*.

**System action:** Processing continues.

**User response:** None required.

---

**GOM0401I** SPOOL USE: MONITOR *smonid* SPACE  
*nn* PERCENT, FILES *nn* PERCENT

**Explanation:** A USAGE or INCREASE condition was encountered and the specified percentage of data space and file count was reached.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0402I** SPOOL CHG: MONITOR *smonid* SPACE  
*nn* PERCENT, FILES *nn* PERCENT

**Explanation:** A USAGE or INCREASE condition was encountered and the specified percentage of data space and file count used has changed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0403I** SPOOL ALERT: MONITOR *smonid* *nn*  
CONDITION REACHED

**Explanation:** A USAGE or INCREASE condition was encountered for the specified spool usage monitor.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0404E** UNABLE TO COMPUTE SPOOL  
PERCENTAGES

**Explanation:** An attempt was made to execute the required commands to monitor spool conditions and a failure occurred.

**System action:** Operations Manager cannot monitor spool usage conditions.

**User response:** Ensure that the CP directory for GOMMAIN has PRIV class D.

---

**GOM0405I** SPOOL MONITOR *smonid*  
PROCESSING LIMIT EXCEEDED

**Explanation:** The spool monitor *smonid* was defined with a LIMIT operand. The start of the spool monitor's ACTION operand is being suppressed because the rule's LIMIT operand count per interval is exceeded. The spool monitor's ACTION operand will start when the LIMIT operand interval expires.

**System action:** The program continues.

---

**User response:** No action is required.

---

**GOM0411E** TCP/IP OPERATION *xxxxxxx* TO  
*tcpvmid* SOCK *xxxx* FAILED AT OFFSET  
*xxxxxxx*

**Explanation:** A TCP/IP operation failed to start. The operations type, the VMID of the TCPIP machine, the socket number, and offset in the module are listed.

**System action:** The program continues.

**User response:** Determine the cause of the error and correct the problem.

---

**GOM0412E** TCP/IP OPERATION *xxxxxxx* TO  
*tcpvmid* SOCK *xxxx* FAILED *xxxxxxx* AT  
OFFSET *xxxxxxx*

**Explanation:** A TCP/IP operation did not complete correctly. The operation type, the VMID of the TCPIP machine, the socket number, the return code, and offset in the module are listed.

**System action:** The program continues.

**User response:** Determine the cause of the error and correct the problem.

---

**GOM0413E** FSOPEN FAILED *rc* FOR *filename filetype*

**Explanation:** The attempt to open a file using FSOPEN failed.

**System action:** The program continues.

**User response:** Determine why the file was unavailable for processing.

---

**GOM0414E** FSREAD FAILED *rc* FOR *filename filetype*

**Explanation:** The attempt to read from a file using FSREAD failed.

**System action:** The program continues.

**User response:** Determine why the file was unavailable for processing.

---

**GOM0415E** FSWRITE FAILED *rc* FOR *filename  
filetype*

**Explanation:** The attempt to write to a file using FSWRITE failed.

**System action:** The program continues.

**User response:** Determine why the file was unavailable for processing.

---



---

**GOM0416E** FSCLOSE FAILED *rc* FOR *filename*  
*filetype*

**Explanation:** The attempt to close a file using FSCLOSE failed.

**System action:** The program continues.

**User response:** Determine why the file was unavailable for processing.

---

**GOM0417I** *req-type* REQUEST *reqid* FROM *userid* TO  
*tcpaid* BEGIN

**Explanation:** The processing of the request *req-type* from *userid* to *tcpaid* is starting.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0418I** *req-type* REQUEST *reqid* FROM *userid* TO  
*tcpaid* ENDED

**Explanation:** The processing of the request *req-type* from *userid* to *tcpaid* is complete.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0419E** *req-type* REQUEST *reqid* FROM *userid* TO  
*tcpaid* FAILED

**Explanation:** The processing of the request *req-type* from *userid* to *tcpaid* encountered errors.

**System action:** The program continues.

**User response:** Determine why the request could not complete without errors.

---

**GOM0420I** *req-type* REQUEST *reqid* FROM *userid* TO  
*tcpaid* QUEUED

**Explanation:** The processing of the request *req-type* from *userid* to *tcpaid* is queued for subsequent processing.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0431I** MACHINE MONITOR *mmonid*  
PROCESSING LIMIT EXCEEDED

**Explanation:** The machine monitor *mmonid* was defined with a LIMIT operand. The start of the machine monitor's ACTION operand is being suppressed because the rule's LIMIT operand count per interval is exceeded. The machine monitor's ACTION operand will start when the LIMIT operand interval expires.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0441I** SCHEDULE *schdid* PROCESSING  
LIMIT EXCEEDED

**Explanation:** The schedule *schdid* was defined with a LIMIT operand. The start of the schedule's ACTION operand is being suppressed because the rule's LIMIT operand count per interval is exceeded. The schedule's ACTION operand will start when the LIMIT operand interval expires.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0451I** PAGE USE: MONITOR *pmonid* SPACE  
*nn* PERCENT

**Explanation:** A USAGE or INCREASE condition was encountered and the specified percentage of page space was reached.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0452I** PAGE CHG: MONITOR *pmonid* SPACE  
*nn* PERCENT

**Explanation:** A USAGE or INCREASE condition was encountered and the specified percentage of page space was reached.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0453I** PAGE ALERT: MONITOR *pmonid* *nn*  
CONDITION REACHED

**Explanation:** A USAGE or INCREASE condition was encountered for the specified page usage monitor.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0454E** UNABLE TO COMPUTE PAGE  
PERCENTAGES

**Explanation:** An attempt was made to issue the required commands to monitor page conditions and a failure occurred.

**System action:** Operations Manager cannot monitor page usage conditions

**User response:** Ensure that the CP directory for GOMMAIN has PRIV class D.

---

**GOM0455I** PAGE MONITOR *pmonid* PROCESSING  
LIMIT EXCEEDED

**Explanation:** The page space monitor *pmonid* was defined with a LIMIT operand. The start of the page space monitor's ACTION operand is being suppressed

because the rule's LIMIT operand count per interval is exceeded. The page space monitor's ACTION operand will start when the LIMIT operand interval expires.

**System action:** The program continues.

**User response:** No action is required.

**GOM0461I IDLE MONITOR *imonid* PROCESSING LIMIT EXCEEDED**

**Explanation:** The idle monitor *imonid* was defined with a LIMIT operand. The start of the idle monitor's ACTION operand is suppressed because the idle monitor's LIMIT operand count per interval has exceeded. The idle monitor's ACTION operand will start with the LIMIT operand interval ends.

**System action:** The program continues.

**User response:** No action is required.

**GOM0462E IDLE MONITOR *imonid resource-type resource-id* NOT FOUND**

**Explanation:** The idle monitor *imonid* could not locate the recourse to monitor.

**System action:** Operations Manager cannot monitor the resource activity.

**User response:** Define the resource to monitor.

**GOM0899I OPERATIONS MANAGER V1R5 BUILT *yy/mm/dd hh:mm***

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you the date and time of the Operations Manager build you are running.

**System action:** The program continues.

**User response:** No action is required.

**GOM0900I *nn* RULE(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many rules are defined.

**System action:** The program continues.

**User response:** No action is required.

**GOM0902I *nn* MACHINE MONITOR(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many monitored virtual machines are defined.

**System action:** The program continues.

**User response:** No action is required.

**GOM0903I *nn* SCHEDULE(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many schedules are defined.

**System action:** The program continues.

**User response:** No action is required.

**GOM0904I *nn* ACTION(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many actions are defined.

**System action:** The program continues.

**User response:** No action is required.

**GOM0905I *nn* ACTION(S) SCHEDULED FOR PROCESSING**

**Explanation:** The STATUS command outputs various information about the current processing environment. This message indicates how many actions are scheduled for processing.

**System action:** The program continues.

**User response:** No action is required.

**GOM0906I *n* GROUP(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates how many groups are defined.

**System action:** The program continues.

**User response:** No action is required.

**GOM0907I *nn* DATE(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many holidays are defined.

**System action:** The program continues.

**User response:** No action is required.

**GOM0908I *nn* SPOOL MONITOR(S) DEFINED**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates how many spool monitors are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0909I** *nn* EVENT MONITOR(S) DEFINED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates how many event monitors are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0910I** *nn* PAGE MONITOR(S) DEFINED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates how many page space monitors are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0911I** IDLE MONITOR(S) DEFINED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates how many idle monitors are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0920I** *nn* SERVICE MACHINE(S) DEFINED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many service machines are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0921I** *nn* USER(S) AUTHORIZED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many users were granted authority.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0922I** *nn* IUCV CONNECTION(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates the number of IUCV connections.

**System action:** The program continues.

**User response:** No action is required.

---



---

**GOM0923I** *nn* DATA SPACE SIZE DEFINITION(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many data space definitions are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0924I** *nn* COMBINED VIEW(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many combined views are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0925I** *nn* USER CONSOLE(S) RECORDING

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many user consoles are recording.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0926I** *nn* TCP/IP APPLICATION(S) DEFINED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many TCP/IP applications are defined.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0940I** RULE *rule-name* TRIGGERED *nn* TIME(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many times the rule *rule-name* was triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0942I** MACHINE MONITOR *mmonid* TRIGGERED *nn* TIME(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many times the monitored virtual machine was triggered.

**System action:** The program continues.

---

## GOM0943I • GOM0960I

**User response:** No action is required.

---

**GOM0943I SCHEDULE** *schdid* **TRIGGERED** *nn*  
**TIME(S)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many times the schedule *schdid* was triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0944I ACTION** *actnid* **PROCESSED** *nn*  
**TIME(S)**

**Explanation:** The STATUS command outputs various information about the current processing environment. This message tells you how many times the action *actnid* was processed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0945I ACTION** *actnid* **SCHEDULED FOR**  
**PROCESSING**  
**(RULE|SCHEDULE|MONITOR/**  
*userid|schedule-name|monitor name)*

**Explanation:** The STATUS command outputs various information about the current processing environment. This message tells you when the action is scheduled for processing and what triggered the action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0946I GROUP** *groupid* **INCLUDES USER** *userid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you the user ID that is included in this group.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0947I DATE** *dateid* **FROM** *yyyy/mm/dd-hh:mm*  
**TO** *yyyy/mm/dd-hh:mm*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you the date and time range for the defined holiday.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0948I SPOOL MONITOR** *smonid* **TRIGGERED**  
*nn* **TIME(S)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many times the specified spool monitor was triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0949I EVENT MONITOR** *emonid* **TRIGGERED**  
*nn* **TIME(S)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many times an event monitor was triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0950I PAGE MONITOR** *pmonid* **TRIGGERED**  
*nn* **TIME(S)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many times a page space monitor was triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0951I IDLE MONITOR** *imonid* **TRIGGERED**  
*nn* **TIME(S)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message indicates how many times an idle monitor was triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0960I SERVICE MACHINE** *srvmid*  
**PROCESSED** *nn* **COMMAND(S)**

**Explanation:** The STATUS command outputs various information about the current processing environment. This message tells you how many commands the specified service machine has processed.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0961I** USER *userid* AUTHORIZED FOR:

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is followed by the authorization(s) for *userid*.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0962I** USER *userid alias* PATHID *pathid*  
MSGLIMIT *nnn* MSGS *nnn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about IUCV connection information for *userid*, *alias*, *pathid*, the allowed number of outstanding messages, and the count of messages exchanged.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0963I** DATA SPACE FOR CONSOLE *pattern*  
SIZE *nnM*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about defined data space size definitions.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0964I** COMBINED VIEW *viewid* INCLUDES  
USER *userid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about combined views.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0965I** CONSOLE *userid* ASIT *xxxxxxxxxxxxxxxxxxx*  
ALET *xxxxxxxx* SIZE *nnM* MSGS  
*msg-count*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about consoles being recorded.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0966I** NAME *tcpaid* USER *tcpip alias* APPL  
*applid* SOCKET *nn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about TCP/IP connections and processing applications.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0967I** ADDR *nnn.nnn.nnn.nnn* PORT *nn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about TCP/IP connections and processing applications.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0968I** ADDR  
*hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh*  
SCOPE *nn* PORT *nn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you about TCP/IP connections and processing applications.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0969I** PARM *parm*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the TCP/IP application PARM operand value.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0970I** USER *userid* SPOOL FILE ID *fileid* IS  
USING *nn* SPOOL BLOCKS

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you the number of spool blocks used by a file belonging to the specified user.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM0971I** USER *userid* HAS *nn* SPOOL FILES  
USING *nn* SPOOL BLOCKS

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells you how many spool files and spool

## GOM1050E • GOM1113E

blocks for the specified user.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM1050E ADDRESS SPACE QUERY FAILED *rc*

**Explanation:** An attempt was made to query an address space, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CP ADRSPACE in the z/VM documentation for a list of return code descriptions.

---

### GOM1051E NUCLEUS DROP FAILED *rc*

**Explanation:** An attempt was made to drop a nucleus extension, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS NUCEXT in the z/VM documentation for a list of return code descriptions.

---

### GOM1052E NUCLEUS LOAD FAILED *rc*

**Explanation:** An attempt was made to load a module, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS NUCEXT in the z/VM documentation for a list of return codes description.

---

### GOM1053E NUCLEUS QUERY FAILED *rc*

**Explanation:** A query attempt was made, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS NUCEXT in the z/VM documentation for a list of return code descriptions.

---

### GOM1054E EXTERNAL HANDER SET FAILED *rc*

**Explanation:** A set attempt was made, however, this attempt failed with a return code of *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS HNDEXT in the z/VM documentation for a list of return code descriptions.

---

### GOM1055E EXTENDED PARAMETER LIST NOT AVAILABLE

**Explanation:** When an Operations Manager command is started, it attempts to obtain the extended parameter list. The Operations Manager command was unable to obtain the extended parameter list.

**System action:** The Operations Manager command terminates.

**User response:** Determine why the Operations Manager command is unable to obtain an extended parameter list.

---

### GOM1099S UNRECOVERABLE ERROR ENCOUNTERED

PSW=xxxxxxxxxxxxxxxxx EP=xxxxxx  
MCB=xxxxxx

**Explanation:** An unrecoverable error condition was encountered.

**System action:** Operations Manager terminates.

**User response:** No action is required.

---

### GOM1101E ABNEXIT SET FAILED, RC=*rc*

**Explanation:** A set attempt was made, however, this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS ABNEXIT in the z/VM documentation for a list of return code descriptions.

---

### GOM1102E ABNEXIT CLR FAILED, RC=*rc*

**Explanation:** A clear attempt was made, however this attempt failed with return code *rc*.

**System action:** Operations Manager terminates.

**User response:** See CMS ABNEXIT in the z/VM documentation for a list of return code descriptions.

---

### GOM1111E LOGICAL DEVICE TABLE FULL

**Explanation:** Too many sessions are active. No new sessions are allowed until some sessions are ended.

**System action:** The program continues.

**User response:** End a session.

---

### GOM1112E LOGICAL DEVICE INIT FAILED CC=*cc* RC=*rc*

**Explanation:** A DIAG 007C failure has occurred.

**System action:** The program continues.

**User response:** See CP Diagnose 007C in the z/VM documentation for a list of return code descriptions.

---

### GOM1113E CMSSTACK FAILED *rc*

**Explanation:** A CMSSTACK operation was made, however, this operation failed with return code *rc*.

**System action:** The program continues.

**User response:** See CMS CMSSTACK in the z/VM

documentation for a list of return code descriptions.

---

**GOM1114E CONSOLE OPEN FAILED *rc*, TRACE IGNORED**

**Explanation:** A CONSOLE operation failed with return code *rc*.

**System action:** The program continues.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM1121E HANDLE NOT FOUND**

**Explanation:** An attempt to TERM a session failed because the specified handle was not found.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM1122E LOGICAL DEVICE TERM FAILED  
CC=*cc* RC=*rc***

**Explanation:** A DIAG 007C failure has occurred.

**System action:** The program continues.

**User response:** See CP Diagnose 007C in the z/VM documentation for a list of return code descriptions.

---

**GOM1123E CONSOLE CLOSE FAILED *rc***

**Explanation:** An attempt was made to close the console, however, this attempt failed with return code *rc*.

**System action:** The program continues.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM1131E HANDLE NOT FOUND**

**Explanation:** An attempt to QUERY a session failed because the specified handle was not found.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM1132E CMSSTACK FAILED *rc***

**Explanation:** A CMSSTACK operation was made, however, the operation failed with return code *rc*.

**System action:** The program continues.

**User response:** See CMS CMSSTACK in the z/VM documentation for a list of return code descriptions.

---

**GOM1133E FSOPEN FAILED *rc***

**Explanation:** An FSOPEN operation failed with return code *rc*.

**System action:** The program continues.

**User response:** See CMS FSOPEN in the z/VM documentation for a list of return code descriptions.

---

**GOM1134E FSWRITE FAILED *rc***

**Explanation:** An FSWRITE operation failed with a return code of *rc*.

**System action:** The program continues.

**User response:** See CMS FSWRITE in the z/VM documentation for a list of return code descriptions.

---

**GOM1135E FSCLOSE FAILED *rc***

**Explanation:** An FSCLOSE operation failed with a return code of *rc*.

**System action:** The program continues.

**User response:** See CMS FSCLOSE in the z/VM documentation for a list of return code descriptions.

---

**GOM1141E HANDLE NOT FOUND**

**Explanation:** An attempt to INPUT a session failed because the specified handle was not found.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM1142E CONSOLE WRITE FAILED *rc***

**Explanation:** A CONSOLE operation failed with a return code of *rc*.

**System action:** The program continues.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

**GOM1151E LOGICAL DEVICE WAS  
TERMINATED**

**Explanation:** The session for a logical device was terminated by CP.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM1152E LOGICAL DEVICE ACCEPT FAILED  
CC=*cc* RC=*rc***

**Explanation:** A DIAG 007C failure has occurred.

**System action:** The program continues.

**User response:** See CP Diagnose 007C in the z/VM

## GOM1153E • GOM2933I

documentation for a list of return code descriptions.

---

### GOM1153E LOGICAL DEVICE STATUS FAILED CC=*cc* RC=*rc*

**Explanation:** A DIAG 007C failure has occurred.

**System action:** The program continues.

**User response:** See CP Diagnose 007C in the z/VM documentation for a list of return code descriptions.

---

### GOM1154E WRITE CCW OPERATION CODE *xx* NOT VALID

**Explanation:** A CCW with operation code *xx* was received. The code *xx* was not recognized code.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM1155E CONSOLE WRITE FAILED *rc*

**Explanation:** A CONSOLE operation failed with return code *rc*.

**System action:** The program continues.

**User response:** See CMS CONSOLE in the z/VM documentation for a list of return code descriptions.

---

### GOM1156E LOGICAL DEVICE PRESENT FAILED CC=*cc* RC=*rc*

**Explanation:** A DIAG 007C failure has occurred.

**System action:** The program continues.

**User response:** See CP Diagnose 007C in the z/VM documentation for a list of return code descriptions.

---

### GOM1157E STRUCTURED FIELD ERRORS SF=*data*

**Explanation:** An unsupported structured field was encountered. The field in error is displayed.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM1158E DATA STREAM ERROR DATA=*data*

**Explanation:** An unsupported data stream was encountered. The data stream in error is displayed.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM1181E LOGICAL DEVICE PRESENT FAILED CC=*cc* RC=*rc*

**Explanation:** A DIAG 007C failure has occurred.

**System action:** The program continues.

**User response:** See CP Diagnose 007C in the z/VM documentation for a list of return code descriptions.

---

### GOM2929I TCP/IP INTERACTION FOR *pattern*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the TCP/IP applications a user is authorized to interact with.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2930I LAST TRIGGERED *date time*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists a rule, a schedule, or a monitor's last trigger date and time.

**System action:** Processing continues.

**User response:** None required.

---

### GOM2931I LAST PROCESSING COMPLETE *date time*

**Explanation:** The STATUS command outputs various information about the current processing environment. This message lists an action's last processing date and time.

**System action:** Processing continues.

**User response:** None required.

---

### GOM2932I MONITOR *resource-type resource-id* TO OCCUR *nn* TIME(S) WITHIN *nn* MINUTE(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the type, ID, and expected execution occurrences of a resource.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2933I EVENT CLASS *classes*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the type of an event monitor.

**System action:** The program continues.

**User response:** No action is required.



---

**GOM2934I** EVENT TYPE *events*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the type of an event monitor.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2935I** SUBTRACT *nnn* FROM LINE SUPPRESS COUNT

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the line suppression value adjustment if the rule is triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2936I** ADD *nnn* TO LINE SUPPRESS COUNT

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the line suppression value adjustment if the rule is triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2937I** SET LINE SUPPRESS COUNT TO *nnn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the line suppression value adjustment if the rule is triggered.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2938I** SUPPRESS NEXT *nnn* LINES

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current value of the line suppression count for a console.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2939I** ESMAPREF(*esmprefix*)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current DEFOPTN ESMAPREF.

**System action:** The program continues.

**User response:** No action is required.

---



---

**GOM2940I** PROCESSING ENV OVERRIDE *env*

**Explanation:** The STATUS command outputs various information about the current processing environment. This message lists if a rule or monitor provides an environment processing override.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2941I** PROCESSING ENV CAN BE OVERRIDDEN

**Explanation:** The STATUS command outputs various information about the current processing environment. This message lists if an action processing environment can be overridden.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2942I** PROCESSING SUPPRESSED *count* TIME(S)

**Explanation:** The STATUS command outputs various information about the current processing environment. This message lists how many times a rule or monitor processing was suppressed due to LIMIT.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2943I** PROCESSING LIMIT IS *count* OCCURRENCE(S) WITHIN *interval nnn*

**Explanation:** The STATUS command outputs various information about the current processing environment. This message lists the *count* and *interval* processing limits of a rule, schedule, or monitor. The interval amount, *nnn*, can be any of the following: .

- *nnn* DAY(S), *nn* HOUR(S), *nn* MINUTE(S), *nn* SECOND(S)
- *nn* HOUR(S), *nn* MINUTE(S), *nn* SECOND(S)
- *nn* MINUTE(S), *nn* SECOND(S)
- *nn* SECOND(S)

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2945I** EXCLUDE GROUP *groupid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the excluded group for a rule.

**System action:** The program continues.

**User response:** No action is required.

---

---

**GOM2946I EXCLUDE USER** *userid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the excluded user for a rule.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2947I AUTHCMD(Y|N)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current setting for DEFOPTN AUTHCMD.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2948I SYMBOLS(abc)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current setting for DEFOPTN SYMBOLS.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2949I DSTADJ(Y|N)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current setting for DEFOPTN DSTADJ.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2950I MIXCASE(Y|N)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current setting for DEFOPTN MIXCASE.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2951I HLDATTR(tttt,a,h,xat,clr)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the setting for DEFOPTN HLDATTR.

**System action:** The program continues.

**User response:** No action is required.

---



---

**GOM2952I ESM(Y|N)**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the current setting for DEFOPTN ESM.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2953I ALTERNATE SECLABEL** *seclabel*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the ALTSECLB for an action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2954I ALTERNATE USERID** *userid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the ALTUSER for an action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2955I EXCLUDE "string" BETWEEN COLUMNS** *begcol AND endcol*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists string matching exclusion information for a RULE.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2956I MATCH "string" BETWEEN COLUMNS** *begcol AND endcol*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists string matching information for a RULE.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2957I GROUP** *groupid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists *groupid* information for a RULE.

**System action:** The program continues.

**User response:** No action is required.

---

---

**GOM2958I USER** *userid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists *userid* information for a RULE.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2959I INTERVAL WHEN**(*interval*)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message appears when using the STATUS DETAIL(SCHEDULE) command and lists interval information from the input on the DEFSCHD command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2960I INTERVAL EVERY**(*interval*)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message appears when using the STATUS DETAIL(SCHEDULE) command and lists interval information from the input on the DEFSCHD command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2961I NEXT RUN** *CCYY/MM/DD hh:mm*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the next execution time information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2962I CURRENTLY ACTIVE**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message tells if a schedule is active.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2963I MAXIMUM DEFERAL** *nn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the maximum deferral time information for a schedule.

**System action:** The program continues.

---

**User response:** No action is required.

---

**GOM2964I schdid**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists conflicting schedules for a schedule.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2965I CONFLICTS WITH SCHEDULE(S) :**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is followed by the conflicts for a schedule.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2966I INCREASE LOW** *low* **HIGH** *high*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the low and high increase values for a spool monitor.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2967I USAGE LOW** *low* **HIGH** *high*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the low and high usage values for a spool monitor.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2968I NEXT CHECK** *CCYY/MM/DD hh:mm*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message list the next execution time information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2969I DEPENDS ON USER** *userid* **STS** *status*  
(*userid status*) {*connector*}

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the dependency information for a machine monitor.

**System action:** The program continues.

---

## GOM2970I • GOM2981I

**User response:** No action is required.

---

### GOM2970I RETRY INTERVAL *nn* MINUTE(S) RETRY COUNT *nn*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the retry information for a machine monitor.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2971I DELAY *nn* MINUTE(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the delay information for a machine monitor.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2972I INTERVAL *nn* MINUTE(S)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the interval information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2973I USER *userid* (*userid status*)

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the user information for a machine monitor.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2974I PARM "*parm*"

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the parm information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2975I HOURS *hours* eligible for execution

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the hours information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2976I DAYS *days* eligible for execution

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the days information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2977I ACTION *actnid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the action ID information for a resource.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2978I SUSPENDED

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists if a resource is suspended from execution.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2979I NEXT ACTION *actnid* DELAY *nn* MINUTES *nn* SECONDS

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the next action information for an action.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2980I COMMAND "*command*"

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the command information for an action.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2981I *attribute*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the input or output actions for an action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2982I OUTPUT ACTIONS:**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is followed by the output actions for an action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2983I INPUT ACTIONS:**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is followed by the input actions for an action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2984I PROCESSING ENV *env***

**Explanation:** The STATUS command outputs various information about the current processing environment. This message lists the processing environment for an action command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2985I LAST RC=*rc* ELAPSED TIME=*time in ms***

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the last action command return code and elapsed time.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2986I *wrkgrpid***

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the work groups eligible to execute an action command.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2987I PARTICIPATING IN PROCESSING ENV WORK GROUP(S):**

**Explanation:** The STATUS command outputs various information about the current processing environment. This message is followed by the work groups eligible to process an action command.

**System action:** The program continues.

**User response:** No action is required.

---



---

**GOM2988I PROCESSING ENV UNIQUE**

**Explanation:** The STATUS command outputs various information about the current processing environment. This message is listed if a service machine may only process actions with an *env()* matching the service name.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2989I PROCESSING ACTION *actnid***

**Explanation:** The STATUS command outputs various information about the current processing environment. This message is listed if a service is currently processing an action.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2990I NOT READY**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is listed if a service is not ready to execute actions.

**System action:** The program continues.

**User response:** Determine why the service machine is not logged on or not executing GOMSVMS.

---

**GOM2991I READY**

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is listed if a service is ready to execute actions.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2992I SPPOL FOR *pattern***

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the spool files a user is authorized to view.

**System action:** The program continues.

**User response:** No action is required.

---

**GOM2993I CONSOLE FOR *pattern***

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the consoles a user is authorized to view.

**System action:** The program continues.

---

## GOM2994I • GOM2999I

**User response:** No action is required.

**User response:** No action is required.

---

### GOM2994I CONFIGURATION COMMANDS

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists if a user is authorized for configuration commands.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2995I CONTROL COMMANDS

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists if a user is authorized for control commands.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2996I EXTERNAL SECURITY MANAGER IN USE

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists if an External Security Manager is in use for authorizations.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2997I *userid*

**Explanation:** The STATUS command outputs various information about the current execution environment. This message lists the users viewing a console.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2998I BEING VIEWED BY:

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is followed by a list of users viewing a console.

**System action:** The program continues.

**User response:** No action is required.

---

### GOM2999I STATUS DETAIL COMPLETE

**Explanation:** The STATUS command outputs various information about the current execution environment. This message is listed when the STATUS DETAIL command output is complete.

**System action:** The program continues.

---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

| Intellectual Property Licensing  
| Legal and Intellectual Property Law  
| IBM Japan Ltd.  
| 19-21, Nihonbashi-Hakozakicho, Chuo-ku  
| Tokyo 103-8510, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample



programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java<sup>™</sup> and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.



---

# Index

## A

- accessing
  - consoles 25
  - spools 26
- accounting
  - files 145
  - records 145
- action processing 2
- action processing servers
  - defining 27
- actions
  - defining 71
  - deleting 105
  - processing 34
- adding
  - comments 126
- adding comments 24
- ALTRCON command 124
- application
  - TCP/IP 26
- assigning
  - authority 25, 39
- audience xvii
- AUTH
  - external security 26
- AUTH command 25, 39
- authority
  - assign 39
  - assigning 25
  - revoking 61
  - TCP/IP apps 26

## C

- captured console data 46
- CMDTCPA command 41
- CMS command 42
- collecting
  - remote file input 151
  - syslog data 149
- commands
  - ALTRCON 124
  - configuration 69
    - CONFIG 43, 69
    - DEFACTN 71
    - DEFDATE 78
    - DEFEMON 79
    - DEFGROUP 85
    - DEFIMON 82
    - DEFMMON 87
    - DEFPMON 90
    - DEFRULE 94
    - DEFSCHD 98
    - DEFSMON 102
    - DELACTION 105
    - DELDATE 106
    - DELEMON 107
    - DELGROUP 108
    - DELMON 110
    - DELPMON 111

commands (*continued*)

- configuration (*continued*)
  - DELRULE 112
  - DELSCHD 112
  - DELSMON 113
  - RESUME 114
  - RUN 116
  - STATUS 64, 118
  - SUSPEND 121
- console access 123
- control 39
  - AUTH 39
  - CMS 42
  - CONFIG 43, 69
  - CP 45
  - DEFDPS 46
  - DEFOPTN 48
  - DEFSRVM 51
  - DEFTCPA 53
  - DEFVIEW 55
  - DELDPS 57
  - DELSRVM 58
  - DELTCPA 59
  - DELVIEW 60
  - QUIT 61
  - REVOKE 61
  - SHUTDOWN 63
  - STATUS 64, 118
  - STOP 67
  - VIEWLOG 131
- DEFTCPA 149
- entering 21
- GOMRSIF 152
- issue dynamically 2
- issuing
  - dynamically 32
- log access 123
- LOGTEXT 126
- rules 21
- spool access 123
- system management 31
  - CMDTCPA 41
  - GOMCMD 32
  - GOMMAIN 31
  - GOMSVM 34
  - GOMTDM 35
  - VIEWCON 127
  - IEWSPL 136
- comments 24
  - adding 126
- components 2
  - message processing 2
- CONFIG command 43, 69
- configuration
  - authority 25
- configuration commands 69
- configuration files
  - managing 28
- configuring
  - Operations Manager 5
- console commands 123

- console data
  - verifying 11
- console data space names 123
- consoles
  - accessing 25
  - viewing 127
- control
  - authority 25
- control commands 39
- controlling
  - Operations Manager 32
- copying
  - sample configuration file 23
- CP command 45
- CP directory 6
- customizing
  - log file management 147

## D

- data space names 123
- DEFACTN command 71
- DEFDATE command 78
- DEFDPS command 46
- DEFEMON command 79
- DEFGROUP command 85
- DEFIMON command 82
- defining
  - action processing servers 27
  - actions 71
  - configuration authority 25
  - console access 25
  - control authority 25
  - event monitors 79
  - groups 85
  - holidays 78
  - monitored virtual machines 87
  - multiple views 55
  - page space monitors 90
  - rules 94
  - schedules 98
  - security options 48
  - spool access 26
  - spool usage monitors 102
  - storage for captured data 46
  - system options 27
  - views 55
- DEFMMON command 87
- DEFOPTN
  - ESM(Y/N) 26
- DEFOPTN command 48
- DEFPMON command 90
- DEFRULE command 94
- DEFSCHD command 98
- DEFSMON command 102
- DEFSRVM command 27, 51
- DEFTCPA command 53, 149
- DEFVIEW command 55
- DELACTN command 105
- DELDATE command 106
- DELDSPTS command 57
- DELEMON command 107
- deleting
  - actions 105
  - authority 61
  - event monitors 107
  - groups 108

- deleting (*continued*)
  - holidays 106
  - monitored virtual machine 110
  - page space monitors 111
  - rules 112
  - schedules 112
  - service machines 58
  - spool usage monitors 113
    - deleting 113
  - storage definition 57
  - TCP/IP application 59
  - views 60
- DELGROUP command 108
- DELMON command 110
- DELPMON command 111
- DELRULE command 112
- DELSCHD command 112
- DELSMON command 113
- DELSRVM command 58
- DELTCPA command 59
- DELVIEW command 60
- directories
  - CP 10

## E

- entering
  - commands 21
- error messages 165
- ESMPREF
  - setting 12
- event monitors
  - defining 79
  - deleting 107
- examples
  - adding comments 24
  - assigning authority 25
  - AUTH command 25
  - defining action processing servers 27
  - DEFSRVM 27
  - GOMCMD command 33
  - STATUS command 65, 119
- external security
  - AUTH 26
- external security manager
  - profile prefix 12
- External Security Manager (ESM) 11, 48

## F

- FACILITY class
  - profiles 13
- files
  - accounting 145
  - configuration 5
  - MAS COMMANDS 31
  - monitoring 146
  - OPMGRM1 CONFSAMP 2, 23, 31
  - PROFILE EXEC 10, 28
  - PROFM1 SAMPEXEC 5
  - PROFS1 SAMPEXEC 7
  - PROFUSR SAMPEXEC 9
  - sending multiple 152
  - startup configuration 21, 23, 27

## G

- GOMCMD 2
- GOMCMD command 32
- GOMGLBL
  - preparing 161
- GOMGLBL facility 161
- GOMMAIN 28
  - stopping 61
- GOMMAIN command 31, 32
- GOMRSIF command 152
- GOMSVM 2
- GOMSVM command 34
- GOMTDM command 35
- groups
  - defining 85
  - deleting 108

## H

- hardware
  - requirements 3
- holidays
  - defining 78
  - deleting 106

## I

- idle monitors
  - defining 82
- IDs
  - administrator 9
  - monitored by Operations Manager 9
  - running GOMMAIN 5, 9
  - running GOMSVM 7
  - using GOMCMD 9
  - using GOMTDM 9
- installing
  - Operations Manager 5

## L

- legal notices
  - notices 191
  - trademarks 193
- log file
  - adding comments 126
  - management 147
- LOGTEXT command 126

## M

- managing
  - configuration files 28
  - log files 147
- MAS COMMANDS file 31
- message retrieval tool
  - LookAt xviii
- messages 165
  - methods for accessing xviii
  - processing 31
- modifying
  - startup configuration file 27
- monitoring
  - files 146
  - Operations Manager 32

- monitoring (*continued*)
  - records 146
  - virtual machines 87

## N

- notices 191

## O

- Operations Manager
  - automating sessions 35, 41
  - benefits 1
  - component interaction 3
  - components 2
  - error messages 165
  - log management 147
  - monitoring 32
  - processing
    - actions 34
    - status 64, 118
    - user interaction 2
    - what is 1
- OPMGRM1 6
- OPMGRM1 CONFSAMP file 23, 31

## P

- page space monitors
  - defining 90
  - deleting 111
- preparing
  - GOMGLBL 161
- processing
  - messages 31
  - remote file input 151
  - syslog data 149
- PROFILE EXEC 7
- PROFILE EXEC file 10
- profiles
  - FACILITY class 13
- PROFM1 SAMPEXEC file 5
- PROFS1 SAMPEXEC file 7
- PROFUSR SAMPEXEC file 9

## Q

- QUIT command 61

## R

- RACF 11
- records
  - accounting 145
  - monitoring 146
- remote file input
  - collecting 151
  - processing 151
- requirements
  - hardware 3
  - software 3
- RESUME command 114
- REVOKE command 61
- revoking
  - authority 61

- rules
  - defining 94
  - deleting 112
- RUN command 116

## S

- sample configuration file 23
  - copying 23
- schedules
  - defining 98
  - deleting 112
- security management 48
- sending
  - file content 151
  - multiple files 152
- servers
  - action processing 7, 27
- service machines
  - deleting 58
  - starting 28
- sessions
  - automating 35, 41
- setting
  - ESMPREF 12
- shutdown
  - GOMMAIN 63
- SHUTDOWN command 63
- software
  - requirements 3
- spool access commands 123
- spool files 156
  - viewing 136
- spool usage monitors
  - defining 102
- spools
  - accessing 26
- starting
  - service machines 28
- STATUS command 64, 118
- STOP command 67
- stopping
  - GOMMAIN 61
- storage definition
  - creating 46
  - deleting 57
- SUSPEND command 121
- syntax diagrams
  - how to read xvii
- syslog data
  - collecting 149
  - processing 149
  - translating 150
- system management
  - commands 31
- system options
  - defining 27

## T

- tables
  - translation 150
- tcp/ip
  - apps authority 26
- TCP/IP application
  - deleting 59

- testing
  - configuration 155
  - installation 155
- trademarks 193
- translating
  - syslog data 150

## V

- verifying
  - console data 11
- VIEWCON command 127
- viewing
  - consoles 127
  - file content 151
  - log 131
  - multiple consoles 55
  - multiple files 152
  - spool files 136
- viewing spool files 156
- VIEWLOG command 131
- views
  - defining multiple 55
  - deleting 60
- IEWSPL command 136
- virtual machine
  - remove monitoring 110
- virtual machines
  - defining 87
  - monitoring 87
- VM Tools Library web page xix





Product Number: 5697-J10

Printed in USA

SC18-9347-18

