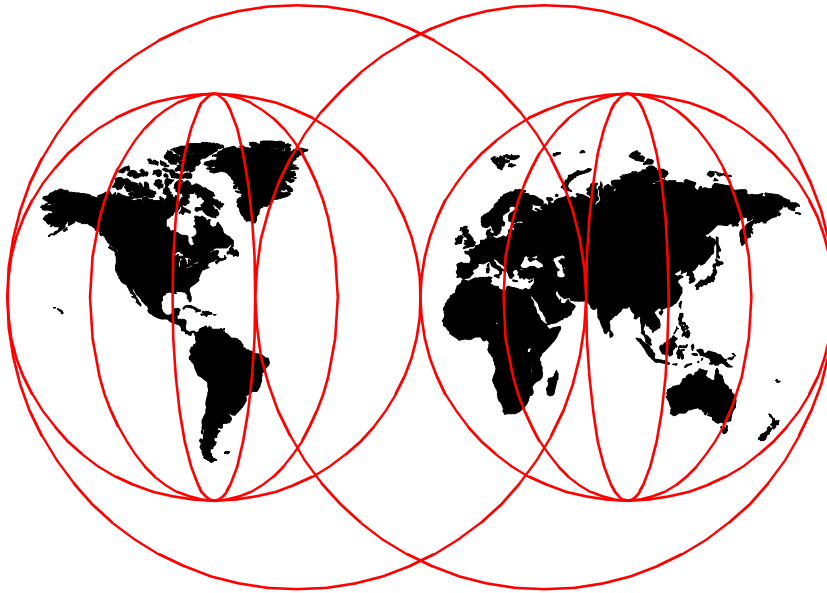


Managing AS/400 with Tivoli Enterprise

Yoichiro Ishii, Jim Coll, Russ Cunningham, Ryan Sumner, Lorraine Vassberg



International Technical Support Organization

www.redbooks.ibm.com

SG24-5517-00



International Technical Support Organization

Managing AS/400 with Tivoli Enterprise

April 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, "Special notices" on page 521.

First Edition (April 2000)

This edition applies to the levels of the respective products current at that time, in particular, 3.6 and 3.6.1. However, most of the information will be of use at both prior and later levels of the Tivoli Management Framework and applications. Where information is very specific to release level, mention of that fact is included in the text.

This document was created or updated on April 26, 2000.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. OSJB Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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

© Copyright International Business Machines Corporation 2000. All rights reserved.

Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Tables	xvii
Preface	xix
The team that wrote this redbook	xix
Comments welcome	xxi
Chapter 1. Introduction	1
1.1 Tivoli Framework and Tivoli Management Agents	1
1.2 AS/400 and Tivoli	2
1.2.1 A few words about the AS/400	2
1.2.2 Managing the AS/400 with Tivoli Enterprise	4
1.2.3 How does Tivoli fit into the picture?	6
1.3 Tivoli 3.6	9
1.3.1 Advantages of Tivoli 3.6	9
1.3.2 Coexistence of different managed resources	12
1.4 Overview of Tivoli Management Agent	12
1.4.1 TMA introduction	13
1.4.2 Tivoli Management Agent and Tivoli Desktop	17
1.4.3 Tivoli Management Agent and Command Line Interface	19
1.5 Management topology with TMA	20
Chapter 2. Tivoli for AS/400 Endpoints product information	23
2.1 Tivoli for AS/400 Endpoints package	23
2.2 Tivoli for AS/400 Endpoint products	23
2.2.1 Tivoli for AS/400 Endpoints interpreter types	25
2.2.2 Tivoli for AS/400 CD-ROM media	26
2.2.3 Tivoli for AS/400 CD-ROM media	26
2.2.4 TEC adapters for AS/400 CD-ROM	27
2.2.5 Version 3.6 of Tivoli for AS/400 CD-ROM	28
2.2.6 Version 3.6.1 of Tivoli for AS/400 CD-ROM	29
2.2.7 Endpoint patches installed on the TMR server	30
2.2.8 AS/400 Endpoint patches for the Endpoint Gateway	31
2.2.9 Tivoli support patches not available on CD-ROMs	32
2.2.10 OS/400 PTF levels	33
2.3 Related documents and information	34
2.4 Future directions	35
Chapter 3. AS/400 Endpoint installation and deployment	37
3.1 Planning for AS/400 Endpoint	37

3.1.1	Installation prerequisites	37
3.1.2	System requirements	38
3.2	AS/400 Endpoint planning considerations	42
3.3	Overview of AS/400 Endpoint installation	43
3.3.1	Endpoint Manager installation	43
3.3.2	Managed Node installation	45
3.3.3	Endpoint Gateway installation	46
3.3.4	AS/400 Endpoint installation	46
3.4	Installing AS/400 Endpoint	47
3.4.1	Using the Software Installation Service (SIS)	47
3.4.2	Using the w4inslcf.pl command	52
3.5	Removing AS/400 Endpoint software	56
3.6	AS/400 Endpoint installation comparison	58
3.6.1	Benefits of using the w4inslcf.pl script	58
3.6.2	Benefits of using the Software Installation Service (SIS).	58
3.7	Planning for AS/400 Endpoint mass installation	59
Chapter 4. AS/400 Endpoint configuration		61
4.1	Understanding the Endpoint login process	61
4.1.1	Endpoint normal login	61
4.1.2	Endpoint initial login	62
4.1.3	Region redirect	65
4.1.4	Isolated login	65
4.2	Understanding Endpoint policy	67
4.2.1	allow_install_policy	68
4.2.2	select_gateway_policy	68
4.2.3	after_install_policy	70
4.2.4	login_policy	70
4.2.5	Policy arguments for AS/400 Endpoint	70
4.2.6	Applying AS/400 Endpoint policies	72
4.3	AS/400 Endpoint configuration files	74
4.3.1	lcf.dat file	74
4.3.2	last.cfg file	74
4.4	Configuring AS/400 Endpoint	75
4.4.1	Starting AS/400 Endpoint	76
4.4.2	Comparing Endpoint options between AS/400 and others.	79
4.4.3	Stopping AS/400 Endpoint	81
4.4.4	Problems with AS/400 Endpoint	82
4.5	Customizing Endpoint policies	83
4.5.1	Example of allow_install_policy	83
4.5.2	Example of select_gateway_policy	84
4.5.3	Example of after_install_policy	84
4.5.4	Example of login_policy	86

4.6	Endpoint migration	87
4.6.1	Endpoint migration completion	89
4.7	TMR redirection	95
4.7.1	TMR redirectors	95
4.7.2	How it works	95
Chapter 5. AS/400 Endpoint considerations and tips		99
5.1	AS/400 Endpoint considerations	99
5.1.1	Using tasks on an AS/400 Endpoint	99
5.1.2	Working with Tivoli Files on the AS/400	106
5.1.3	Dataless profile manager and database profile manager	120
5.1.4	AS/400 Endpoint method management	122
5.2	Hints and tips in AS/400 Endpoint configuration	124
5.2.1	Starting the AS/400 Endpoint daemon after IPL	124
5.2.2	Using the AS/400 Endpoint Web interface	126
5.2.3	Upgrading AS/400 Endpoint daemon	133
Chapter 6. AS/400 security management		139
6.1	Overview of Tivoli security management applications	140
6.1.1	Tivoli User Administration	140
6.1.2	Tivoli Security Management	140
6.2	Tivoli User Administration for AS/400	141
6.2.1	Installing Tivoli User Administration for AS/400	142
6.2.2	Understanding Tivoli User Administration for AS/400	144
6.2.3	Customizing Tivoli User Administration for AS/400	172
6.2.4	Troubleshooting Tivoli User Administration for AS/400	176
6.2.5	AS/400 User Administration management scenarios	181
6.3	Tivoli Security Management for AS/400 Endpoints	209
6.3.1	Installing Tivoli Security Management for AS/400	210
6.3.2	Using Tivoli Security Management for AS/400	211
Chapter 7. AS/400 availability management		217
7.1	Overview of Tivoli availability management applications	217
7.1.1	Tivoli Distributed Monitoring	217
7.1.2	TEC AS/400 Event Adapter	218
7.2	Tivoli Distributed Monitoring for AS/400	219
7.2.1	Installing Tivoli Distributed Monitoring for AS/400	223
7.2.2	Understanding Tivoli Distributed Monitoring for AS/400	225
7.2.3	Customizing Tivoli Distributed Monitoring for AS/400	226
7.2.4	Troubleshooting Tivoli Distributed Monitoring for AS/400	227
7.2.5	AS/400 Distributed Monitoring scenarios	234
7.3	TEC AS/400 Event Adapters	270
7.3.1	Installing the TEC AS/400 Event Adapters	272
7.3.2	Understanding the TEC AS/400 Event Adapters	273

7.3.3 Customizing the TEC AS/400 Event Adapters	307
7.3.4 Troubleshooting TEC AS/400 Event Adapters	313
7.3.5 TEC AS/400 Event Adapter scenarios	341
Chapter 8. AS/400 deployment management	401
8.1 Overview of Tivoli deployment management applications	401
8.1.1 Tivoli Software Distribution	401
8.1.2 Tivoli Inventory	402
8.2 Tivoli Software Distribution for AS/400	402
8.2.1 Installing Tivoli Software Distribution for AS/400	405
8.2.2 Using Tivoli Software Distribution for AS/400	407
8.3 Tivoli Inventory for AS/400	421
8.3.1 Installing Tivoli Inventory for AS/400	421
8.3.2 Using Tivoli Inventory for AS/400	423
Chapter 9. AS/400 management scenarios	437
9.1 AS/400 systems environment	437
9.2 Management scenario	438
9.2.1 Financial environment scenario	439
9.3 AS/400 availability management	440
9.3.1 Subsystem management	440
9.3.2 Alerts and messages	452
9.3.3 Managing system resources	467
9.4 AS/400 user administration and security management	472
9.4.1 Tivoli User Administration for AS/400	472
9.4.2 Tivoli Security Management for AS/400	478
9.5 AS/400 inventory and deployment management	483
9.5.1 Tivoli Inventory for AS/400	483
9.5.2 Tivoli Software Distribution for AS/400	495
Appendix A. Custom scripts for AS/400 Distributed Monitoring	503
A.1 General information	503
A.2 C and C++ monitors	503
A.2.1 Example 1: FileSize.c	503
A.2.2 Example 2: FileSize.c	504
A.2.3 Example 3: FileSize.c	505
A.3 CL monitors	506
A.3.1 Example 4: CLHelp.c	506
A.3.2 Example 5: CLMONHELP.CMD	507
A.3.3 Example 6: SECONDS.CLP	507
A.4 REXX monitors	507
A.4.1 Example 7: MSGWjobs.rexx	507
A.5 QSH monitors	508
A.5.1 Example 8: FileExists.sh	508

A.6 Getting more information	509
Appendix B. AS/400 User Administration task library definition file.	511
Appendix C. w4rmusrprop	517
Appendix D. Special notices	521
Appendix E. Related publications	525
E.1 IBM Redbooks	525
E.2 IBM Redbooks collections	525
E.3 Other resources	525
E.4 Referenced Web sites	527
How to get IBM Redbooks	529
IBM Redbooks fax order form	530
Glossary	531
Index	533
IBM Redbooks review	541

Figures

1. Comparing OS/400 to other operating systems	3
2. Tivoli products history	9
3. Internationalization implementation of Tivoli 3.6	11
4. Endpoint Manager, Endpoint Gateway, and Endpoint	14
5. The Endpoint Manager Icon on the Tivoli Desktop	18
6. Three-tiered management structure	21
7. Test environment for the project	25
8. Version 3.6 of Tivoli for AS/400 CD-ROM	29
9. Version 3.6.1 of Tivoli for AS/400 CD-ROM	30
10. The Tivoli Desktop after TMR server installation completion	44
11. Create Gateway panel	46
12. Add machine dialog when adding AS/400 into SIS	48
13. Selecting Endpoint product within SIS	49
14. Installing the Details dialog within SIS	49
15. Machine attributes for Tivoli for AS/400 Endpoint within SIS	50
16. Endpoint initial login	62
17. Endpoint initial login processes	63
18. Endpoint initial login	64
19. Endpoint isolated login	66
20. Endpoint policies	67
21. Gateway selection and new login interfaces	69
22. New login interfaces from Endpoint Manager	70
23. Endpoint Gateway migration	88
24. Migration completion by migratory login	90
25. Migration completion by migratory upcall	92
26. Migration completion by migratory downcall	94
27. TMR redirection	96
28. AS/400 task library	101
29. Editing AS/400 task	102
30. Executing AS/400 task	103
31. Configuring AS/400 task arguments	104
32. The command task output from AS/400 (shaft)	104
33. The command task output from AS/400 (kong)	105
34. Grant Object Authority	108
35. Start TMA Endpoint (STRTMEEPT)	108
36. Display Physical File Member (LASTCFG)	109
37. Create Physical File (LCFDLOG)	110
38. Copy From Stream File (lcfld.log to LCFDLOG)	111
39. Display Physical File Member (LCFDLOG)	112
40. Work with All Spooled Files: Option 8=attributes	114

41. Work with Spooled File Attributes	115
42. FTP from AS/400 to UNIX (1 of 2)	117
43. FTP from AS/400 to UNIX (2 of 2)	118
44. FTP from UNIX to AS/400 (1 of 2)	119
45. FTP from UNIX to AS/400 (2 of 2)	120
46. Difference between dataless and database profile managers	121
47. Endpoint method cache management	123
48. AS/400 Endpoint Web interface	127
49. Location Configuration	129
50. Reconfiguring Endpoint option setting	131
51. Password dialog box	131
52. AS/400 Endpoint reset message	132
53. Reconfiguration result	133
54. Using Endpoint login_policy for upgrading AS/400 Endpoint daemon. . .	137
55. OS/400 auditing attributes	146
56. OS/400 directory attributes	149
57. OS/400 display session attributes	150
58. OS/400 group membership attributes	152
59. OS/400 international attributes	153
60. OS/400 login attributes	155
61. OS/400 output attributes	156
62. OS/400 password attributes.	158
63. OS/400 privileges attributes.	159
64. OS/400 session startup attributes	161
65. Create user profile (1 of 4)	164
66. Change user profile (page 1 of 4)	165
67. Change user profile (page 2 of 4)	166
68. Change user profile (page 3 of 4)	166
69. Change user profile (page 4 of 4)	167
70. Change user auditing.	169
71. Delete user profile	170
72. AS/400 User Administration task library	173
73. Work with spooled files	177
74. Work with job queues.	178
75. UMBO_SKEL1 joblog (page 1 of 4)	179
76. UMBO_SKEL1 joblog (2 of 4)	180
77. UMBO_SKEL1 joblog (3 of 4)	180
78. UMBO_SKEL1 joblog (4 of 4)	181
79. Profile distribution flow	183
80. AS/400 User profile properties.	188
81. OS/400 login attributes	189
82. OS/400 privileges.	190
83. OS/400 session startup	191

84. Distributing the user profile	192
85. Display user profile (1 of 2)	193
86. Display user profile (2 of 2)	193
87. The Add_Directory_Entry task execution.	194
88. Configuring Add_Directory_Entry task arguments.	195
89. Add_Directory_Entry task output (1 of 2).	196
90. Add_Directory_Entry task output (2 of 2).	197
91. The Remove_Directory_Entry task execution	198
92. Configuring Remove_Directory_Entry task arguments	198
93. Remove_Directory_Entry task output	199
94. The Display_User_Profile task execution	200
95. Configuring Display_User_Profile task arguments	201
96. Display_User_Profile output	202
97. The Change_Object_Ownership task execution	203
98. Configuring Change_Object_Ownership task arguments	204
99. Change_Object_Ownership output	205
100.OS/400 login attributes	206
101.Deleting BDAVIS user record	206
102.User profile with the DBAVIS record removed	207
103.Distributing the user profile	208
104.Work with user profiles	209
105.Creating Security profile	212
106.Populating the security profile	213
107.Security group information	214
108.Security system policy information	215
109.Edit using Source Entry utility	229
110.Work with data in a file	230
111.End Data Entry	231
112.Selecting the profile manager on Tivoli Desktop	235
113.Creating a Distributed Monitoring profile	236
114.The Distributed Monitoring profile	237
115.Selecting a monitoring collection and monitoring source	238
116.Configuring the Distributed Monitoring monitor	239
117.Setting monitoring schedule	240
118.Saving configured Distributed Monitoring profile.	240
119.The confirmation screen for saving profile	241
120.Setting subscribers for the monitors	241
121.Create source physical file	244
122.Work with members using PDM	245
123.Start source entry utility.	246
124.Sample STRLPVMADP program	247
125.SEU exit panel.	248
126.Compiling the STRLPVMADP program.	249

127.Display library	250
128.Work with Active Jobs: LPVMSGADP job running	251
129.Work with Active Jobs: LPVMSGADP not running	252
130.Work with Active Jobs: LPVMSGADP job restarted	253
131.Display QSYSOPR messages	254
132.OS/400 Job exists monitor	255
133.Job monitor responses	256
134.Work with Active Jobs: LPVMSGADP job running	257
135.Job monitor popup response	257
136.DMEPENGINE joblog	258
137.Edit default policies: Set Remote User ID	259
138.Editing the job exists monitor	260
139.PopUp job monitor response	261
140.Work with active jobs: LPVMSGADP job restarting	261
141.DMEPENGINE joblog showing SBMJOB response	262
142.The work with job panel	263
143.Creating the OS/400 subsystem monitor	264
144.Editing the number of jobs monitor	265
145.Work with Active Jobs	266
146.Display Messages	267
147.Selecting Universal monitoring collection	268
148.Configuring custom monitor	269
149.Setting user and group ID	270
150.TME events and non-TME events	272
151.Work with message descriptions	275
152.Display message description (message text)	276
153.Display message description (field data)	277
154.Display message description (message attributes)	278
155.Send message (1 of 2)	279
156.Send message (2 of 2)	280
157.Display message (QSYSOPR)	281
158.Additional message information	282
159.Display message description for CPA58E0	290
160.Display network attributes	293
161.Work with Alert Descriptions	295
162.Display Alert Detail	296
163.Display Recommended Actions	297
164.Alert Filter mechanism	298
165.The LPVFTP Alert Filter mechanism	300
166.STRLPVMADP program source	309
167.Display Software Resources	314
168.STRTECADP command resulting in an error message	316
169.Additional Message Information (TEA0021)	317

170.Display All Messages (STRTECADP error)	318
171.Additional Message Information (TAA0026)	319
172.Joblog of a successful STRTECADP command	323
173.WRKACTJOB showing LPVMSGADP job active	324
174.Display All Message (LPVMSGADP)	325
175.Configuration file specifying TestMode	327
176.DSPMSG MSGQ(QSYSOPR) test messages	328
177.DSPPFM FILE(LPVLIB/MSGEVENTS).	329
178.WRKLNK of /QIBM/UserData/Tivoli/TEC/MSGQ/Cache directory	330
179./QIBM/UserData/Tivoli/TEC/MSGQ/Cache/LPVMSGADP attributes	331
180.ANZPRP: Select Type of System	333
181.ANZPRB: Analyze a Problem	334
182.ANZPRB: Problem Frequency	334
183.ANZPRB: Define Resource	335
184.ANZPRB: Enter Problem Description	336
185.ANZPRB: Save Problem Data.	336
186.ANZPRB: Report Problem.	337
187.ANZPRB: Define Message Text for Alert	338
188.Work with Alerts	339
189.Display Alert Detail (page 2 of 2).	340
190.Work with Objects Using PDM (LPVLIB/LPVMA*)	342
191.Work with Members Using PDM (LPVLIB/LPVMACFG).	343
192.Source Entry Utility (MSGCFG member)	344
193.Source Entry Utility Exit (MSGCFG)	345
194.STRTECADP LPVMSGADP	346
195.ENDTECADP LPVMSGADP	347
196.Work with Objects Using PDM (LPVLIB/LPVMA*)	349
197.Work with Members Using PDM (LPVLIB/LPVMACFG).	350
198.Editing MSGCDS file (1 of 2).	350
199.Editing MSGCDS file (2 of 2).	351
200.View message CPF0995 (left) and CPF1103 (right)	352
201.View Message (CPA7025)	353
202.Send Reply (SNDRPY) command and keywords	354
203.Method 1: CPA7025 inquiry message.	355
204.SNDRPY command issued in an FTP session	355
205.Method 2: CPA7025 inquiry message.	356
206.Method 2: Execute task on selected event	357
207.Method 2: TEC tasks (1 of 2)	358
208.Method 2: TEC tasks (2 of 2)	359
209.Method 2: Select the Managed Nodes	359
210.Method 2: Task output (1 of 2)	360
211.Method 2: Task output (2 of 2)	361
212.Method 3: Edit LPVLIB/LPVMACDS member MSGCDS	362

213.Method 3: Execute task on selected event	363
214.Method 3: TEC tasks (1 of 2)	364
215.Method 3: TEC tasks (2 of 2)	364
216.Method 3: Task output (1 of 2)	365
217.Method 3: Task output (2 of 2)	366
218.Method 4: Execute task on selected event	368
219.Method 5: Execute task on selected event	370
220.Work with Message Descriptions	372
221.Add Message Description (ADDMSGD)	373
222.Filling out the Add Message Description panel	374
223.Specifying more values for parameter format	375
224.The TIV0001 message description	376
225.Editing the program that retrieves the job attributes	377
226.Editing the TIVADMCFG file	378
227.Displaying the TIVADM message queue.	379
228.Browsing TEC event	380
229.Setting task execution privileges	381
230.Executing the LPV_Get_UA_Joblog task	382
231.WRKMSGD MSGID(*FIRST) MSGF(LPVLIB/TIVINVMMSGF)	383
232.Source of SNDINV0003 program	384
233.The AS/400 Screen Design Aid (SDA)	385
234.The design menus panel.	386
235.The specific menu functions panel	387
236.The TIVINV menu	388
237.The define menu commands panel.	389
238.The specify menu functions panel.	390
239.The exit SDA menus panel	391
240.The TIVINV menu after compile	392
241.Displaying the TIVINV message queue.	393
242.Editing the TIVINVCFG configuration file	394
243.Creating automated task	396
244.Setting up the automatic task execution	397
245.Saving Automatic task execution setup.	398
246.Requesting PTF update scan	399
247.Inventory scanning request job	400
248.File package block components	405
249.Display library of QTMELCF	407
250.Creating a profile manager for Tivoli Software Distribution	408
251.Creating a file package in database profile	409
252.Configuring file package properties.	410
253.Configuring file package options for an OS/400 platform	411
254.Starting Tivoli Object Packaging Utility for AS/400	415
255.AS/400 Client Access and Tivoli Object Packaging utility for AS/400	416

256.Tivoli Object Packaging Utility for AS/400	417
257.Selecting the files to be packaged.	418
258.Create Tivoli Filepack dialog	419
259.Keywords setting dialog box	420
260.Tivoli Inventory operations	424
261.Creating Inventory profile	425
262.Configuring Inventory profile	426
263.Distributing Inventory profile	428
264.Query libraries for OS/400 software information	429
265.Using Tivoli Inventory on Tivoli Desktop	430
266.OS/400 software inventory query	431
267.OS/400 software query results	432
268.Inventory profile for scanning AS/400 PTFs	433
269.OS/400 PTFs query results	434
270.Tivoli Inventory queries	435
271.Our Tivoli management environment	439
272.Creating a Distributed Monitoring profile	442
273.Distributed Monitoring Properties	442
274.Choosing a monitor collection and source	443
275.Selecting the monitor severity level.	444
276.Selecting popup message recipients.	444
277.Using the Run Program monitor option	445
278.Selecting the TEC server	445
279.Completed monitor definition.	446
280.Setting monitoring schedule	447
281.Choosing custom times and days	448
282.Customizing the QINTER monitor schedule	449
283.QINTER monitor in the DM_AS/400 profile properties	450
284.Distributing the DM_AS/400 profile	450
285.Tivoli Desktop popup message	451
286.TEC events generated by DM_AS/400 QINTER subsystem monitor	451
287.Message Adapter configuration file - Defining a message queue	454
288.Message Adapter, FINMSGADP, running in QSYSWRK	455
289.Viewing AS/400 message adapter events in TEC.	456
290.AS/400 Task Library	457
291.Selecting endpoint on the Execute Task screen	458
292.Saving a task in the AS/400 Task Library	459
293.Select Event Class screen	460
294.Editing Automatic Task execution criteria	461
295.Select an automated task	462
296.Select Managed Nodes for automated task	462
297.Select nodes for automated task execution	463
298.Completed automated task	464

299.Configuring Alert Adapter filters	466
300.AS/400 Alerts console	467
301.Select the OS/400 Disk monitor	468
302.Configuring a Warning AS/400 Disk Monitor	469
303.Configuring a Critical AS/400 Disk Monitor	470
304.Monitor Profile Properties screen	471
305.Distributing the AS/400 Disk Monitor.	471
306.Creating a new User Administration profile	473
307.Populating the User Administration database	474
308.Select a default user profile attribute for OS/400	475
309.Generated defaults for AS/400 user profiles	476
310.Viewing AS/400 user profiles	477
311.Distributing new AS/400 profiles	477
312.Creating a Security Management profile	479
313.Security Profile screen	479
314.Populating the Security Management database	480
315.Security System Policy records	481
316.Editing the OS/400 login policy	481
317.Distribute security policy	482
318.Creating an inventory profile and subscribers	484
319.Customizing inventory scans.	485
320.Distributing inventory profiles.	486
321.Inventory queries library	487
322.Inventory queries for the AS/400.	487
323.AS/400 hardware query.	488
324.AS/400 hardware query results	489
325.Creating a new inventory profile for AS/400 PTFs	491
326.Customizing an inventory profile for an AS/400 PTF scan	493
327.AS/400 PTF query.	494
328.Creating a software distribution profile	497
329.Software distribution properties.	498
330.Specifying a Before Distribution program	499
331.After Distribution program	500

Tables

1. OS/400 versions supported by Tivoli	6
2. Tivoli for AS/400 Endpoints product summary	8
3. Equipment used for the project	24
4. Tivoli interpreter types supported on the AS/400	25
5. Products and patches required for the TMR server installation	30
6. Products and patches required on EP Gateway for AS/400 Endpoints	31
7. Support patches available for AS/400 Endpoints	32
8. Needed disk space for AS/400 Endpoint	39
9. Required patches to upgrade AS/400 Endpoint to Version 3.6.1	42
10. AS/400 commands authorization	76
11. AS/400 Endpoint (STRTMEEPT command) options	77
12. Comparing Endpoint options between AS/400 and UNIX	79
13. AS/400 Endpoint task library	99
14. Reconfiguring the Endpoint login_interval option	130
15. Endpoint version and Tivoli Management Framework patch	134
16. OS/400 auditing attributes	146
17. OS/400 directory attributes	149
18. OS/400 display session attributes	151
19. OS/400 group membership attributes	152
20. OS/400 international attributes	153
21. OS/400 login attributes	155
22. OS/400 output attributes	157
23. OS/400 password attributes	158
24. OS/400 privileges attributes	159
25. OS/400 session startup attributes	161
26. Tivoli products in our scenarios	438
27. AS/400 test environment	438
28. Suggested Subsystem Monitors and actions	452
29. User Profile Default attributes	473
30. Default User Profile Attributes	478
31. Database type for AS/400 PTF script	490

Preface

With Tivoli Enterprise 3.6, customers can extend their Tivoli implementation to support AS/400s as Endpoints. Sales and service departments, business partners, and customers need an AS/400-specific redbook to complement the platform-neutral redbook, *An Introduction to Tivoli Enterprise*, SG24-5494. Our redbook will be used as a primary source of information when starting a Tivoli Enterprise implementation for AS/400 and will function as a field reference providing technical information and answers for sales situations.

This redbook meets the needs of two audiences:

- Those who know Tivoli but do not know the AS/400
- Those who know the AS/400 but do not know Tivoli

This redbook directs readers to existing Tivoli redbooks; so, they can learn concepts and generic Tivoli Enterprise items. This redbook documents and provides examples of the implementation techniques related to managing AS/400s as Endpoints. It includes simple examples for getting started and more complex examples of the use of Tivoli Enterprise that are relevant to the AS/400.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization.

Yoichiro Ishii is an Advisory I/T Specialist working as a project leader at the International Technical Support Organization, Tivoli Group, Austin Center. He manages Tivoli projects and develops technical documents including redbooks on Tivoli Enterprise and application management topics. He also teaches IBM classes worldwide on all areas of systems management, application management, and network management. Before joining the ITSO in mid-1998, Yoichiro worked at IBM Japan as an I/T Specialist for systems and network management consulting major IBM customers. In this role, he architected systems management designs and solutions for large government projects. He can be reached at ishii@us.ibm.com.

Jim Coll is a Technical Support Specialist at the IBM Technology Campus in Dublin, Ireland. He has over two years experience in AS/400 support. He holds a degree in Computer Applications from Dublin City University. His areas of expertise include AS/400 technical support as well as AS/400 application support in Financial and Manufacturing environments.

Russ Cunningham is a Staff Engineer in Austin Texas. He has 24 years of experience in the computer field having done hardware and software design with over 10 years in support. He has worked at IBM for eight years (contract and Tivoli). His areas of expertise include AIX and other UNIX flavors, Framework, and Distributed Monitoring.

Ryan Sumner is a UNIX System Administrator for Tivoli Systems. He has six years experience administering UNIX and developing software applications. Ryan spent two years performing installations and deployments of Tivoli software in customer environments. His expertise is in assisting customers to implement Tivoli on AS/400 servers and diverse UNIX environments. Ryan has written extensively on the installation, deployments, and configuration of AS/400 Endpoints.

Lorraine Vassberg is a Senior Software Engineer in Level 2 support with Tivoli Systems, Austin. She has 10 years experience working in Level 2 support for IBM. Prior to joining the Tivoli organization, she supported the OfficeVision/400 product at the Westlake Programming Lab and IBM SystemView products in the Research Triangle Park. She specializes in OS/400 and has presented several OV/400 topics at COMMON. She has degrees in Mathematics and Physical Education from Austin College and a masters degree in Computer Science from NTU.

Thanks to the following people for their invaluable contributions to this project:

David Sandifer
IBM Rochester

Rik Lammers
IBM Netherlands

Brian Schlosser, Michael Lugton, Stefan Uelpenich
Tivoli Systems

Theo Winkelmann
International Technical Support Organization

Edson Manoel
International Technical Support Organization

Suehiro Sakai
International Technical Support Organization

Milos Radosavljevic
International Technical Support Organization

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 541 to the fax number shown on the form.
- Use the online evaluation form found at <http://www.redbooks.ibm.com/>
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. Introduction

With Version 3.6.x, Tivoli Enterprise now supports the AS/400 platform. We assume that this redbook will be of particular interest to those who work with or support AS/400 systems and Tivoli Enterprise products. In this redbook, we work with products from the Tivoli Enterprise software family and focus on AS/400 support.

We will not go into much detail about the concept of Tivoli or Tivoli Enterprise in this redbook. We strongly recommend that you refer to the following documentation:

- *All About Tivoli Management Agents*, SG24-5134
- *Tivoli Enterprise Performance Tuning Guide*, SG24-5392
- *An Introduction to Tivoli Enterprise*, SG24-5494
- *Tivoli Enterprise Internals and Problem Determination*, SG24-2034

These redbooks should provide detailed information about Tivoli and Tivoli Enterprise products.

1.1 Tivoli Framework and Tivoli Management Agents

The Tivoli Management Framework provides a set of common services and facilities that enable powerful systems management applications. This framework provides benefits to developers who want to take advantage of services and facilities that hide the complexity of the networking environment. By doing so, the Tivoli Management Framework allows the developer to concentrate on developing solutions that apply across a wide range of operating environments.

Likewise, the Tivoli Management Framework is valuable to those responsible for managing complex environments because it provides common user interface elements and hides differences in the operating environments of managed systems.

The Tivoli Management Framework is based on industry standards, such as the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA), and has wide acceptance with a large number of system management application developers. Designing a Framework, such as the Tivoli Management Framework, requires meeting two, sometimes conflicting, criteria: Stability and extensibility.

Stability gives application developers confidence that the applications they develop will continue to run when new versions of the Framework become available. Extensibility provides customers with the knowledge that the Framework can evolve over time to meet their changing requirements.

Version 3.2 of Tivoli Management Framework introduced major new extensions to the framework's architecture. These new extensions included the Lightweight Client Framework (LCF) architecture. This version of the Framework is a testament to both the stability and extensibility of the Tivoli product's architecture. Although, in Version 3.2, the framework supported these extensions, few Tivoli Management Applications took advantage of them until their next release, which was Version 3.6.

Now that Version 3.6.1 of Tivoli has been released and the applications are taking full advantage of the new architecture extensions, we have an even more powerful set of functions and services for distributed systems management. Version 3.6 of Tivoli Management Framework and applications can be installed and configured in the same way as previous versions. In addition, applications can now take advantage of new client types that run a component called the Tivoli Management Agent (TMA). Utilizing the TMA with Version 3.6 products a new level of extensibility and allows you to scale your management across the entire enterprise.

With Version 3.6 of Tivoli, the Tivoli Management Framework and selected Tivoli applications support AS/400 systems as Tivoli Management Agents in a Tivoli Management Region (TMR).

This redbook describes in detail the features and services provided by the Tivoli for AS/400 Endpoints. Solutions and examples of using the Tivoli for AS/400 Endpoints are also provided.

1.2 AS/400 and Tivoli

What is AS/400? It is the world's most popular multi-user business computing system; it is sold in more than 120 countries, enabled in 51 national languages, and installed in 98 percent of Fortune magazine's list of the top 100 companies. The success of the AS/400 is mainly due to its low total cost of ownership and high reliability and scalability.

1.2.1 A few words about the AS/400

The AS/400 is designed with an integrated operating system that delivers a relational database, security, communications, systems management, and many other functions as part of the base operating system. On other

platforms, the operating system provides no more than basic functionality, and you have to add additional products from other vendors to support communication, security, and system management functions. The user is responsible for the integration of all these products initially and also when upgrading or changing. Quite often, this creates a need for all kinds of specialist or outside services. This places a heavy burden on the cost of computing.

Integration is the strength of the AS/400, and this is one of the reasons that the overall complexity of system operations and management is greatly reduced. IBM provides this integration by utilizing the unique architecture of the AS/400. Third party vendors of additional AS/400 applications can also make use of the AS/400's integrated approach. For the customer, this means shorter development/testing lead-time. When the customer decides to upgrade and expand their AS/400 hardware and software, the third party applications will continue to work unchanged. They inherit the continuing improvements of the AS/400 without any need for recompilation or rewriting. Therefore, a typical organization powered by AS/400 technology has a relatively small I/T department. These staff members tend to specialize in AS/400 as a whole, not just on subsections, such as database design, communications, performance management, and so on. See Figure 1 for a comparison of OS/400 to other operating systems.

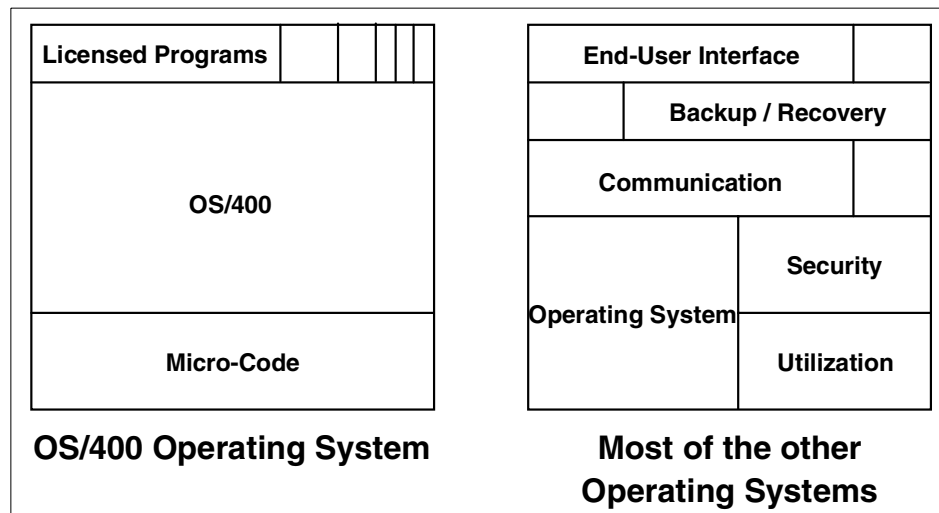


Figure 1. Comparing OS/400 to other operating systems

The AS/400's design allows it to be easily upgraded without affecting your existing applications. Some operations can even be performed concurrently,

thereby, minimizing downtime. The scalability of the system is mainly due to the microcode layer of the AS/400. Refer to Figure 1 on page 3, which provides customers with a hardware-independent interface to develop their applications.

High availability is one of the key features of the AS/400. According to surveys, it is estimated that the average AS/400 has less than nine hours of unplanned downtime a year. That corresponds to 99.94 percent availability and ranks it as one of the most stable and reliable platforms in the world. IBM is continually improving the operating system with regular updates. These fixes are available through PTFs that can be downloaded as needed. Keeping your system up to date with the latest PTFs, coupled with the AS/400's backup and recovery functions, significantly decreases the chance of unplanned downtime.

Today, computer system security is vital and serves to protect your information assets from unauthorized and unintentional access, thus, minimizing the risk to your business. The AS/400 provides integrated operating system security with many functions that allow you to implement the security policy that is right for you. There are three basic layers that enable you to administer operating system security:

- System-wide security policies
- Object level security
- User profiles

Combining these three layers results in a highly-configurable security system in which you can give users as little or as much authority as you wish.

With the advent of e-commerce, the AS/400 has taken the stage to provide a series of new technologies in order to support and develop businesses in the electronic age. The AS/400 is now fully Web-capable and supports the Java programming environment to develop advanced Web-based applications. IBM has also released the new Dedicated Server for Domino (DSD). Now, all your Lotus servers can run on customized platforms and take full advantage of AS/400 stability and scalability.

1.2.2 Managing the AS/400 with Tivoli Enterprise

The proper management of an AS/400 should result in a fast, efficient, and reliable system. It should be secure, utilize auditing capabilities (if necessary), and have rigorous backup and recovery procedures in place to minimize data loss and downtime if something should go wrong. Tivoli products for AS/400 introduce a new approach to managing your AS/400.

They bring new functionality and integration to enhance your existing support service.

The typical day-to-day process of managing an AS/400 environment can usually be classed into two main sections: System administration and system operations.

Systems administration involves the following:

- Implementing the security policy of an organization
- Setting up and maintaining user and group profiles
- Installing software (vendor software, operating system upgrades, and so on)
- Security auditing and reporting
- Creating and configuring new devices, such as new network lines, printer devices, and servers
- Maintaining and ordering the latest versions of integrity and security PTFs
- Planning for future software and hardware needs

System operations deal with the daily tasks of running the AS/400, such as:

- Ensuring that the machine is functioning properly and available for use (checking subsystems and background processing jobs)
- Backups and general housekeeping (cleaning up job logs, removing old spoolfiles, and clearing message queues)
- Monitoring the system resources and performance of the machine (checking job queues, active jobs, CPU usage, printers, and disk utilization)
- Providing general user support for AS/400-related issues.

An organization may have a number of AS/400s installed supporting thousands of users. The AS/400 may also interface with numerous other devices, such as printers, communications devices, such as paging systems, and other computer systems. The AS/400 is, arguably, one of the easiest platforms to maintain because of its integrated systems management approach.

However, as networks grow and computing needs become more complex and demanding, the need for centralized management becomes more apparent. A computing platform can no longer be seen as a stand-alone system if it is to be utilized in an efficient and cost-effective manner. Tivoli exploits the concept of central management. It controls the basic and advanced tasks of

computer system management in a large organization from one central console. Tivoli gives you *The Power to Manage. Anything. Anywhere.*

1.2.3 How does Tivoli fit into the picture?

Tivoli produces a suite of products specifically designed to manage heterogeneous distributed computer systems. Now, Tivoli has extended its array of supported platforms to incorporate OS/400. It does this through the use of a customized version of the Tivoli Endpoint product and the Tivoli Enterprise Console (TEC). Tivoli provides the ability to manage your computer systems, including the AS/400, from a centralized point.

The Tivoli for AS/400 Endpoints support the following Tivoli applications:

- Tivoli Security Management
- Tivoli User Administration
- Tivoli Distributed Monitoring
- Tivoli Enterprise Console Event Adapter
- Tivoli Inventory
- Tivoli Software Distribution

Note

In the current version of Tivoli Enterprise Console, Adapters for AS/400 work as non-TME adapters. Refer to Section 7.3, “TEC AS/400 Event Adapters” on page 270, for more detailed information.

Tivoli for AS/400 Endpoints are available for both CISC and RISC models of the AS/400. Table 1 lists the supported versions of OS/400.

Table 1. OS/400 versions supported by Tivoli

AS/400 system type	OS/400 version
CISC	V3R2
RISC	V3R7
	V4R1
	V4R2
	V4R3
	V4R4

Tivoli also produces a product called Tivoli Enterprise Console (TEC). This is a separate product and allows you to monitor messages and alerts on the AS/400. The AS/400 Alert Adapter and the AS/400 Message Adapter collect events from an AS/400 system and forward them to the Tivoli Enterprise Console event server. TEC allows you to filter non-important alerts and messages leaving only the critical events. You can also create rules for monitoring these events and configure automatic responses for any chosen event.

Using the Tivoli tools can help you centralize your AS/400 support and management tasks. Instead of duplicating work for each AS/400 under your support, Tivoli brings them all together into a Tivoli Managed Region (TMR). With the Tivoli for AS/400 Endpoints and the TEC Event Adapters for AS/400, you can centrally-manage all your system operations:

- Tivoli Inventory can investigate all the AS/400 systems in a TMR and compile a list of installed hardware and software. This gives you the ability to determine which systems are running earlier releases of software and what PTF levels are installed.
- Tivoli Software Distribution automatically distributes, installs, and configures software or database packages. For example, upgraded applications and PTFs can be automatically distributed to all your AS/400s from one central source.
- Tivoli User Administration and Tivoli Security Management enable you to manage all AS/400 user profiles and AS/400 system security. The system also comes complete with its own auditing capabilities.
- Tivoli Distributed Monitoring lets you monitor AS/400 system performance and system resources. Distributed Monitoring can be configured to send events to the TEC server, e-mails to AS/400 operators and administrators, or even text messages to a paging service. You can also create automatic responses to act upon certain events, for example, to restart a background system job if it has ended.
- Tivoli Enterprise Console monitors alerts and messages on the AS/400 and forwards them to the TEC event server. The Message Adapter consolidates your system (QSYSOPR) and application message queues for all your AS/400s in the TMR. The Alert Adapter monitors the AS/400 alert adapters for all your AS/400s. The TEC sever can filter out unimportant messages and alerts leaving only the events that are critical to your operations. TEC also allows you to create rules and actions to apply to these events.

Table 2 summarizes the Tivoli solutions for AS/400 with a brief description and notes the key strengths of each solution.

Table 2. Tivoli for AS/400 Endpoints product summary

Tivoli product	Description	Key strengths
Tivoli Inventory	Collect information about installed hardware, software, and PTFs	Maintains a complete inventory of all hardware, software, and PTFs resources
Tivoli Software Distribution	Distribute new software and PTFs	Provides simple distribution of software and PTFs. Helps maintain consistency of software versions across your network
Tivoli User Administration	Manage user profiles and groups	Centralizes control of user administration
Tivoli Security Management	Manage and monitor your AS/400 security	Simplifies security management. Helps to standardize security policies across your A/400 network
Tivoli Distributed Monitoring	Manage system resources and performance. Create automated responses for particular events	Allows creation of standard/customized monitors to manage your AS/400 resources
Tivoli Enterprise Console (TEC)	Monitor critical system and application alerts and message queues. Filters non-important events and issues automatic responses for critical events	Monitors all AS/400 events from a central console. Integrates with others computer systems on your network

The Tivoli products for the AS/400 have a very small storage footprint; so, they will not take up large amounts of valuable data storage space on your systems. They include many advanced functions that can be tailored to meet the needs of your AS/400 environment. The standard Tivoli graphical user interface allows you to perform most of its functions with a click of the mouse.

Now, all your computing systems, AS/400, UNIX/AIX, and S/390, to name but a few, can be managed from a central source using a standard interface and a common environment.

1.3 Tivoli 3.6

Version 3.6 of Tivoli Enterprise products are an extremely strategic set of products. From now on, we will refer to this set of products as Tivoli 3.6. Tivoli 3.6 encompasses almost all features and services that were provided by the previous version of the Tivoli products and also extends these features. Tivoli 3.6 provides real extensibility and flexibility for customers.

As we mentioned, LCF architecture was available in Version 3.2 of Tivoli Management Framework. However, there were few applications that could support the LCF Endpoint. In Tivoli 3.6, all Tivoli Management core applications support the LCF architecture and can run on TMA machines. Figure 2 shows the Tivoli products history.

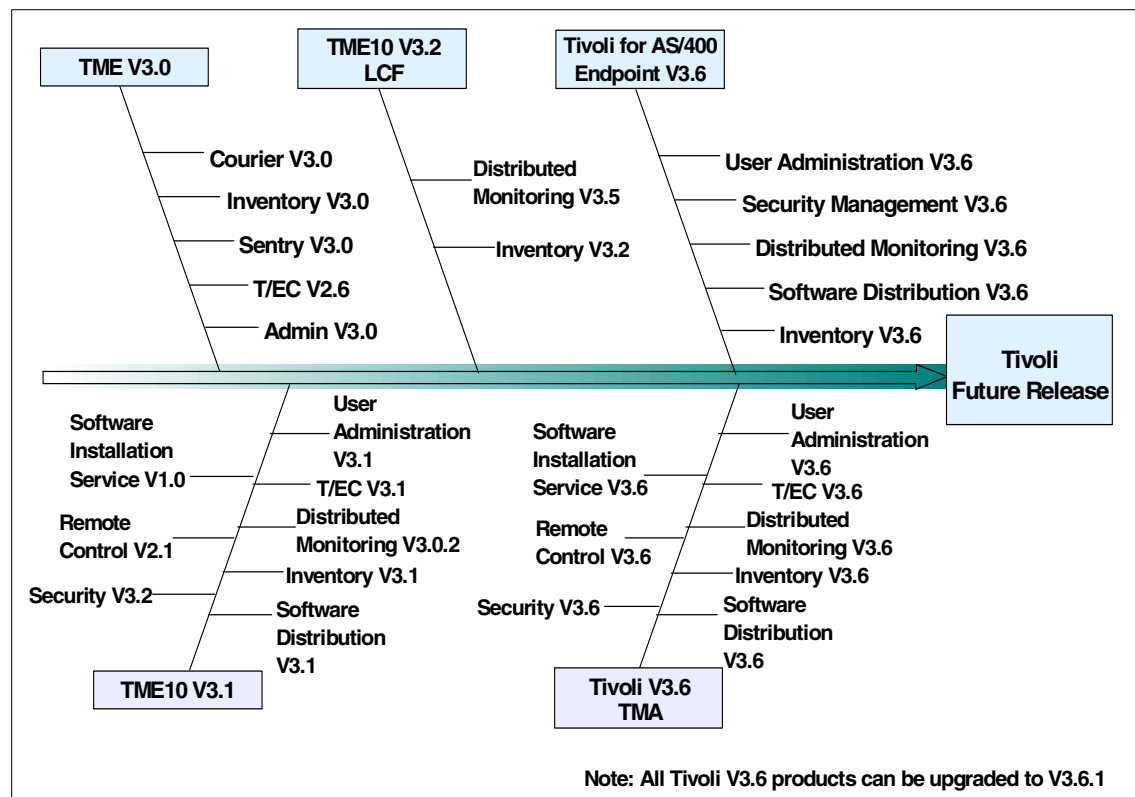


Figure 2. Tivoli products history

1.3.1 Advantages of Tivoli 3.6

The most significant enhancements across the Tivoli 3.6 product set include:

- Support for additional platforms
- Providing TMA support for applications
- Internationalization

1.3.1.1 Supporting many platforms

Tivoli 3.6 supports many platforms with the TMA. The following are the main platforms Tivoli 3.6 TMA supports:

- UNIX
- Windows NT
- Windows 98, 95, 3.x
- NetWare 3, 4
- OS/2
- AS/400
- OS/390

As you can see, Tivoli 3.6 supports many platforms, from the PC to the mainframe. Once the Tivoli Management Framework is installed, the customer can manage, using the Tivoli Desktop interface and the Command Line Interface (CLI), all system types running the TMA or other Tivoli client software with a single operation. This is because the Framework and applications provide platform independence. This means the type of target on which a management operation is to be performed is transparent to the administrator or program initiating the operation. Therefore, seamless operations become available for multiple platforms in the Tivoli 3.6 environment. This platform independence is one of the many benefits provided by the Tivoli architecture.

1.3.1.2 Internationalization

National language support is one of the most important features for customers who do not use English because the user interface is very important in systems management software. Tivoli 3.6 provides the internationalization feature using the implementation illustrated in the Figure 3 on page 11.

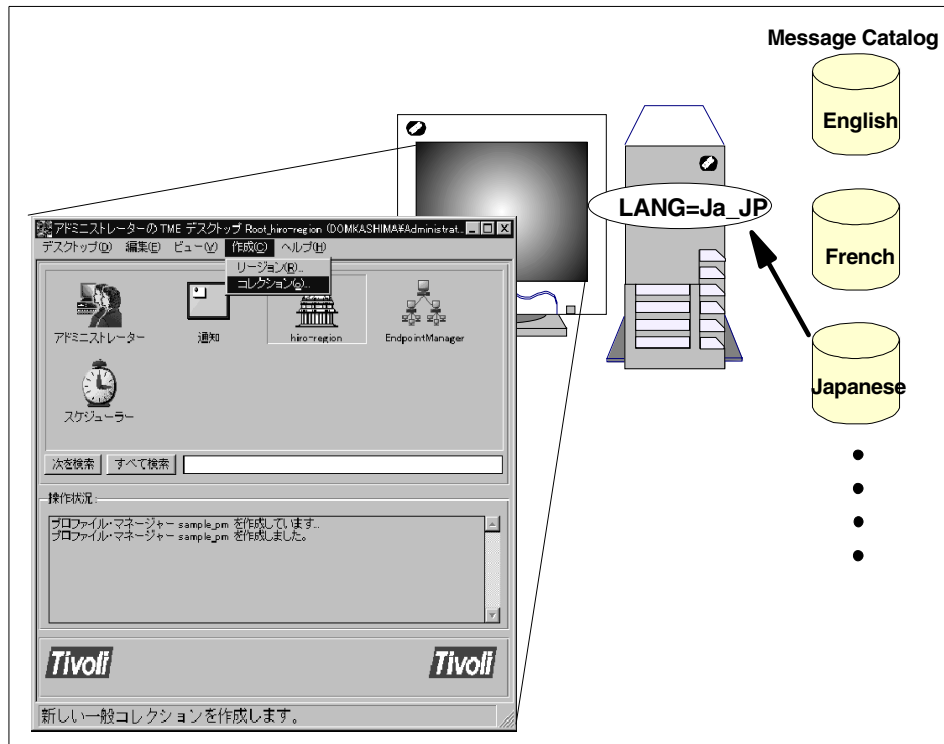


Figure 3. Internationalization implementation of Tivoli 3.6

Previous versions of Tivoli had separate source code for each supported non-English language. Prior to the internationalization implementation, the Tivoli products contained hardcoded message catalogs in their source files. Keeping the source code across all languages at the same level was difficult, at best. As a result, patch modules were needed for each language version. When these versions were not kept synchronized, it caused a lot of confusion for the customer.

The internationalized Tivoli 3.6 has one common set of source code with many language code sets. This means that supporting other languages is easier and faster than with previous versions of Tivoli. This implementation of the internationalization feature is similar to the implementation of the IBM AIX operating system. For example, Figure 3 shows how to load the Japanese language code set. In this case, we simply set the system language environment to Japanese, and Tivoli 3.6 displays the Tivoli Desktop with Japanese characters. All messages are built into message catalogs that are maintained separately from the source code.

Tivoli 3.6.1 presently supports the following languages:

- Chinese (Simplified)
- English
- French
- German
- Italian
- Japanese
- Korean
- Brazilian Portuguese
- Spanish

Note

To enable these languages, you must install the appropriate language support pack from the TME 10 Framework Language Support CD-ROM.

1.3.2 Coexistence of different managed resources

Although the Tivoli Management Agent provides a powerful interface for managing a wide variety of systems, Tivoli 3.6 still supports the following types of managed resources used in previous versions and keeps consistency among these different managed resources.

- Managed Node
- PC Managed Node
- NetWare Managed Site

TMA replaces the above managed resources while using a surprisingly small amount of disk space and memory. As we mentioned, Tivoli 3.6 provides both stability and extensibility to all customers.

1.4 Overview of Tivoli Management Agent

The most visible new feature of Version 3.6 of the Tivoli Management Framework is the Tivoli Management Agent (TMA), previously called the Lightweight Client Framework (LCF) Endpoint. The TMA is an extension of the classic TME 10 Framework that increases scalability of TMRs while reducing the hardware and software requirements on the managed systems.

The following sections describe this new architecture and its main components including the TMA.

1.4.1 TMA introduction

The TMA-related extensions to the Framework introduce three object types that represent system roles in a TMR:

- Endpoint (TMA)
- Endpoint Gateway
- Endpoint Manager

Although each of the above items logically represents a different system's role in the Tivoli environment, it should be noted that a single physical system can contain more than one of these object types. That is, one system could contain an Endpoint Manager, an Endpoint Gateway, and an Endpoint (however, in most environments, Endpoint Gateways will reside on different systems than Endpoint Managers). Figure 4 on page 14 illustrates these three object types.

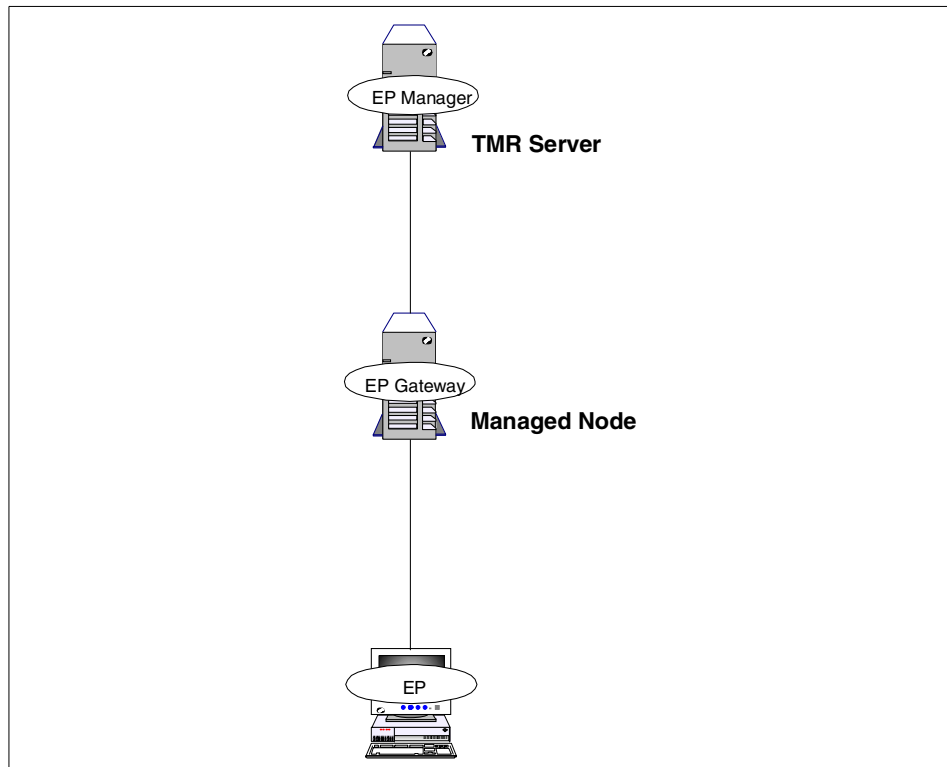


Figure 4. Endpoint Manager, Endpoint Gateway, and Endpoint

1.4.1.1 Endpoint

The Endpoint is installed on systems to be managed. The Endpoint does not include any capability to perform management operations on other systems. That is, as with most end-user workstations, these systems will be managed, but they will not be involved in the management of other nodes. More specifically, the Endpoint does not provide true Tivoli Desktop or command line interface (CLI); so, these resources in the network cannot be managed from the Endpoint.

The Endpoint function resides in the node to be managed. It runs as a small daemon or background task. This daemon is called the *lcmd*. It is responsible for executing methods at the request of a managing system. Its only connection to and knowledge of the rest of the Tivoli world is through an Endpoint Gateway.

When an Endpoint is installed, a minimal number of files are installed on the managed system. Functionally, the only thing that is installed is the *lcmd* itself.

When an application invokes a method to be executed on the managed system (Endpoint), the method is automatically downloaded to the Endpoint and executed by the lcf. The methods that are downloaded to the Endpoint are cached at the Endpoint. As long as that method stays in the cache, it does not need to be downloaded again at a second invocation of the same method. The cache on the Endpoint is a disk cache; therefore, it is persistent across IPLs of the managed system.

1.4.1.2 Endpoint Gateway

The Endpoint Gateway is a software component that runs on a full Managed Node enabling the Managed Node to operate as a Gateway between a cluster of Endpoints and the rest of the TMR. Each TMR can have multiple Endpoint Gateways. The number of Endpoint Gateways will depend on factors, such as available system resources, the number of Endpoints, and network topology. Currently, one TMR server can handle up to approximately 200 Endpoint Gateways. This limit is actually based on the number of Managed Nodes that one TMR server can manage. There is no precise limit to how many Endpoints one Endpoint Gateway can handle. This will depend on system resources, performance requirements, and the type of management being performed. However, testing has been done that indicates that, in many environments, up to 2,000 Endpoints or more may be supported by a single Endpoint Gateway.

The Endpoint Gateway performs the following functions:

Listening for Endpoint requests

The Endpoint Gateway maintains (with help from the Endpoint Manager) a list of the Endpoints for which it is responsible. As the Endpoints come on-line, they will attempt to log in to a specific Endpoint Gateway or broadcast a message searching for an Endpoint Gateway. The Endpoint Gateway will receive these transmissions, and, if it is responsible for the given Endpoint, will proceed with the login process. If the entry of the Endpoint does not exist in the Endpoint Gateway's list, the Endpoint Gateway will forward the login request to the Endpoint Manager so that an Endpoint Gateway can be assigned to the Endpoint. This Endpoint login procedure is called initial login. We will explain the AS/400 Endpoint login procedure in Chapter 4, "AS/400 Endpoint configuration" on page 61.

Listening for downcall method requests

Method invocations from other nodes that are targeted as one of the Endpoints for which an Endpoint Gateway is responsible will pass through the Endpoint Gateway. For downcalls, the Endpoint Gateway is transparent. When it receives a method invocation targeted for the Endpoint for which it is

the Endpoint Gateway, it will pass the method invocation (along with the method and any dependencies, if necessary) on to the Endpoint. It will then wait for any method results and pass them back to the original caller.

Listening for Endpoint upcall requests

If the Endpoint needs to invoke an operation on another system, it must invoke a method on its own Endpoint Gateway. The appropriate application that is stored in the Endpoint Gateway will supply the method. This method will then take advantage of the full function of the Managed Node on which it resides to resolve the location of the target object and invoke the appropriate method(s) upon it.

MDist Repeater activities

The Endpoint Gateways are automatically defined as MDist (Multiplex Distribution) repeaters for all of the Endpoints they serve. In the traditional Tivoli Framework, we defined MDist repeaters using the `wrpt` command. The MDist repeater function provides the fan-out facility for the distribution of files and data in the Tivoli management environment. Therefore, if the same file is being distributed to a set of Endpoints using the same Endpoint Gateway, the file only needs to be sent once to the Endpoint Gateway, and the Endpoint Gateway will then handle the distribution of the file to individual Endpoints. This gives you the benefit of an intelligent distribution mechanism with little or no administrative overhead.

Note

AS/400 will support Endpoint Gateway function in the near future. However, it will not require a full Managed Node. Refer to Section 2.4, “Future directions” on page 35, for more information.

1.4.1.3 Endpoint Manager

The Endpoint Manager stores the association between the Endpoint Gateways and Endpoints. Specifically, it performs the following functions:

Maintaining Endpoint list

The Endpoint Manager maintains the Endpoint list that keeps track of every Endpoint in the TMR. This list tracks which Endpoint Gateway is responsible for each of the Endpoints. Based on site-specific settings, the Endpoint Manager reassigns Endpoints if the Endpoint Gateway is unavailable and dynamically adds new Endpoints as they appear on the network. The Endpoint list contains the information necessary to uniquely identify and manage the Endpoints. This includes:

Name of the Endpoint	A user-friendly name for use in the Tivoli Name Registry (TNR).
Endpoint's interpreter	The string denoting the platform and operating system of the Endpoint (such as Windows NT or OS/2).
Object dispatcher identifier (odnum)	A unique system identifier for the Endpoint.
Endpoint Gateway	The name of the Endpoint Gateway that is responsible for communications with the Endpoint.

Endpoint policy execution

The Endpoint Manager plays a role in enforcing site-specific system policies. For example, policies may be put in place that specify which Endpoint Gateway will be assigned to new Endpoints joining the network. These policies could base their decisions on a variety of information regarding the Endpoint, which is included in the Endpoint's initial login request for a new Endpoint Gateway.

1.4.2 Tivoli Management Agent and Tivoli Desktop

In general, during day-to-day activity, the Tivoli administrator will see little difference when managing systems with the TMA compared to Managed Nodes in previous versions. That is, Tivoli applications will fully support the Endpoint, and you will use the Endpoint as subscribers to profile-based applications just as you used Managed Nodes in previous versions of the Tivoli Management Framework.

However, there are some additions and changes to the Tivoli Desktop that the administrator will notice. For instance, a new icon is added to the Tivoli Desktop to represent the Endpoint Manager (see Figure 5 on page 18). You can create and delete an Endpoint Gateway from this icon. You may use this Endpoint Manager resource to view a list of all Endpoint Gateways and the Endpoints managed by each Endpoint Gateway.



Figure 5. The Endpoint Manager Icon on the Tivoli Desktop

In addition, in the traditional Tivoli environment, Managed Nodes were displayed as such in the various policy regions. Endpoints do not appear in policy regions by default. The decision to not have them linked into a policy region was based on the desire to keep the performance of the Tivoli Desktop at a reasonable level, even when thousands of systems are being managed. However, if desired, you may link an Endpoint with a policy region to have its icon displayed.

Although the icons representing the systems do not show up by default within a policy region, Endpoints will be displayed in dialog boxes showing potential subscribers to profiles and jobs. Therefore, you will still be able to use the Tivoli Desktop GUI to manage systems running the TMA.

Note

To display an Endpoint icon in the specific policy region on the Tivoli Desktop, you can use the `wln` command as follows:

```
wln @Endpoint:ep_label @PolicyRegion:policy_region_label
```

Before executing the above command, you need to configure the Endpoint resource as a managed resource of the policy region.

You can automate the above-mentioned process by using the Endpoint policy (for example, `after_install_policy`). Refer to Chapter 4, “AS/400 Endpoint configuration” on page 61, for more information.

1.4.3 Tivoli Management Agent and Command Line Interface

Version 3.6 of the Tivoli Management Framework includes commands specifically related to helping manage Tivoli Management Agents. In this section, we summarize these commands. Refer to the *Tivoli Framework Reference Manual*, SC31-8434, *Tivoli for AS/400 Endpoints User's Guide*, GC32-0279, and the *Tivoli Framework Release Notes*, G110-3028, for detailed information about these commands:

<code>w4inslcf.pl</code>	This command installs an Endpoint on an AS/400 system and, optionally, starts an AS/400 Endpoint daemon job on one or more AS/400 systems. For more information about installing an Endpoint on an AS/400 system, see Chapter 3, “AS/400 Endpoint installation and deployment” on page 37.
<code>winstlcf</code>	This command installs an Endpoint on a UNIX or Windows NT workstation.
<code>wsetpm</code>	This command enables and disables the profile manager for operation in dataless mode. Since Endpoints do not include a Tivoli object database, profile information is not stored on managed systems the way it is for full Managed Nodes. The profile managers must be enabled for dataless operation to allow Endpoints as subscribers.
<code>lcfcd</code>	This command starts the Endpoint daemon (<code>lcfcd</code>) on the Endpoint and installs and removes the daemon as a service on Windows NT.
<code>lcfcd.sh</code>	This command starts the Endpoint daemon (<code>lcfcd</code>) on the UNIX Endpoints.
<code>wcrtgate</code>	This command creates an Endpoint Gateway.

wdelgate	This command deletes an Endpoint Gateway.
wgateway	This command starts, stops, and lists the properties of an Endpoint Gateway. It is also used to synchronize the Endpoint Gateway method cache with that on the TMR server.
wep	This command performs actions on the Endpoint information contained in the Endpoint list maintained by the Endpoint Manager. This command can list or alter the information related to the Endpoints.
wadminep	This command performs a variety of administrative actions on the Endpoints. In general, once the Endpoints are installed, there is little that needs to be done to administer the lcf daemon. However, this command would be useful when first installing and testing the Endpoint.
wgeteppol	This command lists the body and the constant values of Endpoint policy methods. Use this command to extract a current Endpoint policy method that you can modify and then replace with the wputeppol command.
wputeppol	This command replaces the body of an Endpoint policy method.

We will give examples of the use of most of these commands throughout the rest of this redbook.

1.5 Management topology with TMA

In Version 3.6 of Tivoli, the management topology changed from a two-tiered structure to a three-tiered structure. The three-tiered structure mainly provides the following advantages:

- Off-loading the TMR server
- Increasing the number of systems a single TMR can manage
- Flexible configuration for Endpoints
- High availability for Endpoint operations
- Configuring the MDist repeater automatically

The three-tiered management structure is a natural concept for managing a large-scale environment. According to this concept, the Endpoint Gateway plays the role of mid-level manager. In other words, the Endpoint Gateway takes responsibility for managing the Endpoints that have logged into the

Endpoint Gateway. Therefore, all requests from the Endpoints must be received by the Endpoint Gateway, and most of them will be processed by the Endpoint Gateway instead of the TMR server. Only requests that the Endpoint Gateway cannot handle would be forwarded to the TMR server where the TMR server processes them. In the three-tiered structure, the manager system for the Endpoints is the Endpoint Gateway; so, the Endpoint must send all requests to the Endpoint Gateway.

In the prior TMR structure, the TMR server had to manage all managed systems (Managed Nodes). This was a disadvantage when managing a large-scale environment. In a three-tiered structure, a single Endpoint Gateway can handle several hundreds of Endpoints, and the TMR server (Endpoint Manager) can handle up to 200 Endpoint Gateways. What does this mean? This means we can now say there are few limitations to the number of managed systems in a single TMR. Interconnecting multiple TMRs can still be a very strategic solution for managing a large-scale environment. However, with TMA, we have another powerful solution that has become available. Figure 6 depicts the three-tiered management structure.

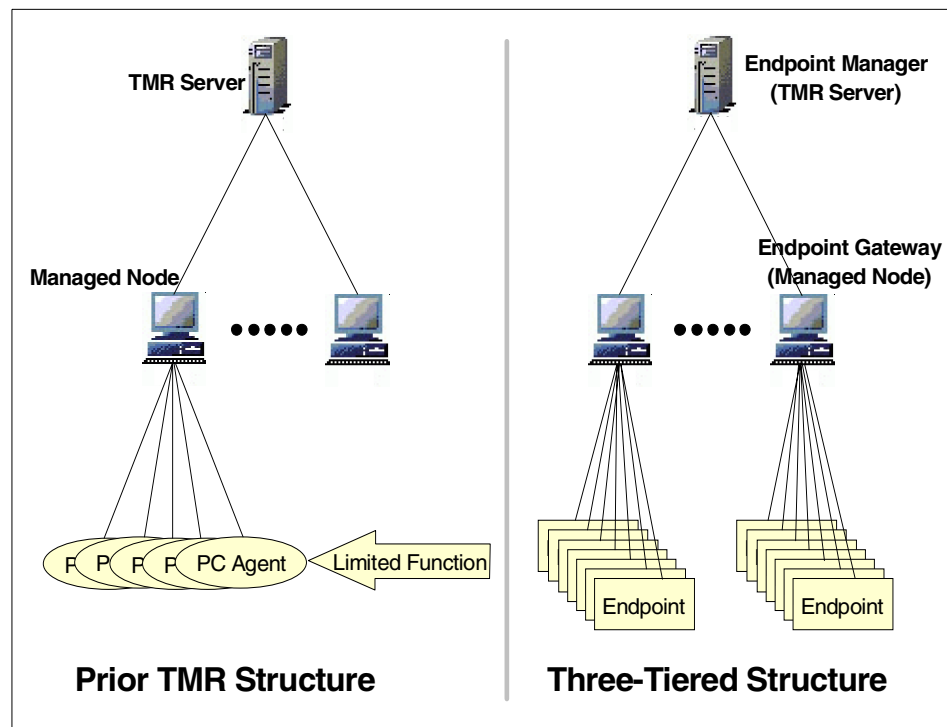


Figure 6. Three-tiered management structure

If you have ever installed a Managed Node, you understand the relationship between the TMR server and the Managed Node. Basically, the TMR server that is used to install the Managed Node manages the Managed Node forever unless you reinstall or reconfigure it. In this situation, if the TMR server becomes unavailable, all management operations will be unavailable.

Although we still have a dependency on the TMR server in a three-tier architecture, much of the TMR server's functions can be off-loaded to an Endpoint Gateway. The Endpoint has the flexibility to automatically log in through an alternate Gateway if its primary Gateway becomes unavailable. In this case, the available Endpoint Gateway is called the alternate Gateway. The Endpoint attempts to log in to the alternate Gateway automatically if the Endpoint detects that the assigned Gateway is unavailable. This is called an isolate login. But, if you do not install or configure any alternate Gateways, the Endpoint will not be able to receive management operations when the assigned Gateway goes down.

Version 3.6 of Tivoli provides improved functions; however, we have to understand them thoroughly if we want to use them efficiently. From this point of view, in the three-tiered environment, the following concepts are very important:

- TMR design
- Management resource allocation (EP Manager, EP Gateway, Endpoint)
- Endpoint configurations (login interfaces information)

You can read about these in the following publications:

- *All About Tivoli Management Agents*, SG24-5134
- *Tivoli Enterprise Performance Tuning Guide*, SG24-5392

Later in this redbook, we will talk about each Tivoli Enterprise product's specific subjects and provide hints and tips for your implementation and AS/400 management scenarios.

Chapter 2. Tivoli for AS/400 Endpoints product information

This chapter contains information and advice regarding the components of Tivoli that you will need to install and configure the Tivoli for AS/400 Endpoints and the Tivoli Enterprise Console Adapter for AS/400. We will not discuss the components required to install and configure the TMR server, TEC server, or Endpoint Gateways.

2.1 Tivoli for AS/400 Endpoints package

In the course of this book, we will discuss the installation, setup, and use of the components of the Tivoli Management Framework for AS/400. The following products are supported by the AS/400 Endpoint:

- Tivoli Inventory
- Tivoli Software Distribution
- Tivoli User Administration
- Tivoli Security Management
- Tivoli Distributed Monitoring

As well as installing the base Framework software, we will need to install all the Tivoli base components of the products listed above that we will be using in the AS/400 management environment. For installation steps, refer to the appropriate product manual or installation guide.

In this book, we will also discuss the Tivoli Enterprise Console (TEC) Adapters for the AS/400. TEC Adapters do not use the Endpoint for communication but, instead, communicate directly with a TEC server. The TEC server can be installed on the same machine as the TMR server, which is the case in the test environment we used for this book.

2.2 Tivoli for AS/400 Endpoint products

This chapter contains details of the products and patches required for the installation of the Tivoli for AS/400 Endpoints and the TEC Adapters for AS/400.

There are two stages to installing the Tivoli for AS/400 Endpoints. First, you need to install Version 3.6 of Tivoli for AS/400 Endpoints. When this is installed, you have to apply an upgrade package with patches to bring the

version up to 3.6.1. You will also require various other patches used to fix problems and improve overall performance and stability of the Endpoints.

The TEC Adapters are much easier to install. Currently, Version 3.6.x is available for the AS/400. The software is installed by sending a save-file directly to the AS/400 and restoring the product (5763TA1) on the AS/400 machine. Refer to Section 7.3, “TEC AS/400 Event Adapters” on page 270, for more detailed information about AS/400 Event Adapter installation.

For the purposes of testing, we used the equipment listed in Table 3 to install and test the AS/400 Endpoints and TEC Adapters.

Table 3. Equipment used for the project

Machine Name	Function	Machine type	Operating system
itso1	TMR server and TEC server	RS/6000	AIX V4.1
itso2	Endpoint Gateway	RS/6000	AIX V4.1
ishii3	Endpoint Gateway	IBM PC	Windows NT 4.0
ishii1	Endpoint	AS/400	OS/400 V4R4
ishii2	Endpoint	AS/400	OS/400 V4R4

Figure 7 on page 25 shows the test environment we used for this project.

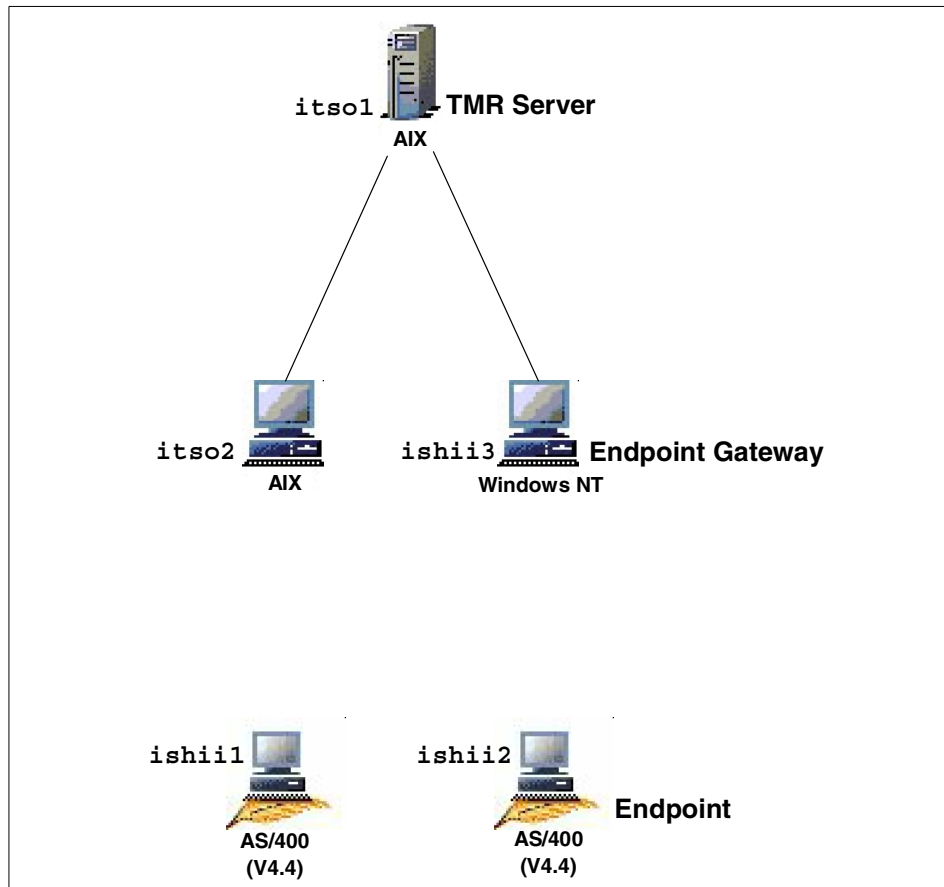


Figure 7. Test environment for the project

2.2.1 Tivoli for AS/400 Endpoints interpreter types

When installing the AS/400 Endpoint software, you must select the correct interpreter (interp) type for the AS/400 on which you want to install. The two types currently available for the AS/400 are shown in Table 4.

Table 4. Tivoli interpreter types supported on the AS/400

Interp type	AS/400 platform supported
os400-v3r2	Supported on AS/400 CISC models with Version 3 Release 2 of OS/400
os400-v3r7	Supported on AS/400 RISC models with Version 3 Release 7 of OS/400 or higher

Both versions of the interpreter type are available with Version 3.6 and Version 3.6.1 of Tivoli for AS/400 Endpoints.

2.2.2 Tivoli for AS/400 CD-ROM media

The Tivoli for AS/400 CD-ROMs contain all the required software to install Tivoli for the AS/400 Endpoint and TEC Adapters. The AS/400 installation packages for the following Tivoli products are contained on the CD-ROM media.

- Tivoli Framework for AS/400 Endpoints
- Tivoli Distributed Monitoring for AS/400 Endpoints
- Tivoli Software Distribution for AS/400 Endpoints
- Tivoli Inventory for AS/400 Endpoints
- Tivoli User Administration for AS/400 Endpoints
- Tivoli Security Management for AS/400 Endpoints
- Tivoli Enterprise Console Adapters for AS/400

The AS/400 Endpoint products are installed as a series of patches listed in Table 5 on page 30 and Table 6 on page 31. TEC Adapter is installed directly on the AS/400 and configured from there.

Note

Tivoli for AS/400 CD-ROM does not contain the TEC Event Adapters for AS/400 code. Refer to Section 2.2.4, “TEC adapters for AS/400 CD-ROM” on page 27, for more information.

Tivoli for AS/400 CD-ROM is currently shipped on a separate CD-ROM from Tivoli Enterprise application’s CD-ROMs, such as Tivoli Distributed Monitoring.

Before installing the software for these products, we need to know where they are located in the set of Tivoli installation media.

2.2.3 Tivoli for AS/400 CD-ROM media

The products and patches for the AS/400 Endpoint are contained on the following two CD-ROMs:

- Version 3.6 of Tivoli for AS/400 CD-ROM
- Version 3.6.1 of Tivoli for AS/400 CD-ROM

Both CD-ROMs have the same directory structure as follows:

```
/1tme1cf  
/as4patch  
/books  
/obj_pack
```

The directories contain the following:

- **/1tme1cf** - AS/400 Licensed Internal Program (1tme1cf)
- **/as4patch** - Tivoli for AS/400 Endpoints upgrade patches
- **/books** - Tivoli for AS/400 Endpoints User's Guide
- **/obj_pack** - Object Packager Utility for Software Distribution

2.2.4 TEC adapters for AS/400 CD-ROM

The TEC adapters for AS/400 installation software can be found on the Tivoli Enterprise Console Version 3.6 CD-ROM.

Version 3.6 of the Tivoli Enterprise Console product CD-ROM is not exclusively for AS/400 Event Adapters and, therefore, it contains a lot of other items that we will not need for our installation.

This CD has the following directory structure:

```
/books  
/new  
/non-tme  
/plus  
/Q5733TA1  
/TANDEM  
/tools  
/upgrade
```

The directories applicable to the AS/400 TEC adapters are:

- **/books** - TEC adapter manuals
- **/non-tme/as400** - AS/400 save files for TEC licensed programs
- **/upgrade** - Patches for TEC Adapters

Note

TEC Event Adapters for AS/400 can be installed directly from the CD-ROM onto the AS/400. Please refer to Section 7.3, “TEC AS/400 Event Adapters” on page 270 for more detailed information about AS/400 Event Adapter installation.

2.2.5 Version 3.6 of Tivoli for AS/400 CD-ROM

Version 3.6 of the Tivoli for AS/400 CD-ROM contains all the patches you need to enable AS/400 Endpoint support on Version 3.6 of Tivoli. You must install all these patches before attempting to install Version 3.6.1 of Tivoli for AS/400. Select the **Install Patches** option on the Tivoli Desktop to install the patches on the TMR server and Endpoint Gateway nodes as shown in Figure 8 on page 29.

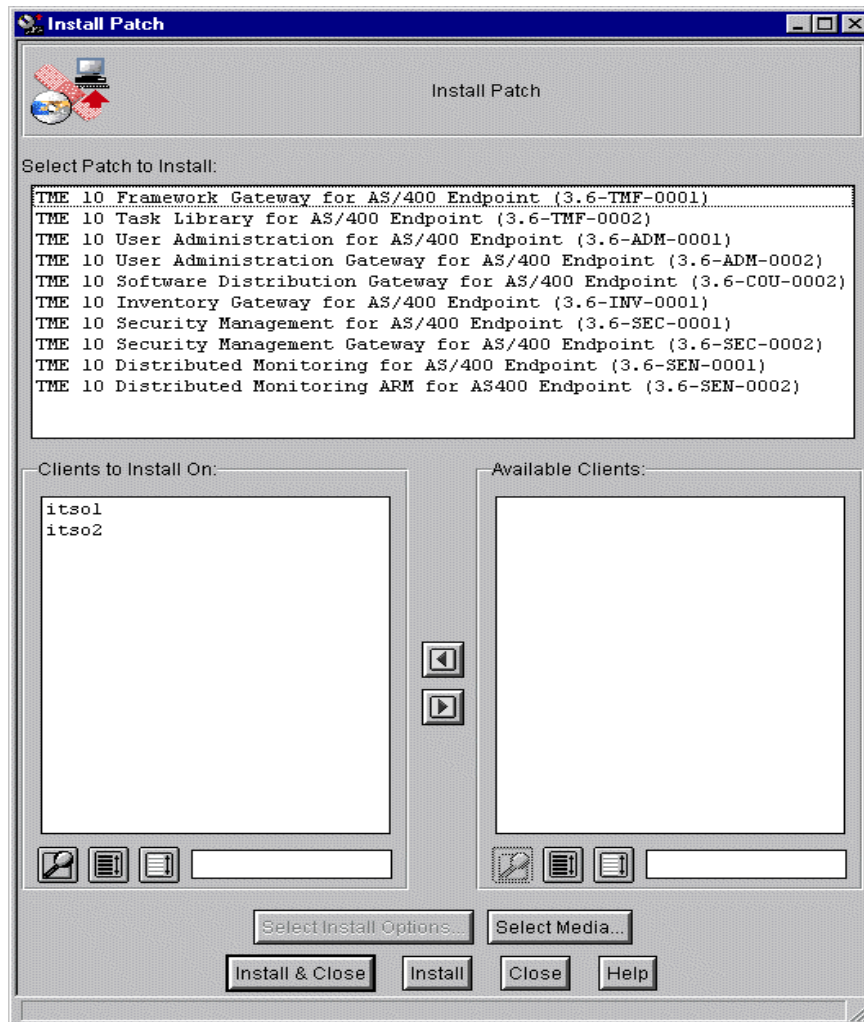


Figure 8. Version 3.6 of Tivoli for AS/400 CD-ROM

2.2.6 Version 3.6.1 of Tivoli for AS/400 CD-ROM

Version 3.6.1 of the Tivoli for AS/400 CD-ROM contains all the patches needed to enable AS/400 Endpoint support on Version 3.6.1 of Tivoli. Before installing these patches, make sure that you have installed all the patches for Version 3.6 of Tivoli for AS/400. Select the **Install Patches** option on the Tivoli Desktop to install the patches on the TMR server and Endpoint Gateway nodes as shown in Figure 9 on page 30.



Figure 9. Version 3.6.1 of Tivoli for AS/400 CD-ROM

2.2.7 Endpoint patches installed on the TMR server

The patches required on the TMR server for Version 3.6.1 of Tivoli for the AS/400 are detailed in Table 5, which contains details of the patch names and references.

Table 5. Products and patches required for the TMR server installation

Product name	Patch reference
TME Framework Gateway for AS/400 Endpoint	3.6-TMF-0001

Product name	Patch reference
Tivoli Task Library for AS/400 Endpoint	3.6-TMF-0002
Tivoli Management Agent (Gateway) 3.6.1 for AS/400 Endpoint	3.6.1-TMF-0001
Tivoli User Administration for AS/400 Endpoint	3.6-ADM-0001
Tivoli User Administration for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-ADM-0001
Tivoli Security Management for AS/400 Endpoint	3.6-SEC-0001
Tivoli Security Management for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-SEC-0001
Tivoli Inventory for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-INV-0001
Tivoli Distributed Monitoring for AS/400 Endpoint	3.6-SEN-0001
Tivoli Distributed Monitoring for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-DMN-0001

2.2.8 AS/400 Endpoint patches for the Endpoint Gateway

The products and patches listed in Table 6 are required on the Endpoint Gateway for Version 3.6.1 of Tivoli for the AS/400. Table 6 contains details of the product and patch names as well as the patch reference.

Table 6. Products and patches required on EP Gateway for AS/400 Endpoints

Product name	Patch reference
Tivoli Framework Gateway for AS/400 Endpoint	3.6-TMF-0001
Tivoli Task Library for AS/400 Endpoint	3.6-TMF-0002
Tivoli User Administration for AS/400 Endpoint	3.6-ADM-0001
Tivoli User Administration for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-ADM-0001
Tivoli 10 User Administration Gateway for AS/400 Endpoint	3.6-ADM-0002
Tivoli Software Distribution Gateway for AS/400 Endpoints 3.6.1 Upgrade	3.6.1-COU-0001
Tivoli Software Distribution Gateway for AS/400 Endpoint	3.6-COU-0002
Tivoli Inventory Gateway for AS/400 Endpoint	3.6-INV-0001
Tivoli Inventory for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-INV-0001
Tivoli Inventory Gateway for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-INV-0002
Tivoli Security Management Gateway for AS/400 Endpoint	3.6-SEC-0002

Product name	Patch reference
Tivoli Distributed Monitoring for AS/400 Endpoint	3.6-SEN-0001
Tivoli Distributed Monitoring for AS/400 Endpoint 3.6.1 Upgrade	3.6.1-DMN-0001

2.2.9 Tivoli support patches not available on CD-ROMs

The next two sections contain information about a number of extra patches that we installed in our test environment. Most of these patches are suggested fixes for problems encountered with the various Tivoli products. These patches are not contained on the Tivoli for AS/400 CD-ROM media and must be downloaded from the following Tivoli Web site:

<http://www.tivoli.com>

Note

You should regularly check the Tivoli Systems Internet support pages for new patches available for the AS/400 Endpoint software. The URL for this site is:

<http://www.tivoli.com/support/>

Patches are listed by product in the Downloads section of the support page. You will be able to obtain the readme files as well as patches, and we strongly recommend that you check the readme files before applying the patches.

2.2.9.1 Support patches available for AS/400 Endpoints

These patches address a number of problems and performance issues related to the AS/400 Endpoints. Table 7 lists all patches that were applied to both the TMR server and the Endpoint Gateways in our AS/400 environment.

Table 7. Support patches available for AS/400 Endpoints

Tivoli patch	Patch description
Tivoli Distributed Monitoring Patch 3.6.1-DMN-0008	Prerequisite for patch 3.6.1-DMN-0013
Tivoli Distributed Monitoring Patch 3.6.1-DMN-0013	Prerequisite for patch 3.6.1-DMN-0018
Tivoli Distributed Monitoring Patch 3.6.1-DMN-0018	Fixes dmengine job performance problem

Tivoli patch	Patch description
Tivoli Framework Patch 3.6.1-TMF-0004	Fixes various Framework problems
Tivoli Management Agent Gateway 3.6.1 w/ DBCS for AS/400 (3.6.1-TMF-0012)	Fixes various AS/400 Endpoint problems
Tivoli Framework Patch 3.6.1-TMF-0035	Fixes various Endpoint management problems
Tivoli Framework Patch 3.6.1-TMF-0041	AS/400 Endpoint Fixes
Tivoli Software Distribution Patch 3.6.1-COU-0011	Fixes problems with creating file packages using Object Packager utility

The patches were first downloaded to our TMR server via the Internet and then installed using the Install Patch option on the Desktop menu of the Tivoli Desktop.

Note

After installing patch 3.6.1-TMF-0012 and 3.6.1-TMF-41 you must upgrade your AS/400 Endpoint daemons using the following command from the TMR server or Managed Nodes in the TMR:

```
wadminep ep_label upgrade
```

where *ep_label* is an AS/400 Endpoint label.

When you have installed all the Tivoli AS/400 patches for Version 3.6 and Version 3.6.1, the Endpoint software version level should be 21 or greater. To see the endpoint version, run the following command from the TMR Server or Endpoint Gateway:

```
wadminep ep_label view_version
```

where *ep_label* is an AS/400 Endpoint label.

2.2.10 OS/400 PTF levels

Although there are no Tivoli-specific OS/400 PTFs for the products we have installed in our test environment, it is recommended that you install the latest AS/400 Cumulative PTF package available for your release of OS/400. Since we are using OS/400 V4R4, we ordered and applied the latest Cumulative PTF package for this release, C9230440.

Note

The Cumulative PTF Package for OS/400 V4R4, C9230440, was the latest version available for this release as of October 27, 1999. The Cumulative PTF packages are updated on a regular basis and vary for each release of OS/400.

2.3 Related documents and information

The following are useful sources of information about Tivoli products:

- *Tivoli Framework AS/400 3.6.2 User's Guide*, GC32-0279
- *Tivoli for AS/400 Release Notes Version 3.6.1*, GI10-8016
- *TME10 Inventory 3.6 User's Guide*, GC31-8381
- *TME 10 Software Distribution 3.6 User's Guide*, GC31-8330
- *User Admin 3.6.2 User's Guide*, GC32-0291
- *Security Management User's Guide 3.6*, SC31-8475
- *Distributed Monitoring User's Guide 3.6*, GC31-8382
- *TME 10 Enterprise Console User's Guide Version 3.6*, GC31-8506
- <http://www.tivoli.com/support/>
- <http://www.tivoli.com/products/documents/whitepapers/>
- <http://www.redbooks.ibm.com/solutions/tivoli>

The following are useful sources of information for the AS/400:

- *Basic System Operation, Administration, and Problem Handling*, SC41-5206
- *TCP/IP Configuration and Reference V4R4*, SC41-5420
- *AS/400 System Operation V3.6.0*, SC41-4203
- *OS/400 Security Reference V4R4*, SC41-5302
- *AS/400 Alerts Support V4R1*, SC41-5413
- *Communications Configuration V4R1*, SC41-5401
- <http://www.as400.ibm.com>
- <http://www.redbooks.ibm.com>

2.4 Future directions

Tivoli is currently beta testing the Tivoli Endpoint Gateway for AS/400. This product will enable you to manage Endpoints from an AS/400. With this code, the AS/400 will be able to act as an Endpoint Gateway. The Endpoint Gateway for AS/400 does not require Managed Node; so, only gateway processes will run on the AS/400 system. The gateway process running will be independent of the Managed Node on all platforms in the near future. The AS/400 Gateway will be generally available (GA) this year.

It is also announced that Tivoli Manager for Domino and Tivoli Manager for MQSeries will be GA this year. It enables you to manage your important applications running AS/400 platforms as efficiently as other platforms, such as UNIX or Windows NT.

Chapter 3. AS/400 Endpoint installation and deployment

In this chapter, we discuss the planning and installation of the AS/400 Endpoints. This chapter includes the following information:

- Hardware, software, and configuration requirements for AS/400 Endpoints
- AS/400 Endpoint-related considerations
- Overview of AS/400 Endpoint installation and configuration
- Planning for mass installation

We recommend that you refer to the appropriate manuals and documents that we introduce in this chapter as complete sources of information.

3.1 Planning for AS/400 Endpoint

The following sections provide information regarding planning for the AS/400 Endpoint. This information is based on Version 3.6.1 of Tivoli for AS/400 Endpoints released in April, 1999. Refer to the book, *Tivoli for AS/400 Endpoints Release Notes*, GI10-8016, for this or later versions of the latest information.

3.1.1 Installation prerequisites

The following are the management objects required to install and use AS/400 Endpoints:

TMR Server	The Tivoli Management Server component includes the libraries, binaries, data files, and graphical user interfaces needed to install and manage the Tivoli environment.
Endpoint Manager	The Endpoint Manager runs on the TMR server. The Endpoint Manager maintains the information related to known Endpoints and Endpoint Gateways. The Endpoint Manager is automatically installed on TMR servers. The Endpoint Manager's primary role is to assign the Endpoint to the Endpoint Gateway when the Endpoint performs the initial login. The Endpoint Manager must always be involved if an alternate Gateway will be used by an Endpoint. This might occur through explicit administrative action to migrate an Endpoint from one Gateway to another or if an

Endpoint attempts another initial login due to its primary Gateway being unavailable for any reason.

Endpoint Gateway The Endpoint Gateway provides the primary interface between a set of Endpoints and the rest of the TMR. As part of this role, it also assumes some of the functions previously performed by the TMR server. By shifting a share of the management processes to the Endpoint Gateway, the TMR servers are free to service more managed systems than with previous versions of Tivoli. A single Endpoint Gateway can support communications with thousands of Endpoints. In Version 3.6 of the Tivoli Management Framework, Endpoint Gateways must be installed on Managed Nodes.

Endpoint The Endpoints are managed systems taking advantage of the Lightweight Client Framework. You can gather required management information from thousands of Endpoint machines and remotely manage those machines with very little overhead. Another advantage of Endpoints is the relatively small demand they make on computer resources.

3.1.2 System requirements

This section will lay out the system requirements (hardware, networking, operating system, and software) for successfully installing and operating a Tivoli Endpoint for AS/400.

3.1.2.1 Hardware requirements

Currently, the Tivoli AS/400 Endpoint runs on both CISC- and RISC-based processors. To be able to manage the AS/400 Endpoint, there must be physical networking to the AS/400 (for example, Ethernet, Token-Ring, and so on).

Disk space usage by the Tivoli Endpoint program is very limited on an AS/400. Determining the exact amount that will be used is difficult. This is because there are many different environments. The default Tivoli Endpoint installation on an AS/400 Endpoint uses a certain amount of space, but, once the Endpoint starts to be used with Tivoli applications (for example, Distributed Monitoring, User Administration, or Inventory), more space needs to be used for binaries and libraries that are retrieved from the Endpoint Gateway.

There are currently two supported Tivoli interpreter types (this is product-specific code for each version of OS/400) available for installation. Table 8 shows the amount of space needed for the Tivoli AS/400 Endpoint binaries.

Table 8. Needed disk space for AS/400 Endpoint

AS/400 Model	Interpreter Type	Needed Disk Space
CISC	os400-v3r2	3.20 MB
RISC	os400-v3r7	4.40 MB

3.1.2.2 Networking requirements

To be able to install and run an AS/400 Endpoint, the TCP/IP networking protocol suite is needed. Tivoli uses two layers of this communication protocol. *UDP and TCP are both needed to run Tivoli AS/400 Endpoint.* Most communications with the Endpoint are through TCP, but initial login and Endpoint Gateway migrations use UDP for communications.

3.1.2.3 Operating system requirements

OS/400 revisions are extremely important when determining whether a particular AS/400 model (RISC or CISC) is supported as being an Endpoint. Tivoli does not currently have prerequisites for any PTFs on AS/400 machines before installing the Tivoli Endpoint code to the machine. The important fact is that *the only version of OS/400 that is supported on CISC models is V3R2*. When running RISC-based processors, V3R7 or higher is needed. You also need an AS/400 user profile with appropriate authority to perform Tivoli management operations in your environment.

3.1.2.4 Software requirements

In Tivoli for AS/400 Endpoints Version 3.6.1, there are approximately 11 patches distributed with the CD-ROM. All of these patches reside in the /as4patch directory on the CD-ROM. Most of these patches need to be applied to Endpoint Gateways that will be hosting AS/400 Endpoints. Each product has individual patches that need to be applied. Scenarios that will provide a layout for each product environment and the steps needed to accomplish them are described below. Each of these environments assume that you have a TMR, Endpoint Gateways appropriately attached to that TMR, and products already installed on the affected machines.

Tivoli AS/400 Endpoint (no other Tivoli products)

Perform the following steps to install Tivoli for AS/400:

1. Apply the 3.6.1-TMF-0001 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint.

2. Install the AS/400 Endpoint using Software Installation Service (described in Section 3.4.1, “Using the Software Installation Service (SIS)” on page 47) or the `w4inslcf.pl` command (described in Section 3.4.2, “Using the `w4inslcf.pl` command” on page 52).

Tivoli User Administration

Perform the following steps:

1. Install Endpoint Gateway patches, and install the AS/400 Endpoint. Refer to “Tivoli AS/400 Endpoint (no other Tivoli products)” on page 39 in this section for detailed steps.
2. Apply the 3.6.1-ADM-0001 patch to the TMR server only. If you install this patch to the Endpoint Gateway or Managed Node, the installation will be corrupted. This patch contains all the appropriate user interface graphical extensions, therefore, it is only needed on the TMR server.
3. Apply the 3.6.1-ADM-0002 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint. This will place the User Administration methods and binaries on the Endpoint Gateway to enable it to manage AS/400 Endpoints with User Administration.

Tivoli Security Management

Perform the following steps:

1. Install Endpoint Gateway patches and install the AS/400 Endpoint. Refer to “Tivoli AS/400 Endpoint (no other Tivoli products)” on page 39 in this section for detailed steps.
2. Apply the 3.6.1-SEC-0001 patch to the TMR server only. If you install this patch to the Endpoint Gateway or Managed Node, the installation will be corrupted. This patch contains all the user interface graphical extensions for the Tivoli Desktop.

Note

To install this patch, you need **super** and **security_admin** Tivoli security authorities.

Tivoli Distributed Monitoring

Perform the following steps:

1. Install Endpoint Gateway patches and install the AS/400 Endpoint. Refer to “Tivoli AS/400 Endpoint (no other Tivoli products)” on page 39 in this section for detailed steps.
2. Apply the 3.6.1-DMN-0001 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint. This patch contains the lightweight

Distributed Monitoring engine (Sentry engine) that will be pushed out to the Endpoints from the Endpoint Gateway when the Distributed Monitoring profile is initially distributed to the Endpoint.

Tivoli Distributed Monitoring ARM

Perform the following steps:

1. Install Endpoint Gateway patches and install the AS/400 Endpoint. Refer to “Tivoli AS/400 Endpoint (no other Tivoli products)” on page 39 in this section for detailed steps.
2. Apply the 3.6.1-DMN-0001 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint. This patch contains the lightweight Distributed Monitoring engine (Sentry engine) that will be pushed out to the Endpoints from the Endpoint Gateway when the Distributed Monitoring profile is distributed to the Endpoint initially.
3. Apply the 3.6.1-DMN-0002 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint. This patch contains the actual Application Response Management (ARM) agent that will be pushed out of the AS/400 Endpoint from the Endpoint Gateway to monitor availability and performance transactions.

Tivoli Software Distribution

Perform the following steps:

1. Install Endpoint Gateway patches and install the AS/400 Endpoint. Refer to “Tivoli AS/400 Endpoint (no other Tivoli products)” on page 39 in this section for detailed steps.
2. Apply the 3.6.1-COU-0001 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint. This patch simply contains the methods that will enable this Endpoint Gateway to be able to communicate with the AS/400 with Software Distribution.

Tivoli Inventory

Perform the following steps:

1. Install Endpoint Gateway patches and install the AS/400 Endpoint. Refer to “Tivoli AS/400 Endpoint (no other Tivoli products)” on page 39 in this section for detailed steps.
2. Apply the 3.6.1-INV-0001 patch to the TMR server only. If you install this patch to the Endpoint Gateway or Managed Node, the installation will be corrupt. This patch contains the server and database schema scripts for customizing your inventory repository.

3. Apply the 3.6.1-INV-0002 patch to the Endpoint Gateway that will be hosting the AS/400 Endpoint. This patch contains the Inventory methods for AS/400 Endpoints.

Tivoli Enterprise Console

Refer to Section 7.3, “TEC AS/400 Event Adapters” on page 270, for more information about this Tivoli product and its compatibility with the AS/400.

Table 9 summarizes the information provided in this section.

Table 9. Required patches to upgrade AS/400 Endpoint to Version 3.6.1

Product name	Patch
Tivoli Management Framework (AS/400 Endpoint)	3.6.1-TMF-0001
Tivoli User Administration	3.6.1-ADM-0001
	3.6.1-ADM-0002
Tivoli Security Management	3.6.1-SEC-0001
Tivoli Distributed Monitoring	3.6.1-DMN-0001
Tivoli Distributed Monitoring ARM	3.6.1-DMN-0001
	3.6.1-DMN-0002
Tivoli Software Distribution	3.6.1-COU-0001
Tivoli Inventory	3.6.1-INV-0001
	3.6.1-INV-0002

3.2 AS/400 Endpoint planning considerations

When preparing for installation or deployment of AS/400 Endpoints, proper planning is important in order to save time and prevent inconvenience. Tivoli supports two methods of installation: Software Installation Service (SIS) and the w4inslcf.pl script. When preparing to use either of these solutions, one should consider the reasons for using one instead of the other in order to save time in deployment. Carefully read each of the methods described in later sections to be able to determine this. We also provide a section that describes the benefits of each of these methods.

3.3 Overview of AS/400 Endpoint installation

To utilize the AS/400 Endpoint, you must have both an Endpoint Manager and an Endpoint Gateway installed and configured properly. In Version 3.6 of the Tivoli Management Framework, the Endpoint Manager must run on the TMR server, and the Endpoint Gateway must run on a Managed Node.

The following sections describe the installation of Endpoints and their prerequisites.

3.3.1 Endpoint Manager installation

As we mentioned, the Endpoint Manager is automatically installed when the TMR server is installed. We will not review the details of installing the TMR server. This is well-documented in the *TME 10 Framework 3.6 Planning & Installation Guide*, SC31-8432. Once installation is complete, the Endpoint Manager will be available. If you are familiar with Version 3.1 of the Tivoli Management Framework, you will notice only one significant difference on the Tivoli Desktop in a Version 3.6 installation: The Endpoint Manager icon (see Figure 10 on page 44).

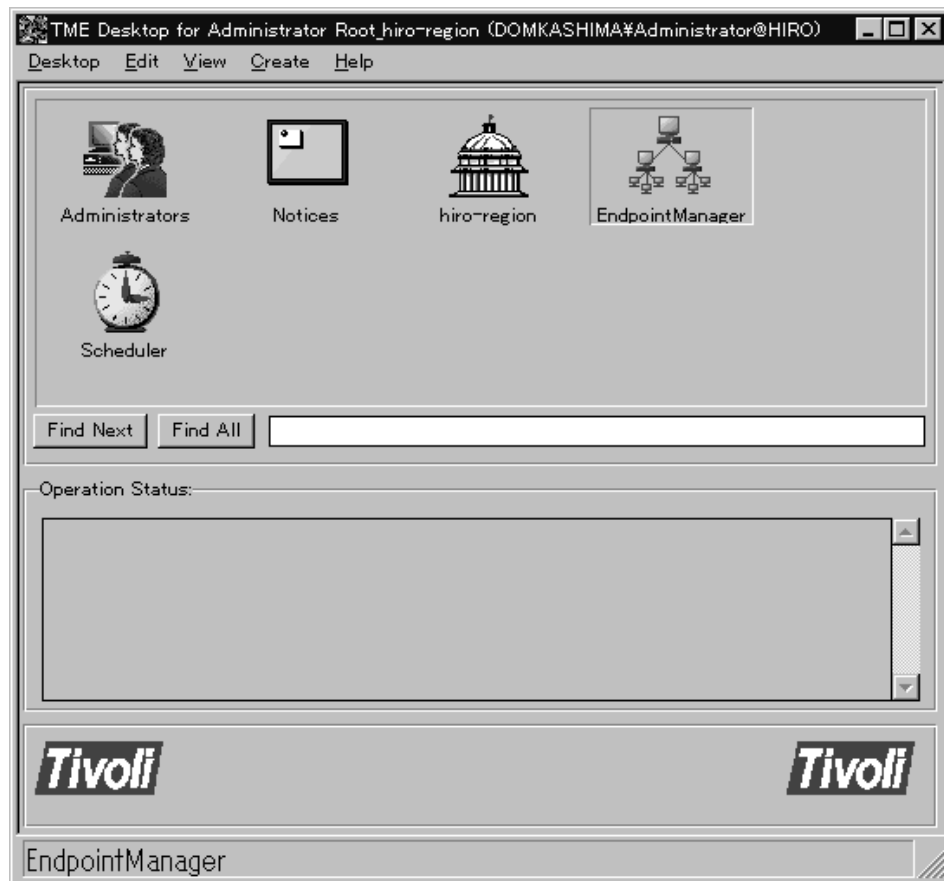


Figure 10. The Tivoli Desktop after TMR server installation completion

From the command line interface, you can verify that the Endpoint Manager exists by issuing the `wls` command. The following screen contains an example of this command.

```
C:\Tivoli>wls
Notices
Administrators
hiro-region
EndpointManager
Scheduler
C:\Tivoli>
```

3.3.2 Managed Node installation

The Endpoint Gateways must be created on a Managed Node or on the TMR server. Except for very small or test environments, it is, typically, not recommended to install an Endpoint Gateway on the TMR server since this goes against one of the goals of the architecture, that is, to off-load the TMR server. This means, of course, that we have to create the Managed Node, which will be configured as the Endpoint Gateway before the Endpoint Gateway installation. Several methods of installing a Managed Node are described in the following sections.

Local installation (Windows NT Managed Node only)

You can set up the Windows NT Managed Node locally from the Tivoli Management Framework CD-ROM. After this operation, you can install the Endpoint Gateway using the Tivoli Desktop interface or the `wortgate` command. Both operations are able to be performed on the target Managed Node (the Tivoli Desktop module should be installed on the target node when you use the Tivoli Desktop interface). The commands to install the Managed Node and the Tivoli Desktop are:

- `setup client` for Windows NT Managed Node
- `pc/desktop/disk1/setup` for Tivoli Desktop

Installation from TMR Server

You can install the Managed Nodes remotely from a TMR server. Since you can install all the Managed Nodes from a single machine, this is one of the most convenient methods of installing Managed Nodes. The `install_client` authorization role for the administrator is required for this operation.

On the Tivoli Desktop, double-click on the policy region icon in which you want to create the Managed Node. On the policy region window, select the **Create** menu to display the Managed Node Installation dialog. You can create the Managed Node from this dialog. Before you install the Framework on Windows NT systems, you must install the Tivoli Remote Execution Service (TRIP) on a single Windows NT system in your TMR. If your TMR server is Windows NT, this step is performed automatically. Refer to the *TME 10 Framework 3.6 Planning & Installation Guide*, SC31-8432, for more information.

The `wclient` command can also be used to install a Managed Node remotely from the TMR server. Refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more information.

3.3.3 Endpoint Gateway installation

To install an Endpoint Gateway onto an existing Managed Node, you can use the following methods. The senior authorization role is required for these operations.

Software Installation Service (SIS)

Tivoli Software Installation Service (SIS) enables you to install multiple Endpoint Gateways onto multiple systems in parallel. SIS performs a prerequisite check and user specified prerequisite check if it is defined. SIS also enables you to install TRIP, a Managed Node, and Tivoli Management Applications.

Tivoli Desktop

From the Tivoli Desktop interface, right-click the **Endpoint Manager** icon and select the **Create Gateway** menu. A popup panel appears, and you can install the Endpoint Gateway to the specified Managed Node that has already been created (see Figure 11).

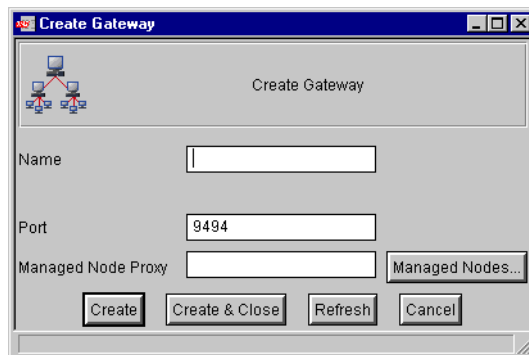


Figure 11. Create Gateway panel

wcrtgate command

The `wcrtgate` command creates the new Endpoint Gateway on the specified Managed Node. If you do not specify the name of the Managed Node, the Endpoint Gateway is created on the Managed Node from which the command was invoked. Refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more information.

3.3.4 AS/400 Endpoint installation

There are also several ways to install the AS/400 Endpoints. We will introduce them in this section.

Software Installation Service (SIS)

Tivoli Software Installation Service (SIS) enables you to install AS/400 Endpoints onto multiple systems in parallel. SIS performs a prerequisite check and user-specified prerequisite check if defined. It has a detailed graphical user interface; so, Tivoli administrators can view installations on an enterprise level.

w4instlcf.pl script

The w4instlcf.pl script is a Perl script written to make installations quick and easy on either a large or small scale. It has no graphical interface to it, but does prompt users for certain options once executed.

3.4 Installing AS/400 Endpoint

In this section, we describe AS/400 Endpoint installation methods, SIS, and the w4instlcf.pl script in detail. You will be able to understand the difference between SIS and the w4instlcf.pl script through this section as well as each method's feature.

3.4.1 Using the Software Installation Service (SIS)

Installing Tivoli for AS/400 Endpoints using the SIS is quite simple and standard for using SIS to install any Endpoint. In this section, we will discuss in detail the process of adding an AS/400 Endpoint through SIS. This will include each step and the commands that are executed to accomplish the installation. For detailed information about using and installing SIS, refer to the *Software Installation Services 3.6 User's Guide*, GC31-5121.

To be able to install an AS/400 Endpoint, one must first start SIS and add the AS/400 Endpoint to the list of available machines to which to install. This is illustrated in Figure 12 on page 48.

Figure 12. Add machine dialog when adding AS/400 into SIS

When installing AS/400 Endpoints, SIS uses the FTP protocol to establish connections, transfer files, and execute remote commands. Because of this, we must select **FTP/Account(OS400 Only)** as the access type. After supplying a correct user ID and password, click the **Submit** button and the Status and Interp Type fields will be automatically updated with the correct information about the newly-added AS/400.

Note

If you do not provide a valid user ID and password when adding the machine, your installation will fail. The inspection process of the AS/400 when adding the machine to your list in SIS does not require a valid user ID and password. It only requires a valid hostname of an AS/400. This can seem quite confusing when the installation process actually begins.

Once the machine is added to the list of available machines within SIS, you should select **TME 10 Endpoints** as the install product as shown in Figure 13. This process is as simple as finding the product in the list of imported products, selecting it, and clicking **OK**. If you are not already done, you may need to import the AS/400 Interp types into SIS. These file packets are available for import from the *Tivoli for AS/400 Endpoints* CD-ROM, LK3T-3639. For more information and instruction on how to import products into SIS, refer to the *Software Installation Services 3.6 User's Guide*, GC31-5121.

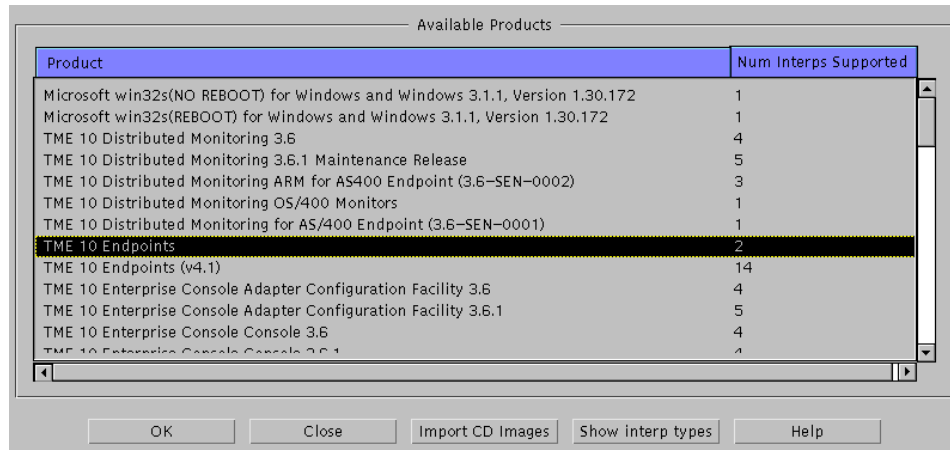


Figure 13. Selecting Endpoint product within SIS

After adding the AS/400 to the machine-list and selecting the appropriate product to install, your Install Details screen will be complete and should look like the example shown in Figure 14.

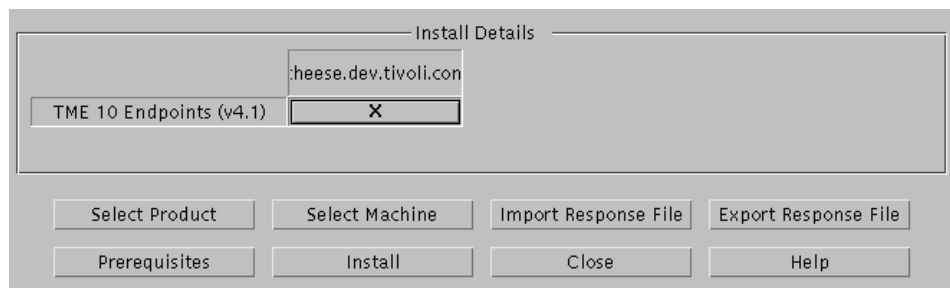
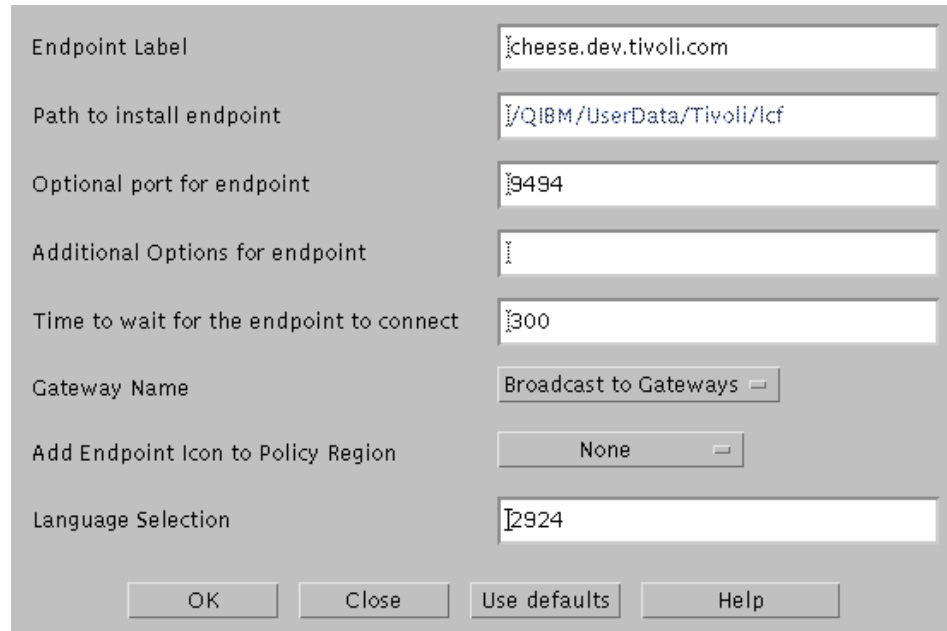


Figure 14. Installing the Details dialog within SIS

To activate the installation to a particular machine, you must place an X in that cell by clicking on it. If the cell is green, the installation has already been executed on this machine, and, if the cell is grey, the installation cannot be executed onto this machine. A detailed explanation on this can be obtained from the *Software Installation Services 3.6 User's Guide*, GC31-5121.

If you right-click on the cell, a Machine Attributes dialog will appear. This dialog will allow you to change specific installation attributes about this machine for the AS/400 Endpoint product.



The dialog box is titled 'Machine Attributes' and contains the following fields and controls:

- Endpoint Label:** Text input field containing 'cheese.dev.tivoli.com'.
- Path to install endpoint:** Text input field containing '/QIBM/UserData/Tivoli/lcf'.
- Optional port for endpoint:** Text input field containing '9494'.
- Additional Options for endpoint:** Text input field containing ' '.
- Time to wait for the endpoint to connect:** Text input field containing '300'.
- Gateway Name:** Dropdown menu with 'Broadcast to Gateways' selected.
- Add Endpoint Icon to Policy Region:** Dropdown menu with 'None' selected.
- Language Selection:** Text input field containing '2924'.
- Buttons:** OK, Close, Use defaults, and Help.

Figure 15. Machine attributes for Tivoli for AS/400 Endpoint within SIS

In this dialog, things, such as the Endpoint Label, the Optional port for endpoint, the Additional Options for endpoint, the Gateway Name, Add Endpoint Icon to Policy Region, and Language Selection can be altered. Each of these options is defined in the following list:

Endpoint label

This is the label that will be registered in the Endpoint Manager database. You can actually change this to be anything that you wish, but, usually, the hostname of the machine is used here for naming convention purposes. The reason that anything can be used here is that the Endpoint Manager does not track an

Endpoint by its label but by its unique identifier, which is assigned upon initial login to an Endpoint Gateway in the TMR.

Path to install endpoint

This is the location on the physical system where the Endpoint code will be installed. This option cannot be changed from the Machine Attributes dialog. This option can only be changed as in the Product Attributes, which can be changed by right clicking the product itself on the Install Details dialog.

Optional port for endpoint

This allows you to select a port other than the default of 9495 on which to install.

Additional options for endpoint

This is where you can place additional options to pass the AS/400 lcf daemon when starting up. Refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more options to pass to the Endpoint.

Time to wait for the endpoint to connect

This allows you to change the default to wait (300 seconds) for the Endpoint to successfully contact its Endpoint Gateway before the installation process responds with a failed installation attempt.

Gateway Name

This is where you can specifically tell your Endpoint what Endpoint Gateway to connect to or just tell it to broadcast to all known Endpoint Gateways.

Add Endpoint Icon to Policy Region

This allows you to add your machine (and add an icon) to a specific policy region in your TMR. This is much easier than linking the Endpoint after installation.

Language Selection

This allows you to specify something other than the default English language. International support is extensive with the AS/400 Endpoint.

To continue with the installation, simply click the **Install** button. The next screen will prompt you to select an installation algorithm. The default is *Install to Machines Independently Maximizing Network Bandwidth*. If you wish to explore other options, refer to your *Software Installation Services 3.6 User's Guide*, GC31-5121, for more information on each of these options.

The installation will now proceed and push the Endpoint binaries onto the AS/400. Communication to the AS/400 from the machine that has SIS

installed on it is done 100 percent through the FTP protocol. SIS has its own internal FTP client that will be used to communicate with the AS/400. By examining the SIS installation logs, you can actually see the FTP commands that are being sent to the AS/400 to test prerequisites, install the software, and actually start the Endpoint.

3.4.2 Using the `w4inslcf.pl` command

Installing AS/400 Endpoints using the `w4inslcf.pl` Perl script is an effective way to install either a few Endpoints or many. To be able to use the `w4inslcf.pl` command, you must first apply the 3.6.1-TMF-0001 patch to your TMR server and to any machines that you want to act as Endpoint Gateways. By applying this patch, the `w4inslcf.pl` script and all supporting scripts will be placed under the `$BINDIR/./lcf_bundle/generic` directory.

The following is the syntax and a short summary of what each of the command-line options means:

Usage: `w4inslcf.pl` <options> host-spec, ...

-h	Extensive help, show more information and examples.
-a	Asynchronous install (do not wait for <code>lcfd</code> to do initial Gateway login).
-F	Force overwrite of existing install.
-s <dir>	Tivoli Management Agent source image location.
-f <file>	Add host names in this file to target list.
-l <lcfd_port>	Port number that Endpoint uses.
-g <gateway_interfaces>	List of Endpoint Gateways that the Endpoint attempts to log in.
-L <switches>	Pass switches to <code>lcfd</code> .
-P	Prompt for the user password at each host.
-Y	Proceed without confirmation.
-I	To only install, it will not start the Endpoint.
-N	To install with a set of languages any of 2924, 2928, 2938, 2962, 2980.
-k	To keep and not delete the installation library and files.

-T Local IP address to use when connected to AS/400.

So, if you want to install an AS/400 Endpoint named *kong* to connect to an Endpoint Gateway named *itso2* on port 9494, which is the default port, you would use the following syntax:

```
$BINDIR/../../lcf_bundle/generic/w4inslcf.pl -g itso2+9494 kong
```

The following screen-shot is what is immediately displayed after executing the command and all input is entered.

```
# $BINDIR/../../lcf_bundle/generic/w4inslcf.pl -g itso2 kong

We need a user account, please type or just hit enter to use [default]
login for kong([QSECOFR]): rsumner
Password for account rsumner:
Ready to install product:
    source: itsol:/usr/local/Tivoli/bin/lcf_bundle/os400_install
    destination: kong+9495
login gateway: ITSO2+9494
    User: rsumner
    file: FILE2.PKT
    NLS: 2924
LCFOPTS:
lcs.login_interfaces=ITSO2
gateway_port=9494

Continue? [yYna?]
```

After typing the command (with its appropriate syntax), you will be prompted for a username and password to access the AS/400 (if one was not supplied on the command line). To supply a username and password on the command line, you simply type `hostname username password` in the host-spec. Once the username and password are supplied, the installation will continue with the installation to the supplied machine.

If you have many machines on which to do an installation, you can supply the `w4inslcf.pl` command with a flat-text file filled with hostnames of AS/400s on which to do the installations. Again, if you do not supply the username and password in this file, each machine will be prompted for a username and password (just like in the example above).

The following is a detailed step-by-step set of instructions about what happens during the installation of the AS/400 Endpoint while using the `w4inslcf.pl` script.

1. Parse the list of Endpoints from either the command line or the file (if the `-f` option was passed).
2. Start a loop to go through the list of Endpoints for installation. Endpoints are installed in serial (first come, first served) while using the `w4inslcf.pl` script.
3. Get the login and password if the username and password were passed on the command line or in the flat-file. If neither was provided, we look in the `.netrc` file to see if it exists. If none of these places have our username and password, we prompt the user for one.
4. Open an FTP socket connection to the hostname of the AS/400. If the FTP session does not respond with `220-QTCP`, this machine must not be an AS/400, and we return a failure. If it does, we do not need to log in, but we merely pass a `SYST` command via FTP, which gives us the version, revision, and modification level of the OS/400 on that machine.
5. Confirm the host installation by prompting the user for a *confirmation of installation* unless the `-y` option was passed from the command-line.
6. Open another socket connection to the AS/400 by binding to the local hostname of the machine from which you are installing (this is obtained by issuing the `hostname` command) and establishing an FTP session with a 1800 second timeout.
7. Check the prerequisites on the machine by performing the following steps:
 - a. Create the library, QTMELCFINS, by running `QSYS/CRTLIB LIB(QTMELCFINS) AUT(*ALL) CRTAUT(*ALL)`.
 - b. Remove the QTMELCFINS/QLCFINS save file. To check whether a new file exists (overwriting a previous failed installation) issue the command `QSYS/DLTF FILE(QTMELCFINS/QLCFINS)`.
 - c. Create the save file, QLCFINS, in the library that we just created by running `QSYS/CRTSAVF FILE(QTMELCFINS/QLCFINS)`.
 - d. FTP the install file into the save file that we just created. The file to be FTPed will be `$BINDIR/../lcf_bundle/os400_install/install_sis/bin/<INTERP>/qlcfins.savf` where `<INTERP>` is the interp-type of your machine.
 - e. Restore the save file that we just populated by issuing `QSYS/RSTOBJ QLCFINST QTMELCFINS DEV(*SAVF) SAVF(QTMELCFINS/QLCFINS)`.
 - f. Before running any commands, we check to see if the `RSTOBJ` command returns any errors. If it does, we must change the user privileges.
 - g. If there was no error, we run the program file that we just restored with the following options: `CALL QTMELCFINS/QLCFINS ('-S1' '-E' '-C P'`

'<anything passed with the -g option>' '<IP address of the interface to bind to>'.

- h. We then parse the output of that command to determine if there was sufficient disk space - if the user is authorized to the `RSTLPCPGM` command, if the `QALWOBJRST` system value is set to `*ALL`, if the Endpoint can communicate with the Endpoint Gateway provided, and if the Endpoint is already installed. If any of these things fails, we return the appropriate error.
8. Install the binaries to the AS/400 by performing the following steps:
 - a. Remove the save file to ensure that you have nothing left over from a previous failed install by running `QSYS/DTLTF FILE(QTMELCFINS/ATMELCF)`.
 - b. Now, create that same save file again by running `QSYS/CRTSAVF FILE(QTMELCFINS/ATMELCF)`.
 - c. FTP the file, `qlcfins.savf`, again into the save file that we just created.
 - d. Run the following command to restore the binaries onto the system:
`CALL QTMELCFINS/QLCFINS('-S2' '-C I' ATMELCF)`.
9. Next, we push out all the language support to the AS/400. We will call this NLS. Each supported language file is under `$BINDIR/./lcf_bundle/os400_install/install_sis/nls/<INTERP>`. Each of these files can be sent by performing the following steps:
 - a. Remove the save file if it exists `QSYS/DTLTF FILE(QTMELCFINS/TMA<language number>)`.
 - b. Create that same savefile that you just removed by issuing:
`QSYS/CRTSAVF FILE(QTMELCFINS/TMA<language number>)`.
 - c. FTP the `TMA<language number>.sf` file into the newly-created save file.
 - d. Then, restore it by calling the following command: `CALL QTMELCFINS/QLCFINS('-S3' '-C I' '<language number>')`.
10. Next, we create the Endpoint configuration file in the temporary directory. This directory is retrieved by running the `wtemp` command. This command is a Tivoli command that returns a directory that can be used for temporary usage. The log file for installation is also placed in this directory. We temporarily create a `lcf.d.cfg` file in that directory (we use the `/tmp` directory) and populate it with all the LCF options that were passed on the command-line.
11. Delete the configuration file on the AS/400. This will make sure we have a clean start by running `QSYS/DTLTF FILE(QGPL/QALCFCFG)`.

12. Create a new source file on the AS/400 so that we can populate it with the configuration file by running `QSYS/CRTSRCPF FILE(QGPL/QALCFCFG) CCSID(37)`.
13. FTP the created `lcfd.cfg` file into that newly-created source file.
14. Next, we actually start the Endpoint on the AS/400 by running `CALL QTMELCFINS/QLCFINS('-S4' '-C R')`.
This step would not happen if the `-I` parameter was used.
15. After starting the Endpoint, we try three times (while waiting ten seconds between attempts) to get the status of the Endpoint through the HTTP interface. During this step, we check to see if the Endpoint was ever able to connect to its Endpoint Gateway successfully. If it was, the installation was successful. If we did not, it tells you that it started but could not connect to the Endpoint Gateway. We will also return an error if we could not connect to the HTTP interface, which most likely means that we could not start the Endpoint for some reason.
16. After the installation is done, we remove the temporarily config file from the temp-dir. We also remove the temporary installation library from the AS/400 by running `DLTLIB LIB(QTMELCFINS)`.

This concludes the installation steps that are accomplished when using the `w4inslcf.pl` script. When debugging problems, you can follow these steps and determine what has and has not yet been done on the AS/400 by examining the QTMELCFINS library.

Note

It is possible to verify that the Tivoli Management Agent for the AS/400 product is properly installed using the Display Software Resources and Check Product Options commands. Section 7.3.4, "Troubleshooting TEC AS/400 Event Adapters" on page 313 shows an example of using these commands for the TEC AS/400 Adapters. This example can easily be modified to verify the TMA Endpoint installation.

3.5 Removing AS/400 Endpoint software

To remove the AS/400 Endpoint code from an AS/400 Endpoint, there are several steps that you must perform. With a normal licensed program, you can simply delete the licensed program using the `DLTLICPGM` command, and everything will be gone. The 1TMELCF licensed program, which is the TME 10 for AS/400 Endpoints, creates user profiles, and there are also objects that

the licensed program may not know about that may be created; this is why the removal process is a little bit more complicated.

The following steps should be performed to remove the AS/400 Endpoint software from an AS/400 machine.

1. Stop all Tivoli jobs or any other jobs that are started by Tivoli products and are running on the Endpoint.

Note

When you stop Tivoli jobs, you should remember the following:

- Use the `ENDTIMEEPT` command to end the Endpoint job.
- Use the `WRKACTJOB` command to find other jobs that are running under any of the Tivoli user accounts. To end a job, you can either use the `ENDJOB` command or the 4 option.
- Endpoint processes can include the `lcf` daemon process as well as any jobs started by Tivoli products. For example, you must stop the Distributed Monitoring engine and any monitor processes running on the AS/400 Endpoint.

2. Delete the Licensed Program by entering the following command:

```
DLTLICPGM LICPGM(1TMELCF)
```

3. Execute the following `WRKOBJOWN` commands on each Tivoli user account to display all objects owned by the specified user:

```
WRKOBJOWN USRPRF(QTIVUSER)
WRKOBJOWN USRPRF(QTIVROOT)
WRKOBJOWN USRPRF(QTIVOLI)
```

4. The `WRKOBJOWN` command lists all objects that belong to a user. To prevent any listed object from being deleted, you must change the ownership of that object by selecting option **9** for the object from the *Work with Objects by Owner* display. Repeat this step for each object that you want to save.
5. Delete the user account names by specifying the `*DLT` option, which deletes all objects owned by a user. The `DLTUSRPRF` commands must be entered in the following order:

```
DLTUSRPRF USRPRF(QTIVUSER) OWNBJOPT(*DLT)
DLTUSRPRF USRPRF(QTIVROOT) OWNBJOPT(*DLT)
DLTUSRPRF USRPRF(QTIVOLI) OWNBJOPT(*DLT)
```

Note

If you delete QTIVOLI before deleting other user accounts, some objects may not be deleted, and you might have problems reinstalling the Endpoint on the AS/400 system.

3.6 AS/400 Endpoint installation comparison

The installation methods just described can be taken into an environment and used, but, you should keep in mind that each of the methods has strong points that should be considered when deciding on a method of installation. In the following sections, we describe the benefits of using each of these methods.

3.6.1 Benefits of using the w4inslcf.pl script

Installing AS/400 Endpoints using the `w4inslcf.pl` script is a widely-used method of installation. This method is mostly used in environments that either do not have the system resources to support a high-end graphical environment (using SIS), or have only one or two machines to install. This script (and all supporting scripts) are installed when applying the 3.6.1-TMF-0001 patch; therefore, no additional patches or products need to be installed to use this installation method. It is quick and extremely straightforward.

Even though this installation method is extremely useful in environments where a small number of AS/400 Endpoints will be installed, there is also an option for the script that will allow a list of machines that are to be installed to be retrieved from a flat file. This makes mass installations much easier.

To summarize this installation method, one could conclude that it is much more suitable for an environment that does not already use SIS on a regular basis. This is because of the time saved by not using the graphical interface.

3.6.2 Benefits of using the Software Installation Service (SIS)

The Software Installation Service (SIS) is a graphical interface for installing Tivoli products and managed resources. SIS keeps an installation repository (IR), which is a massive collection of all the source media for Tivoli products. This makes the installation of numerous products to numerous machines easy.

A drawback to using SIS while installing AS/400 Endpoints, or any Endpoints for that matter, is that no other Tivoli products will be applied to the Endpoint itself during installation. Therefore, there are not too many benefits for using SIS to install the Endpoint itself. However, the SIS installation engine can install numerous Endpoints in parallel, while the `w4inslcf.pl` script does the installations serially (one at a time). Using SIS in these large-scale environments saves time and network resource usage.

The major benefits of using SIS in an Endpoint deployment would be for creating your Endpoint Gateways that manage AS/400 Endpoints. This is because a Tivoli administrator can create a Managed Node, make it an Endpoint Gateway, and install AS/400 enablement patches to Endpoint Gateways and all product patches, all in the same operation. Again, installations can be done in parallel; therefore, the installation time for this operation can be dramatically reduced.

3.7 Planning for AS/400 Endpoint mass installation

When preparing for a mass installation of AS/400 Endpoints, you should first decide what installation method will be best for your environment. This is the most important part of the process.

One thing to consider about the `w4inslcf.pl` script is that it accepts an option to pass the names (usernames and passwords) of machines that you wish to install. You can also pass a `-y` option, which will bypass all questions. Log files are always created with failed installation attempts; therefore, if you do fifty AS/400 Endpoint installations at one time, you can refer to the log file for a list of machines that failed. This list is quite useful because you can then pass that same file back later into the `w4inslcf.pl` script for installation to those machines only (seeing that your other machines were successful installations).

The `w4inslcf.pl` script does installations to machines one at a time in the order they were passed in the file or in the order they were retrieved from a multiple host-spec on the command line. This can cause for a very lengthy installation if many machines are provided, but only uses the amount of bandwidth needed to communicate with one machine at a time, which can preserve much of your network for other uses besides AS/400 Endpoint installations.

On the other hand, SIS can install to many machines at one time (in parallel). This can congest many networks, but, at the same time, it can provide for a very quick installation. SIS does not create single file of failed machines that

is usable for import later, which can be a problem. It is also somewhat difficult and time-consuming to look in all the log files of the installation to determine which machines must be reinstalled. This is the reason that the `w4inslcf.pl` script might want to be used.

All in all, both methods of installation in large-scale environments have their faults and benefits, but the decision really has to do with your particular environment's needs.

Chapter 4. AS/400 Endpoint configuration

In the Tivoli three-tiered management structure, the Endpoint must be configured to communicate with an appropriate Endpoint Gateway. To allow for both flexibility and availability, the association between the Endpoint and the Endpoint Gateway can be dynamically determined and may change if a particular Endpoint Gateway is not available. Tivoli has extended its management capability to the AS/400 platform, and these configuration methods of Endpoint and Endpoint Gateway are applied to AS/400 Endpoints as well as other Endpoints running on other platforms, such as UNIX and Windows NT.

In this chapter, we describe how you can control the AS/400 Endpoint behavior and configure it in your Tivoli management environment.

4.1 Understanding the Endpoint login process

When the AS/400 Endpoint is started, it performs the Endpoint login process to participate in the TMR. There are two types of logins that the AS/400 Endpoint performs: normal login and initial login. The following sections describe each Endpoint login process.

4.1.1 Endpoint normal login

When the AS/400 Endpoint has already been a member of the TMR, the AS/400 Endpoint performs the *normal login* upon start-up. The normal login sequence is quite simple. The AS/400 Endpoint logs in to its assigned Gateway, and the Endpoint Gateway acknowledges it. During normal login, communication only takes place between the AS/400 Endpoint and the Endpoint Gateway. The AS/400 Endpoint sends subsequent login packets and communicates directly to the Endpoint Gateway listed in the login information (lcf.dat) file. Since the Endpoint Gateway has the Endpoint's information in its Endpoint list, communications are established immediately without contacting the TMR server or the Endpoint Manager.

Figure 16 on page 62 shows the normal login procedure.

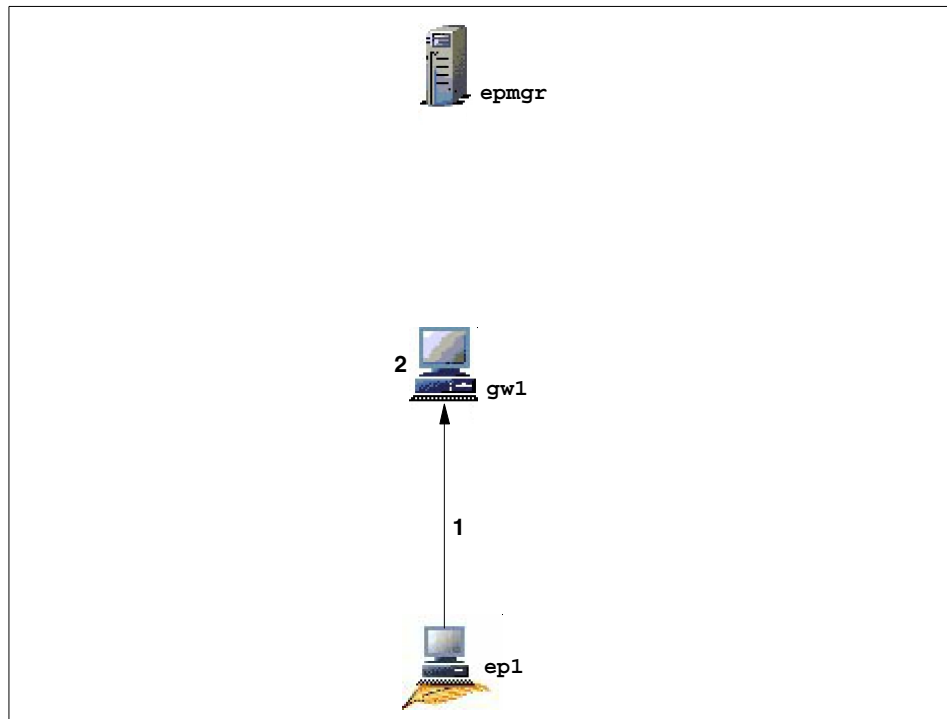


Figure 16. Endpoint initial login

The following events occur

1. The AS/400 Endpoint (ep1) attempts to login to the assigned Gateway (gw1).
2. The AS/400 Endpoint completes the normal login to the assigned Gateway.

4.1.2 Endpoint initial login

As a precondition for TMA operation, the Endpoint must become a member of the TMR. This process establishes the Endpoint as an active member of its TMR by assigning it to the Endpoint Gateway. When the AS/400 Endpoint service begins to run, it attempts to establish its region identity from the lcf.dat file. If this file does not exist, the AS/400 Endpoint performs the “initial login” in order to establish that identity.

Three Step Processes

Endpoint Initial login has the following three major phases:

- The Endpoint establishes communication with the TMR.

- The Endpoint Manager selects the Endpoint Gateway for the Endpoint.
- The Endpoint receives its Gateway assignment information and performs the normal login to the assigned Gateway.

The following figure (Figure 17) shows the Endpoint initial login process flow.

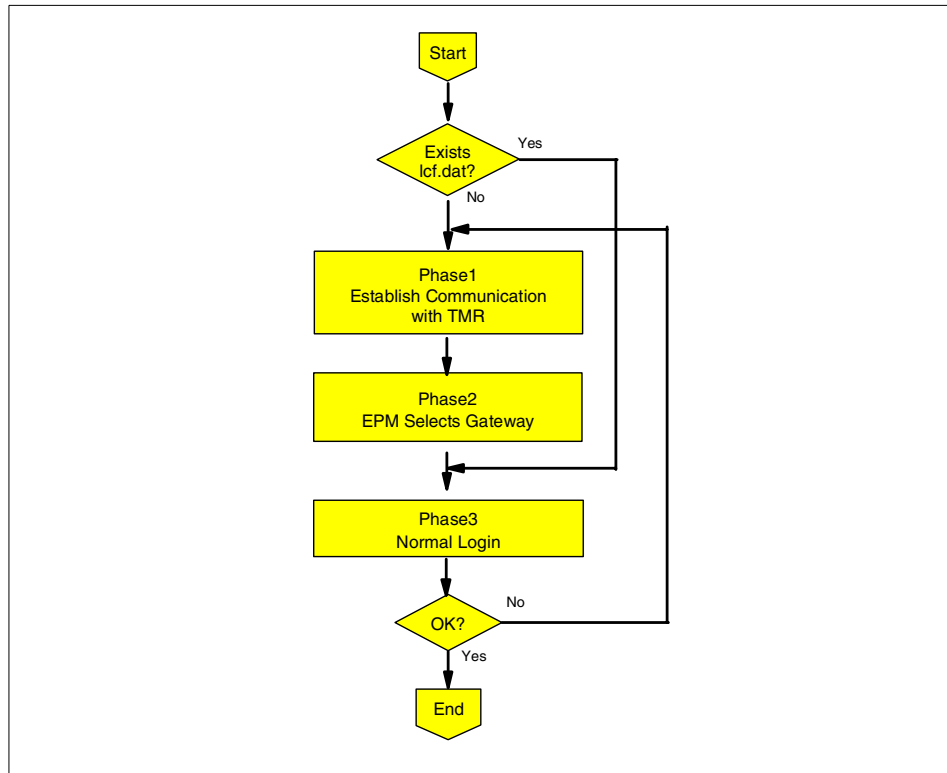


Figure 17. Endpoint initial login processes

The following example (Figure 18) shows the typical initial login procedure.

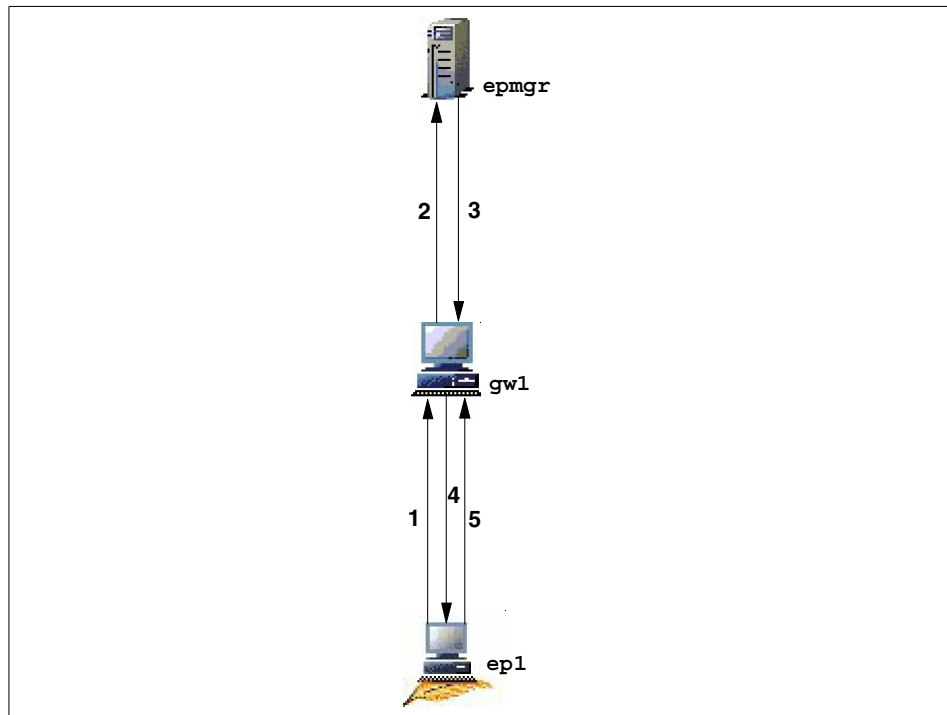


Figure 18. Endpoint initial login

1. The AS/400 Endpoint (ep1) attempts to perform the initial login to the Endpoint Gateway (gw1) that is specified by AS/400 Endpoint option (LGNINTRFC).
2. The Endpoint Gateway forwards the login request to the Endpoint Manager.
3. Having no defined policy, the Endpoint Manager assigns the Endpoint to the intercepting Gateway and sends the new login information to the Endpoint Gateway.
4. The Endpoint Gateway relays the login information to the Endpoint.
5. The Endpoint logs into its assigned Gateway, gw1.

These Endpoint login process is complex, but highly configurable and flexible. Its features and failure modes must be studied carefully before attempting any sort of rollout. Please refer to the *All About Tivoli Management Agents*, SG24-5134, for more detailed information about Endpoint login process.

4.1.3 Region redirect

The region redirect feature is a special case of the initial login. During the Endpoint Gateway selection process, it is possible for the user to specify the Endpoint Gateway in another interconnected TMR. If it does, the Gateway assignment is returned with a special status to tell the Endpoint to start the initial login process using the specified Endpoint Gateway as the new intercepting Gateway. By maintaining a well-known redirector TMR, you can gain some flexibility in Endpoint installations. We will describe more detailed information about TMR redirection later on in this chapter.

4.1.4 Isolated login

When the Endpoint attempts to perform the normal login to its assigned Gateway, for example, it may happen that the assigned Gateway is unreachable. In this case, the Endpoint is said to be isolated. The Endpoint falls back to the initial login process of connecting to the TMR via the login interfaces determined at initial login. This is similar to the initial login except that for isolation, the Endpoint already exists. Therefore, the new Endpoint identity is not established. The Endpoint Manager simply performs the Endpoint Gateway selection and returns the Endpoint Gateway assignment to the Endpoint through the intercepting Gateway.

The following example (Figure 19) shows the typical isolated login process.

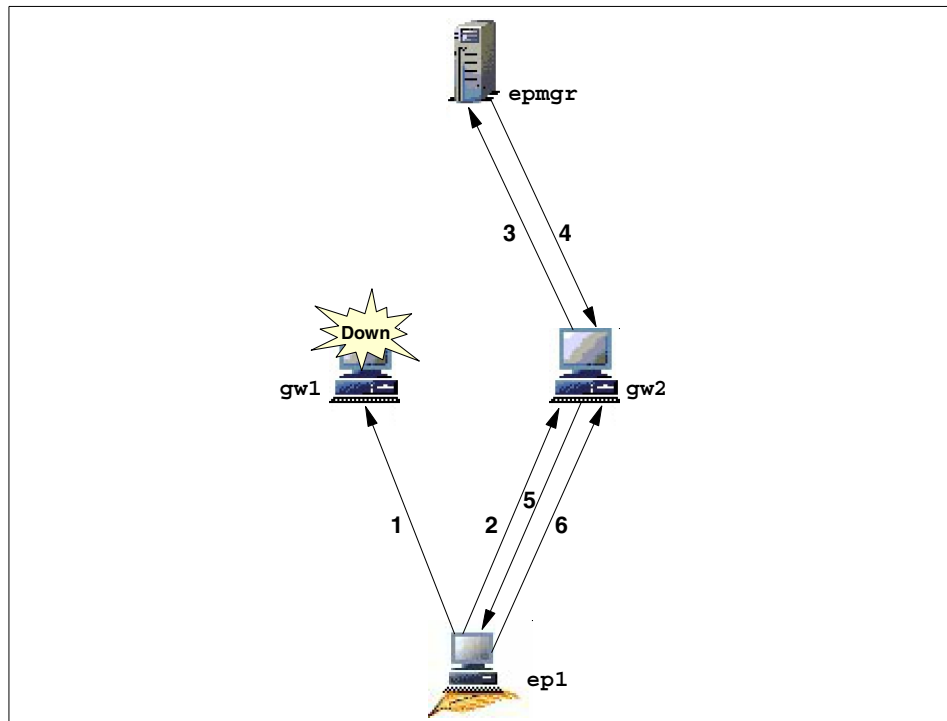


Figure 19. Endpoint isolated login

1. The Endpoint (ep1) attempts to perform the normal login to the assigned Gateway (gw1). However, the assigned Gateway (gw1) is not available. The Endpoint (ep1) attempts to perform the normal login again until the timeout occurs.
2. After the timeout, the Endpoint (ep1) is considered isolated. Then, the Endpoint (ep1) attempts to send the login request to the Endpoint Gateway (gw2), which is contained in the login interfaces list of the lcf.dat file.
3. The Endpoint Gateway (gw2) forwards the login request to the Endpoint Manager.
4. Having no defined policy, the Endpoint Manager assigns the Endpoint (ep1) to gw2 and sends the new login information to the intercepting Gateway (gw2).
5. The intercepting Gateway (gw2) relays the login information to the Endpoint (ep1).
6. The Endpoint (ep1) logs into its assigned Gateway (gw2).

2. After the timeout, the Endpoint (ep1) is considered isolated. Then, the Endpoint (ep1) attempts to send the login request to the Endpoint Gateway (gw2), which is contained in the login interfaces list of the lcf.dat file.

3. The Endpoint Gateway (gw2) forwards the login request to the Endpoint Manager.

4. Having no defined policy, the Endpoint Manager assigns the Endpoint (ep1) to gw2 and sends the new login information to the intercepting Gateway (gw2).

5. The intercepting Gateway (gw2) relays the login information to the Endpoint (ep1).

6. The Endpoint (ep1) logs into its assigned Gateway (gw2).

Refer to the book, *All About Tivoli Management Agents*, SG24-5134, for more information about the isolated login.

4.2 Understanding Endpoint policy

You can configure the Endpoint's login behavior and communication patterns by developing scripts that execute at various times in the process. There are four such hooks. Of these, the `login_policy` runs on the Endpoint Gateway, and the other three run at the Endpoint Manager. These are `allow_install_policy`, `select_gateway_policy`, and `after_install_policy` (see Figure 20).

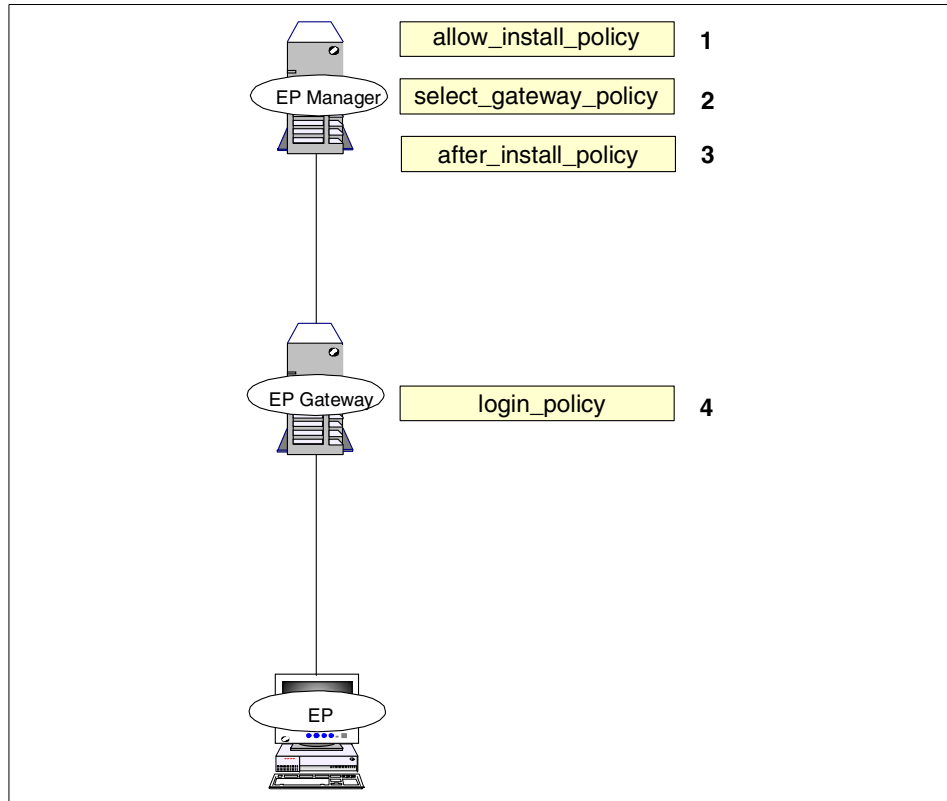


Figure 20. Endpoint policies

We will describe these Endpoint policies in the order in which they execute.

4.2.1 allow_install_policy

The `allow_install_policy`, or `install policy` for short, allows you to terminate the login immediately when the Endpoint Manager receives the login request from the intercepting Gateway. For example, you can decide to refuse the login request based on the Endpoint's IP address. To do this, simply exit the script with a non-zero value. You can also use this policy to perform any pre-login actions you might need.

4.2.2 select_gateway_policy

The `select_gateway_policy` (SGP for short) is unique in that it is the only Endpoint policy script that is specified to produce output. The output is a list of Endpoint Gateway object references that is used by the Endpoint Manager to make the Gateway assignment for that Endpoint. The Endpoint Manager will assign the Endpoint to the first available Endpoint Gateway. This same list is also returned to the Endpoint for use later if the assigned Gateway is unavailable. In this later context, the Endpoint Gateway list is called the login interfaces list.

The Endpoint policy overrides the Endpoint Manager's default selection process, and it is recommended to use the Endpoint policy in multiple Gateway environments. The Endpoint Manager tries to contact each Endpoint Gateway in the order listed in the Endpoint policy script. The first Endpoint Gateway that the Endpoint Manager contacts successfully is the Endpoint Gateway to which the Endpoint is assigned. The intercepting Gateway is also added to the end of the login interfaces list to ensure that the Endpoint has at least one definite contact. If the Endpoint Gateways listed in the script cannot be contacted, the Endpoint Manager assigns the intercepting Gateway to the Endpoint.

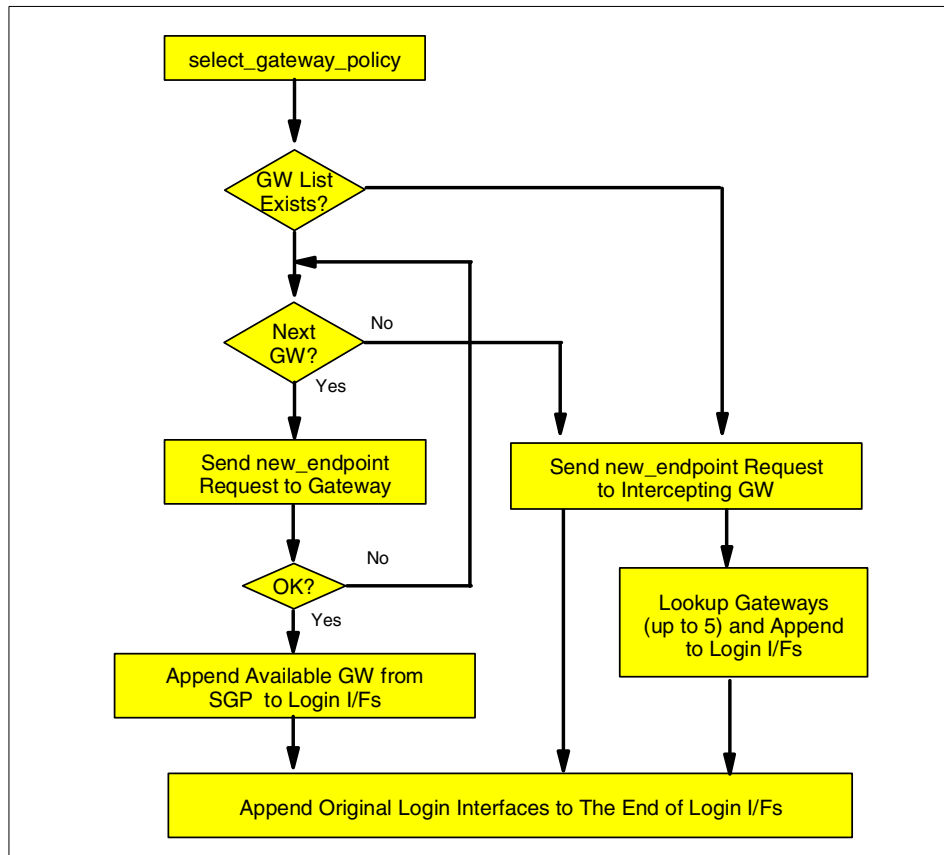


Figure 21. Gateway selection and new login interfaces

In Figure 21, we tried to clarify how the new login interfaces would be created after running the Endpoint Gateway selection process on the Endpoint Manager. Figure 22 on page 70 shows how the Endpoint Manager creates the Endpoint Gateway list.

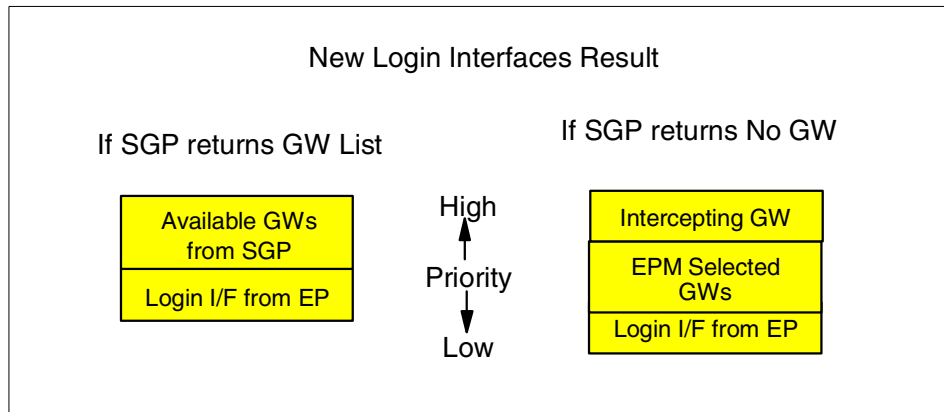


Figure 22. New login interfaces from Endpoint Manager

4.2.3 after_install_policy

The `after_install_policy` script is invoked after the Gateway assignment is made but before the Endpoint is notified. Management operations, such as a downcall, cannot be invoked on the Endpoint machine at this time. The standard example for the `after_install_policy` is to add the Endpoint to a policy region. This Endpoint policy is run after the initial login only; it will not be run on subsequent Endpoint logins.

4.2.4 login_policy

This policy is run on the Endpoint Gateway upon every normal login and performs any action you need. For example, the `login_policy` can be useful when you configure the auto upgrade function of the Endpoint software. If the `login_policy` exits with a non-zero value, the Endpoint login will not fail, but the Endpoint Gateway does not invoke the `boot_method`. The same `login_policy` script is run on all of the Endpoint Gateways in the TMR. This policy does not support the use of binaries.

4.2.5 Policy arguments for AS/400 Endpoint

The scripts defined in the Endpoint policy all have the same argument conventions and ship with a default script, `exit 0`, which means that no special policy applies. This policy and its functional equivalents are illustrated by the following excerpt from the default Endpoint policy.

```
#
# The following are the command line arguments passed to this script
# from the Endpoint Manager.
#
# $1 - The label of the Endpoint machine
# $2 - The object reference of the Endpoint machine
# $3 - The architecture type of the Endpoint machine
# $4 - The object reference of the Gateway that the Endpoint logged into
# $5 - The ip address of the Endpoint logging in.
# $6 - region
# $7 - dispatcher
# $8 - version
```

These arguments do not all exist at every point in the Endpoint login process. For example, Endpoint IP address (\$5) is always defined. However, the object reference of the Endpoint machine (\$2) is not created until after the `allow_install_policy` exits with a value of 0; so, \$2 has the value of `OBJECT_NIL` when passed to the `allow_install_policy` script.

We can run the following script to see all of these arguments in the AS/400 operating system environment.

```
#!/bin/sh
LOGFILE=/tmp/policylog
printline() {
    echo 'date +%Y/%m/%d %H:%M:%S: ' $* >> $LOGFILE
    return
}
printline "[allow_install_policy]"
printline "The label of the ep machine: $1"
printline "The object reference of the ep machine: $2"
printline "The architecture type of the ep machine: $3"
printline "The object reference of the gateway: $4"
printline "The ip address of the ep logging in.: $5"
printline "region: $6"
printline "dispatcher: $7"
printline "version: $8"
printline "LCF_LOGIN_STATUS=${LCF_LOGIN_STATUS}"
printline "Exiting ..."
exit 0
```

We show the result of this script for AS/400 Endpoint as follows.

```

1999/12/07 13:49:50: [allow install policy]
1999/12/07 13:49:50: The label of the ep machine: ishiil
1999/12/07 13:49:50: The object reference of the ep machine: OBJECT NIL
1999/12/07 13:49:50: The architecture type of the ep machine: os400-v3r7
1999/12/07 13:49:50: The object reference of the gateway: 1790075743.2.19#TMF_Gateway::Gateway#
1999/12/07 13:49:50: The ip address of the ep logging in.: 146.84.32.7+9495
1999/12/07 13:49:50: region: 1790075743
1999/12/07 13:49:50: dispatcher: 0
1999/12/07 13:49:50: version: 20
1999/12/07 13:49:50: LCF_LOGIN_STATUS=0
1999/12/07 13:49:52: Exiting ...
1999/12/07 13:49:52: [select gateway policy]
1999/12/07 13:49:52: The label of the ep machine: ishiil
1999/12/07 13:49:52: The object reference of the ep machine: OBJECT NIL
1999/12/07 13:49:52: The architecture type of the ep machine: os400-v3r7
1999/12/07 13:49:52: The object reference of the gateway: 1790075743.2.19#TMF_Gateway::Gateway#
1999/12/07 13:49:52: The ip address of the ep logging in.: 146.84.32.7+9495
1999/12/07 13:49:52: region: 1790075743
1999/12/07 13:49:52: dispatcher: 0
1999/12/07 13:49:52: version: 20
1999/12/07 13:49:52: LCF_LOGIN_STATUS=0
1999/12/07 13:49:54: Exiting ...
1999/12/07 13:49:56: [after install policy]
1999/12/07 13:49:56: The label of the ep machine: ishiil
1999/12/07 13:49:56: The object reference of the ep machine:
1790075743.299.508#TMF_Endpoint::Endpoint#
1999/12/07 13:49:56: The architecture type of the ep machine: os400-v3r7
1999/12/07 13:49:56: The object reference of the gateway: 1790075743.2.19#TMF_Gateway::Gateway#
1999/12/07 13:49:56: The ip address of the ep logging in.: 146.84.32.7+9495
1999/12/07 13:49:56: region: 1790075743
1999/12/07 13:49:56: dispatcher: 29
1999/12/07 13:49:56: version: 20
1999/12/07 13:49:56: LCF_LOGIN_STATUS=0
1999/12/07 13:49:58: Exiting ...
1999/12/07 13:50:09: [login policy]
1999/12/07 13:50:09: The label of the ep machine: ishiil
1999/12/07 13:50:09: The object reference of the ep machine:
1790075743.299.508#TMF_Endpoint::Endpoint#
1999/12/07 13:50:09: The architecture type of the ep machine: os400-v3r7
1999/12/07 13:50:09: The object reference of the gateway: 1790075743.2.19#TMF_Gateway::Gateway#
1999/12/07 13:50:09: The ip address of the ep logging in.: 146.84.32.7+9495
1999/12/07 13:50:09: region: 1790075743
1999/12/07 13:50:09: dispatcher: 29
1999/12/07 13:50:09: version: 20
1999/12/07 13:50:09: LCF_LOGIN_STATUS=
1999/12/07 13:50:11: Exiting ...

```

You can use the above arguments in your scripts running in Endpoint policies for AS/400 Endpoints.

4.2.6 Applying AS/400 Endpoint policies

The CLIs for manipulating the Endpoint policies are the `wputepool` and `wgetepool` commands. The Tivoli Management Framework is installed with default Endpoint policy scripts. They do not contain any logic, and simply return the value of zero to the Endpoint Manager or Endpoint Gateway. To implement useful Endpoint policies for your management environment, you need to replace the default Endpoint policies with your own scripts. The following steps are required to do this:

1. Retrieve the current Endpoint policy script using the `wgetepool` command.
2. Modify this script to add the appropriate logic.

3. Replace the current Endpoint policy script with the new script using the `wputteppol` command.

The `wputteppol` and `wgetteppol` commands take a single parameter specified by the Endpoint policy whose script you are retrieving or replacing. The script itself is routed to stdout for the `wgetteppol` command and read from the stdin for the `wputteppol` command. Therefore, you would normally use redirection as shown in the next example. For more information regarding the `wgetteppol` and `wputteppol` commands, refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434.

When you want to customize the `allow_install_policy`, invoke the following command:

```
wgetteppol allow_install_policy > allow_install_policy.sh
```

This command will create the file, `allow_install_policy.sh`, including the current policy script. Then, this file can be edited to include logic, such as the samples shown in this section.

After saving the modified shell script, execute the following command to enable the policy for subsequent Endpoint login processes:

```
wputteppol allow_install_policy < allow_install_policy.sh
```

All policy scripts are stored in the Tivoli object databases to make it possible to keep consistency between the Endpoint Manager and Endpoint Gateways. In this way, all Endpoint Gateways receive the latest `login_policy` script. Note that the same `login_policy` script is applied to all Endpoint Gateways in the TMR. If the logic defined in the `login_policy` depends on a specific operating system, the `login_policy` script must have the logic that determines on which platform it is executing and act accordingly.

Note

You do not have to pass the full Endpoint policy name to the `wgetteppol` and `wputteppol` commands. Instead, you can pass unique abbreviations. For example, to retrieve the `allow_install_policy`, you can use:

```
wgetteppol al
```

For the other Endpoint policies, use the following:

select_gateway_policy	<code>wgetteppol se</code>
after_install_policy	<code>wgetteppol af</code>
login_policy	<code>wgetteppol lo</code>

4.3 AS/400 Endpoint configuration files

AS/400 Endpoint as well as other Endpoints use two files to store configuration information. These files are located under the /QIBM/UserData/Tivoli/lcf directory of AS/400 Endpoint.

4.3.1 lcf.dat file

The lcf.dat file contains the login information related to the Endpoint. Since this file is a binary file, you cannot edit it directly. However, when you start the AS/400 Endpoint daemon, you can overwrite certain information in this file using the `STRIMEEPT` command parameters. Once the AS/400 Endpoint connects to its assigned Gateway, the Endpoint Gateway address, port number, any network aliases for the assigned Gateway, and alternate Gateway information are stored in the lcf.dat file.

4.3.2 last.cfg file

All other configuration information is stored in the last.cfg file. Once the Endpoint and Endpoint Gateway are connected, for example, once the initial login is completed, the configuration information is stored in the last.cfg file. For every subsequent startup, the startup command for the AS/400 Endpoint daemon (`STRIMEEPT` command) retrieves the configuration information from the lcf.dat and last.cfg files. The following is an example of the last.cfg file for an AS/400 Endpoint.

```

lcf_d_port=9495
lcf_d_preferred_port=9495
gateway_port=9494
bcast_disable=1
log_threshold=3
start_timeout=120
run_timeout=120
lcf_d_version=24
logfile=/QIBM/USERDATA/TIVOLI/LCF/lcf_d.log
config_path=/QIBM/USERDATA/TIVOLI/LCF/last.cfg
run_dir=/QIBM/USERDATA/TIVOLI/LCF
load_dir=/QIBM/USERDATA/TIVOLI/LCF/bin/os400-v3r7/mrt
lib_dir=/QIBM/USERDATA/TIVOLI/LCF/lib/os400-v3r7
cache_loc=/QIBM/USERDATA/TIVOLI/LCF/cache
cache_index=/QIBM/USERDATA/TIVOLI/LCF/cache/Index.v5
cache_limit=104857600
log_queue_size=1024
log_size=1024000
udp_interval=300
udp_attempts=6
login_interval=1800
lcs.machine_name=shaft
lcs.crypt_mode=196608
lcf_d_alternate_port=9496
local_ip_interface=0.0.0.0

```

The last.cfg file contains the most recent configuration information. You can alter this file to affect the options that will be used the next time the AS/400 Endpoint is started. For more information, refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, and the *Tivoli Framework AS/400 3.6.2 Endpoints User's Guide*, GC32-0279.

To modify the configuration of the Endpoint, you can either, replace the last.cfg file or restart the AS/400 Endpoint using the `STRIMEEPT` command with the appropriate parameters. If you start the AS/400 Endpoint from the command line interface using the `STRIMEEPT` command, the parameters you specify override the equivalent entries in the last.cfg file. If you replace the last.cfg file, the new configuration information is used when you restart the AS/400 Endpoint daemon. Once the AS/400 Endpoint performs the Endpoint login, the configuration information is stored in the last.cfg file again. You can also use the Endpoint Web interface to modify the contents of the last.cfg file. Refer to Section 5.1.2, "Working with Tivoli Files on the AS/400" on page 106, for more information about handling AS/400 files.

4.4 Configuring AS/400 Endpoint

Configuring an already installed AS/400 Endpoint is quite simple since the help dialogs are installed for the AS/400; so, by simply pressing the **F4** key

after typing the `STRIMEEPT` command, a complete dialog with available options will be prompted to the user.

Tivoli provides the following two Control Language (CL) commands to start and stop the AS/400 Endpoint daemon:

<code>STRIMEEPT</code>	This command starts the AS/400 Endpoint daemon. When installing an AS/400 Endpoint, SIS or the <code>w4instlcf.pl</code> script starts the AS/400 Endpoint daemon job by executing the <code>STRIMEEPT</code> command. You must use this command to manually start the AS/400 Endpoint daemon job. You can also use the <code>STRIMEEPT</code> command in the auto-initialization program (QSYS/QSTRUP) to start the AS/400 Endpoint daemon job after an initial program load (IPL), or reboot.
<code>ENDTIMEEPT</code>	This command ends the AS/400 Endpoint daemon, performs clean up operations on the AS/400, and disconnects from the Endpoint Gateway.

The user ID should be given authority to start or stop AS/400 Endpoint if the command is being run from QSTRUP.

These commands are native AS/400 commands that you can execute from any AS/400 command line after the Endpoint code is installed on the AS/400 machine. You can also launch both commands from a remote machine. Both commands are shipped with authority set to `PUBLIC(*EXCLUDE)`. You must grant privileges to any users who will be using these commands. Additionally, you must also authorize these users to access the following associated command processing programs:

Table 10. AS/400 commands authorization

Library	Command processing program	Authority needed
QTMELCF	QLCFSTAR	*USE
QTMELCF	QLCFEND	*USE

4.4.1 Starting AS/400 Endpoint

The `STRIMEEPT` command starts the AS/400 Endpoint daemon process for a Tivoli Endpoint. This AS/400 Endpoint daemon communicates with the Endpoint Gateway to receive and launch Endpoint methods. The AS/400 Endpoint daemon job is always submitted to the QSYS/QSYSNOMAX job queue.

Note

The `STRTMEEPT` command is equivalent to the

```
$BINDIR/./lcf_bundle/generic/lcfd.sh
```

script that is available for other platforms.

The `STRTMEEPT` command starts the QLCFD job on the AS/400 with the appropriate configuration information. To enter Endpoint configuration parameters from a prompt screen, press the **F4** key. The default value for all parameters is `*NONE`. If `*NONE` is used, the value of the parameter is set from the `last.cfg` file. If these parameters are not available in the `last.cfg` file or the `last.cfg` file is not available, the parameter values are assigned from internal default values. The following table lists and describes parameters that are available for the `STRTMEEPT` command:

Table 11. AS/400 Endpoint (`STRTMEEPT` command) options

Keyword and Description	Parameter	Value
LGNINTRFC This specifies the IP address, or host name and port number, of one or more gateways to which the endpoint will send its login packet. This option is required for the endpoint to log in to a gateway on a different subnet or to log in to a specific gateway when two or more gateways exist on a subnet.	Login interface	*NONE
	Name	Hostname
	Port	Port number
GATEWAY This specifies the name of the Tivoli Management Gateway that will be used after the endpoint has successfully logged in. If the endpoint has not previously logged in, use the <code>LGNINTRFC</code> (or <code>lcs.login_interfaces</code> option in the configuration file) to provide one or more gateways through which the endpoint may log in.	Gateway	*NONE
	Name	Hostname
	Port	Port number
BCASTDSBL This disables the UDP broadcast. If you set this option to yes, you must use the <code>lcs.login_interfaces</code> option.	Broadcast Disable	*NONE *YES or 1 to disable *NO or 0 to enable

Keyword and Description	Parameter	Value
EPTNAME This specifies the name of this endpoint. This name is displayed when the <code>wep ls</code> command is executed in the gateway.	Endpoint Name	*NONE *HOSTNAME Use the host name as the name for this endpoint. <i>endpoint_name</i> Use a valid AS/400 endpoint name
PORT This specifies the port on which the endpoint daemon (LCFD) monitors gateway communications. The default value is 9494.	Local TCP/IP Port	*NONE <i>port_number</i>
MACHINEID This identifies the endpoint.	Machine Unique ID	<i>machine_unique_id</i> Use a string that contains a unique identifier
THRESHOLD This defines the level of debug messages written to the <code>lcf.d.log</code> file. Note: Level 3 and greater logging generates a large number of messages. For troubleshooting endpoints, Tivoli recommends using Level 2.	Log Threshold	The following are valid entries: 0 No message logging 1 Minimal logging (default) 2 Tracing and moderate output 3 Data buffers and tight loops 4 Data
LOGSIZE This specifies the maximum size in bytes of the log file.	Log Size	*NONE <i>size</i> Use a value between 10240 and 10240000.
LOGFILE This specifies the name of the log file to be used for logging messages.	Log Name	*NONE <i>path_name</i> Where path name is the name of the integrated file system (IFS) file used for the log.

Keyword and Description	Parameter	Value
CACHESIZE This specifies the maximum size of the method cache. Note: Once the maximum size is reached, the least recently used methods are deleted from the cache.	Cache Size	*NONE <i>size and integer</i>
UPINTRVL This is the number of seconds between endpoint broadcast calls.	UDP Interval	*NONE <i>seconds</i>
UDPATTMPT This is the number of times an endpoint will transmit a broadcast call.	UDP Attempts	*NONE number of times where number is an integer value
STRTIMEOUT This is the amount of time in seconds before a communications timeout occurs during login.	Start Timeout	*NONE <i>seconds</i>
RUNTIMEOUT This is the amount of time in seconds before a communications timeout occurs following a successful login.	Run Timeout	*NONE <i>seconds</i>
CFGFILE This is the name of the configuration file to be used to start the endpoint.	Configuration File Name	*NONE config_file_name
RUNDIR This specifies the name of the directory from which the endpoint code will run on the AS/400.	Running Directory	*NONE directory_name

4.4.2 Comparing Endpoint options between AS/400 and others

We showed available options for the `STRTIMEEPT` command (AS/400 Endpoint). Table 12 shows other platform Endpoint options that correspond to each AS/400 Endpoint option.

Table 12. Comparing Endpoint options between AS/400 and UNIX

Endpoint option	AS/400 Endpoint	Other Endpoint
Login interface	LGNINTRFC	-g
		lcs.login_interfaces

Endpoint option	AS/400 Endpoint	Other Endpoint
Gateway	GATEWAY	lcs.gateway_address
		lcs.gateway_port (or -p)
Broadcast disable	BCASTDSBL	bcast_disable
Endpoint name	EPTNAME	-n
		lcs.machine_name
Local TCP/IP port	PORT	-P
		lcfd_port
Machine unique ID	MACHINEID	lcs.machine_unique_id
Log threshold	THRESHOLD	-d
		log_threshold
Log size	LOGSIZE	log_size
Log queue size	LOGQSIZE	log_queue_size
Log name	LOGFILE	-l
		logfile
Cache size	CACHESIZE	cache_limit
UDP interval	UPINTRVL	udp_interval
UDP attempts	UDPATTMPT	udp_attempts
Start timeout	STRTMEOU	start_timeout
Run timeout	RUNTMEOU	run_timeout
Configuration file name	CFGFILE	config_path
Running directory	RUNDIR	run_dir

As you can see, some Endpoint options are not supported by the `STRIMEEPT` command, for example, `login_interval` (1800 seconds by default). You can specify the options using the `last.cfg` file even if the `STRIMEEPT` command does not support these options. However, you cannot use these options with the `STRIMEEPT` command.

Note

In most cases, the following Endpoint daemon option setting is recommended for initial login. These recommended options can be useful for Endpoints running on other platforms, such as UNIX or Windows NT.

LGNINTRFC ep_gw1:9494, ep_gw2:9494

BCASTDSBL YES

THRESHOLD 0

The log threshold option depends on the case. If you need to perform problem determination or troubleshooting, you had better set level 2 to the log threshold. It may affect AS/400 Endpoint performance; so, level 0 or 1 is recommended if you do not have any problem with the Endpoint.

Some timeout parameters, such as UPINTRVL or UDPATTMPT, may affect AS/400 Endpoint behaviors. Refer to the *Tivoli Enterprise Performance Tuning Guide*, SG24-5392. You can obtain more detailed information about Endpoint timeouts.

4.4.3 Stopping AS/400 Endpoint

The `ENDTIMEPT` command stops the AS/400 Endpoint daemon process for a Tivoli Endpoint. The job may be on a job queue, it may be active within a system, or it may have already completed running. Spooled files for an Endpoint process that has been stopped remain in the output queue. Note, however, that the `ENDTIMEPT` command does not end any application jobs that have been started by the AS/400 Endpoint daemon; these jobs continue to run.

Use the `DELAY` option with the `ENDTIMEPT` command to specify whether the AS/400 Endpoint is to end in a controlled manner with a time delay. The following syntax specifies valid parameters for the `ENDTIMEPT` command:

```
ENDTIMEPT [OPTION(*CTRLD|*IMMED) [DELAY(30|1-9999999)]]
```

where:

- *CTRLD** Specifies the default option that enables the AS/400 Endpoint daemon to finish any pending requests within the amount of time specified by `DELAY` (in seconds). The default delay is 30 seconds.
- *IMMED** Specifies to stop the AS/400 Endpoint immediately. Use this option only when a `*CTRLD` end fails to end the job and a delay time is not used.

`DELAY` Specifies the timed delay in seconds after which the AS/400 Endpoint daemon is stopped.

4.4.4 Problems with AS/400 Endpoint

When dealing with problems with certain AS/400 Endpoints, you may have to reconfigure the machine so that it can make its initial login to the Endpoint Gateway again. This can be done quite simply with a few CLI commands. Perform the following steps:

1. The first thing that you need to do is shut down the AS/400 Endpoint by running the following command:

```
ENDTIMEPT *IMMED
```

2. After the AS/400 Endpoint daemon has ended, you can remove a few of the configuration object links. These links include:

```
* /QIBM/UserData/Tivoli/lcf/last.cfg
* /QIBM/UserData/Tivoli/lcf/lcf.dat
* /QIBM/UserData/Tivoli/lcf/lcfd.log
```

3. After removing those files, you can run the `STRTIMEPT` command and press the **F4** key. You will then be prompted for the options.

This will force your AS/400 to make an initial connection back to the TMR server and your Endpoint Gateway. Because of this, you will need to make sure that your AS/400 Endpoint does not already exist (and have an object) in the TMR server. You can do this by running:

```
wep ls | grep -i ep_label
```

If the machine exists in your list, you can run the following command:

```
wdelep ep_label
```

This will remove the Endpoint from the TMR server's Endpoint Manager. After restarting the Endpoint daemon on the AS/400, you should have a new object created on the TMR server for that Endpoint.

Note

In this case, the AS/400 Endpoint daemon code remains on the AS/400 machine. This means the Endpoint daemon level is kept. Therefore, when you perform this way, you do not need to upgrade Endpoint daemon code again by running the `wadminep` command.

This is very convenient, especially for testing AS/400 Endpoint, and is very different from reinstalling AS/400 Endpoint.

4.5 Customizing Endpoint policies

In this section, we introduce some useful examples of the AS/400 Endpoint policies for your reference.

4.5.1 Example of allow_install_policy

The following is an example of the allow_install_policy script. This example does not allow the Endpoints on the subnet 9.3.2 to log in to the TMR, nor does it allow the Endpoints that do not have a hostname to perform the Endpoint login.

```
#!/bin/sh
set -e
#
# Don't allow endpoints from subnet 9.3.2 to log into this TMR.
#
SUBNET='echo $5 | awk -F"." '{ print $1"."$2"."$3 }'
if [ "$SUBNET" = "9.3.2" ]; then
    exit 1
fi
#
# Don't allow endpoints that do not have a name
#
if [ "$1" = "" ]; then
    exit 1;
fi
exit 0
```

The following example allows only the Endpoints found in the file, eplist.txt, to perform the Endpoint login.

```
#!/bin/sh
#
# Allow only the endpoints found in eplist.txt
. /etc/Tivoli/setup_env.sh
while read x
do
    if [ $1 = $x ]; then
        exit 0
    fi
done < $DEDIR/eplist.txt
exit 1
```

These examples are very useful when you test the Endpoints.

4.5.2 Example of select_gateway_policy

The following example returns all Endpoint Gateways that are on the same subnet as the Endpoint.

```
#!/bin/sh
#
# only ep_ip is needed for this example
ep_ip=$5
#
FOUNDONE=FALSE
# we just want the subnet of the endpoint
SUBNET='echo $ep_ip | cut -d'.' ' -f3'
# get all gateways and find ones that are on the same subnet
GATEWAYS='wlookup -ar Gateway -o'
for gwoid in $GATEWAYS
do
    gwproxy='idlattr -tg $gwoid proxy Object'
    mnips='wifconfig -h $gwproxy | grep -v Device | awk '{print $2}''
    # a managed node might have multiple interfaces, so check
    # each of them if the gateway subnet matches the endpoint
    # subnet, return gwoid if it matches
    for ip in $mnips
    do
        wsub='echo $ip | cut -d'.' ' -f3'
        if [ $gwsb -eq $SUBNET ]; then
            # echo $gwsb such as '1189622596.4.21#TMF_Gateway::Gateway#'
            echo $gwoid
            FOUNDONE=TRUE
        fi
    done
done
# if you did not find a gateway, and you still want the
# endpoint to log in, exit 0, else exit 1
if [ "$FOUNDONE" = "TRUE" ]; then
    exit 0
else
    exit 1
fi
```

4.5.3 Example of after_install_policy

The following example subscribes a new Endpoint to a profile manager that represents the Endpoints that are of similar architecture type. If the policy region or profile manager does not exist, this after_install_policy creates them.

```

#!/bin/sh
#
LCF_POLICY_REGION=LCF-Endpoints
PROFILE_MANAGER=LCF-$3
EP=$1
#
# Check to see if our top-level policy region already
# exists.If not create it and put it on this administrators
# desktop.
#
# Disable "exit on error" for this call since we will handle
# the failure.
#
set +e
wlookup -r PolicyRegion $LCF_POLICY_REGION > /dev/null
ERR=$?
set -e
if [ $ERR -ne 0 ]; then
    ALI='objcall 0.0.0 get_security_objid'
    set `objcall $ALI get_identity`
    ADMIN="$1"
    ADMIN_OID="$2"
    wcrtptr -m ProfileManager -a $ADMIN $LCF_POLICY_REGION
    idlcall $ADMIN_OID refresh_collection
fi

#
# Check to see if our interp specific profile manager
# already exists. If not create it and make it dataless so
# that we can subscribe the endpoint to it.
#
# Disable "exit on error" for this call since we will handle
# the failure.
#
set +e
wlookup -r ProfileManager $PROFILE_MANAGER > /dev/null
ERR=$?
set -e
if [ $ERR -ne 0 ]; then
    wcrtpfmgmr $LCF_POLICY_REGION $PROFILE_MANAGER
    wsetpm -d /Library/ProfileManager/$PROFILE_MANAGER
fi
#
# Subscribe the endpoint to the profile manager which
# contains the endpoints for that specific interp type.
#
wsub /Library/ProfileManager/$PROFILE_MANAGER @Endpoint:$EP

exit 0

```

After running this policy, the Endpoint will be subscribed to a profile manager automatically.

4.5.4 Example of login_policy

The following example sends a Tivoli notice to the LCF-Endpoints notice group every time the Endpoint logs into the Endpoint Gateway and automatically upgrades the Endpoint software (AS/400 Endpoint daemon).

```
#!/bin/sh
#
# Invoke the upgrade script to check the current version of
# the endpoint software and upgrade if necessary.
BO='objcall 0.0.0 self'
OS='objcall $BO getattr oserv'
INSTALLDIR='objcall $OS query install_dir|tr '\\\\' '\\\'
$INSTALLDIR/lcf_bundle/upgrade/upgrade.sh $1 $8 $3
#
LCF_NOTICE_GROUP=LCF_Endpoints
#
# Send a notice to LCF endpoint notice group every time this
# endpoint logs in.
#
set +e
wlookup -r TMF_Notice $LCF_NOTICE_GROUP > /dev/null
ERR=$?
set -e
if [ $ERR -ne 0 ]; then
    NTFGM='wlookup -r Classes TMF_Notice'
    idlcall -T top $NTFGM \
        TMF_Notice::NoticeManager::create_notice_group \
        '$LCF_NOTICE_GROUP' " 72"
fi
GW='idlcall $4 _get_label'
EPOID='wlookup -o -r Endpoint $1'
wsndnotif $LCF_NOTICE_GROUP Notice << LCF_NOTICE
Endpoint $1 ($EPOID) of interp type $3, logged into gateway
$GW ($4).
LCF_NOTICE
exit 0
```

After running this policy, the Tivoli notice will appear in the appropriate notice group. The following example simply logs the Endpoint login information to the log file (eplogin.log file).

```
#!/bin/sh
#
. /etc/Tivoli/setup_env.sh
LOGFILE=$DEDIR/eplogin.log

printline() {
    echo `date +%Y/%m/%d %H:%M:%S` ` $*` >> $LOGFILE
    return
}

printline "Endpoint $1 ($5) logs in"
exit 0
```

4.6 Endpoint migration

You can change the Gateway assignment made at initial login. When the Endpoint changes its Gateway assignment from Endpoint Gateway A to Endpoint Gateway B, it is said to have migrated from A to B. Use the `wep` command to reassign the assigned Gateway as follows.

```
wep ep_label migrate gw_label
```

where

`ep_label` Specifies the Endpoint to be migrated.

`gw_label` Specifies the Endpoint Gateway to which the Endpoint is reassigned.

Please refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more detailed information about the `wep` command.

The command to migrate the assigned Gateway prompts the Endpoint Manager to update the Endpoint list on both the newly-assigned Gateway and the formerly-assigned Gateway. For stability reasons and because the Endpoint may not be reachable at the desired migration time, the Endpoint is not contacted by default at the time of the migration; so, the Endpoint Gateways must expect to receive the normal login requests from the Endpoint that they no longer manage. When this happens, the formerly-assigned Gateway acts as an intercepting Gateway during the initial login, obtaining the new Gateway assignment from the Endpoint Manager and relaying this to the Endpoint, which then performs the normal login to the new Endpoint Gateway. The Endpoint is said to be *migrating* until it learns of its new Gateway assignment. The actions undertaken by the Endpoint during this time are called *migratory*. When the Endpoint learns of its new Gateway assignment, the migration is said to be *complete*.

Figure 23 shows the process flow of the Endpoint Gateway migration.

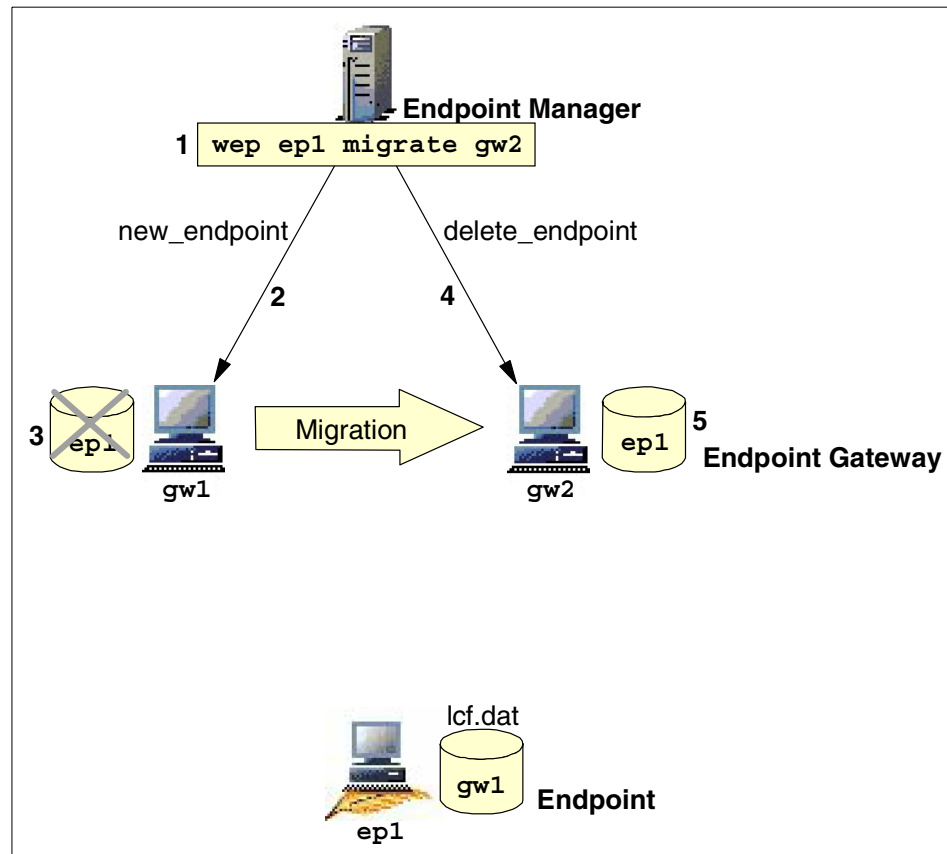


Figure 23. Endpoint Gateway migration

Perform the following steps for an Endpoint Gateway migration:

1. Execute the `wep migrate` command.
2. The `delete_endpoint` method is invoked.
3. Delete the Endpoint from the original Endpoint Gateway's cache.
4. The `new_endpoint` method is invoked.
5. Add the new Endpoint to the new Endpoint Gateway's cache.

As you can see, during the Endpoint migration process, there is no communication with the target Endpoint itself.

4.6.1 Endpoint migration completion

The Endpoint can be told of its new Gateway assignment at the time of migration. By default, the Endpoint does not immediately discover its new Gateway assignment. The Endpoint can make the discovery in one of the following ways:

- Performing a management operation (upcall)
- Performing a management operation (downcall)
- Endpoint login, such as when the Endpoint is rebooted

For example, `wep set gateway -e ep_label` is a good way to update the Endpoint after migration. Since it is basically a downcall, it is called a migration completion by downcall. The following sections describe each case of migration completion.

4.6.1.1 Migration completion by migratory login

The handling of the login request from the migrated Endpoint to its formerly assigned Gateway is similar to the case of completion by migratory upcall. The former Gateway intercepts the Endpoint's login request; then, it obtains the new Gateway assignment for the Endpoint from the Endpoint Manager. This information is forwarded to the Endpoint. The Endpoint then performs the normal login to its newly-assigned Gateway.

The example shown in Figure 24 on page 90 shows the typical procedure of migration completion by migratory login.

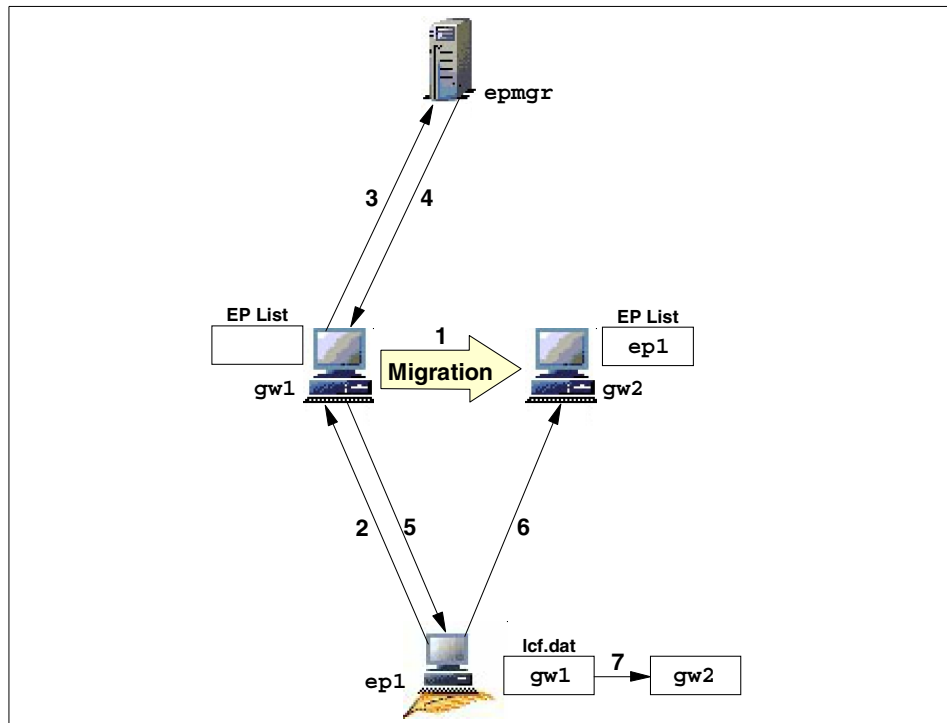


Figure 24. Migration completion by migratory login

Perform the following steps to carry out a migration completion by migratory login:

1. The Endpoint migration is performed with the `wep ep1 migrate gw2` command.
2. Since the Endpoint still keeps the previous Gateway assignment information, the Endpoint attempts to perform the normal login to the formerly-assigned Gateway (gw1).
3. The formerly-assigned Gateway (gw1) receives the login request from the Endpoint (ep1), but the formerly-assigned Gateway (gw1) no longer manages the Endpoint (ep1); so, the Endpoint Gateway (gw1) obtains the new Gateway assignment from the Endpoint Manager.
4. The new Gateway assignment information is returned to the formerly-assigned Gateway (gw1). In this case, the Endpoint Gateway selection process does not occur.
5. The formerly-assigned Gateway (gw1) forwards the new assignment information to the Endpoint (ep1).

6. Then, the Endpoint (ep1) recognizes the assigned Gateway is changed by the migration and obtains the new assignment information. The Endpoint (ep1) attempts to perform the login to the newly-assigned Gateway (gw2).
7. After the Endpoint login completes, the Endpoint (ep1) modifies its assigned Gateway information stored in the DAT (lcf.dat) file.

4.6.1.2 Migration completion by migratory upcall

An upcall is when the management application running on the Endpoint invokes a method on the Managed Node hosting its Endpoint Gateway. For security reasons, the method request passes through the Endpoint service that forwards it to the Endpoint Gateway. When the Endpoint is migrating, it sends the request to the formerly-assigned Gateway, which is considered the intercepting Gateway for the upcall because this Endpoint Gateway no longer manages the Endpoint. The intercepting Gateway determines the current Gateway assignment for the Endpoint and returns this to the Endpoint. The Endpoint performs the normal login and resends the upcall request to the new assigned Gateway. The Endpoint Gateway selection process does not occur at the Endpoint Manager because the migration will normally have occurred recently.

Figure 25 on page 92 shows the typical procedure of migration completion by migratory upcall.

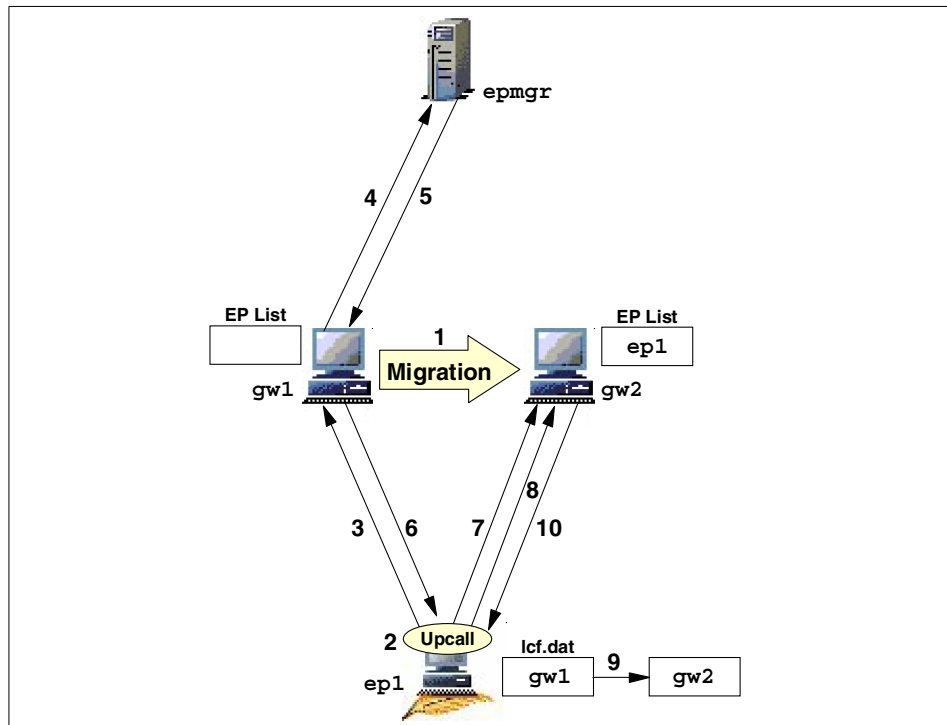


Figure 25. Migration completion by migratory upcall

Perform the following steps to carry out a migration completion by migratory upcall:

1. The Endpoint migration is performed using the `wep ep1 migrate gw2` command.
2. In this situation, the Endpoint issues an upcall to the Endpoint Gateway.
3. Since the Endpoint (ep1) still keeps the previous Gateway assignment information, the Endpoint attempts to issue the upcall to the formerly-assigned Gateway (gw1).
4. The Endpoint Gateway (gw1) that receives the upcall no longer manages the Endpoint (ep1); so, the formerly-assigned Gateway (gw1) attempts to obtain the new Gateway assignment from the Endpoint Manager.
5. The new Gateway assignment information is returned to the formerly-assigned Gateway (gw1). In this case, the Endpoint Gateway selection process does not occur.
6. The formerly-assigned Gateway (gw1) forwards the new assignment information to the Endpoint (ep1).

7. Next, the Endpoint (ep1) recognizes that the assigned Gateway is changed by the migration and obtains the new assignment information. The Endpoint (ep1) attempts to perform the login to the newly-assigned Gateway (gw2).
8. The Endpoint (ep1) also sends the upcall request to the newly-assigned Gateway (gw2).
9. After the Endpoint login completes, the Endpoint (ep1) modifies its assigned Gateway information stored in the DAT (lcf.dat) file.
10. Then, the Endpoint (ep1) receives the result of the upcall from the newly-assigned Gateway (gw2).

4.6.1.3 Migration completion by migratory downcall

A downcall is when the management application running on the Endpoint Gateway host invokes a method on the Endpoint. This means the Endpoint Gateway is able to automatically receive the migration information from the Endpoint Manager so that a downcall issued by the newly-assigned Gateway reaches the Endpoint without a problem. For each downcall, the Endpoint checks the peer address of the requesting Endpoint Gateway. If the address has changed, the Endpoint makes a note of this and rewrites its lcf.dat file. The subsequent Endpoint logins and upcalls will proceed normally. If the address change resulted from a migration, the Endpoint is able to avoid the failover logic description, which we discussed in Section 4.6.1.1, “Migration completion by migratory login” on page 89 and Section 4.6.1.2, “Migration completion by migratory upcall” on page 91.

Figure 26 on page 94 shows the typical procedure of migration completion by migratory downcall.

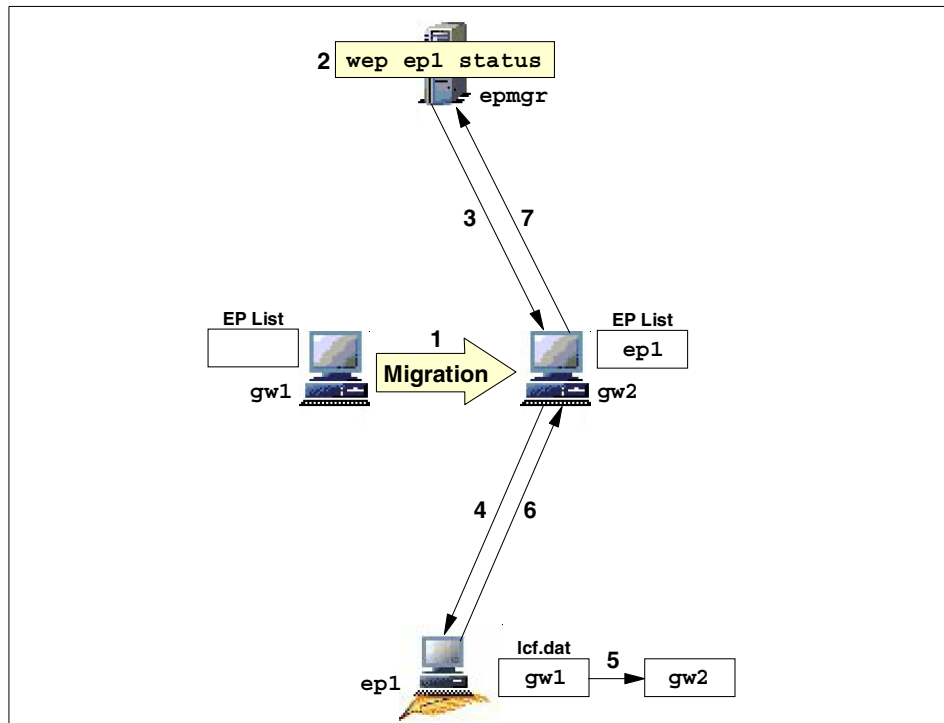


Figure 26. Migration completion by migratory downcall

Perform the following steps to carry out a migration completion by migratory downcall:

1. The Endpoint migration is performed using the `wep ep1 migrate gw2` command.
2. To issue a downcall, the `wep ep1 status` command is invoked on the Endpoint Manager.
3. The remote invoke request is sent to the newly-assigned Gateway (gw2).
4. The newly-assigned Gateway (gw2) issues a downcall to the Endpoint (ep1).
5. Then, the Endpoint (ep1) recognizes the assigned Gateway is changed by the migration and modifies its assigned Gateway information stored in the DAT (lcf.dat) file.
6. The Endpoint returns the result of the downcall to the newly-assigned Gateway (gw2).

7. The newly-assigned Gateway (gw2) also forwards the result of the downcall to the Endpoint Manager.

4.7 TMR redirection

TMR redirection is one of the solutions for managing multiple TMR environments, and it makes Endpoint configuration easy. In this section, we introduce TMR redirection and discuss its advantages.

4.7.1 TMR redirectors

Version 3.6.1 of the Tivoli Management Framework supports the feature known as TMR redirection. This feature allows you to set up a special TMR to function as the master router for the Endpoint logins across many regions.

4.7.2 How it works

For the Endpoint logging in, the normal `select_gateway_policy` script outputs Endpoint Gateway object references as candidates for the Endpoint Gateway. When you use TMR redirection, nothing changes. Simply output Endpoint Gateway object references for Endpoint Gateways in the region of your choice as follows:

```
echo `wlookup -r Gateway gw_label`
```

The Endpoint Manager understands that these references are to Endpoint Gateways in the remote region and will behave accordingly. The only requirement is that the redirector Gateway must be interconnected to all regions referenced in the policy.

Upon the completion of the `select_gateway_policy`, when the Endpoint Manager detects that remote object references have been specified, it simply rewrites the Endpoint login interfaces and sets a special flag to tell the Endpoint about the redirection. This data is relayed back to the Endpoint, which then behaves as if it had been restarted with the `-g x` option, where `x` is the `login_interfaces` specified by the redirecting `select_gateway_policy`.

Note that Endpoint Gateway selection does not actually occur. The Endpoint simply gets a new set of Endpoint Gateways to use as interceptors in the (second round) initial login process. Once the Endpoint contacts the new region, the initial login process proceeds as described. In particular, any `select_gateway_policy` in the new region is honored.

If this policy is trivial, the described default behavior makes the intercepting Gateway the assigned Gateway. In this way, one can manage all Gateway

selections from a single policy script. Moreover, one can create the hierarchy of redirectors to benefit from delegation. You can also create redirection cycles; so, be careful.

Figure 27 shows the typical procedure of TMR redirection.

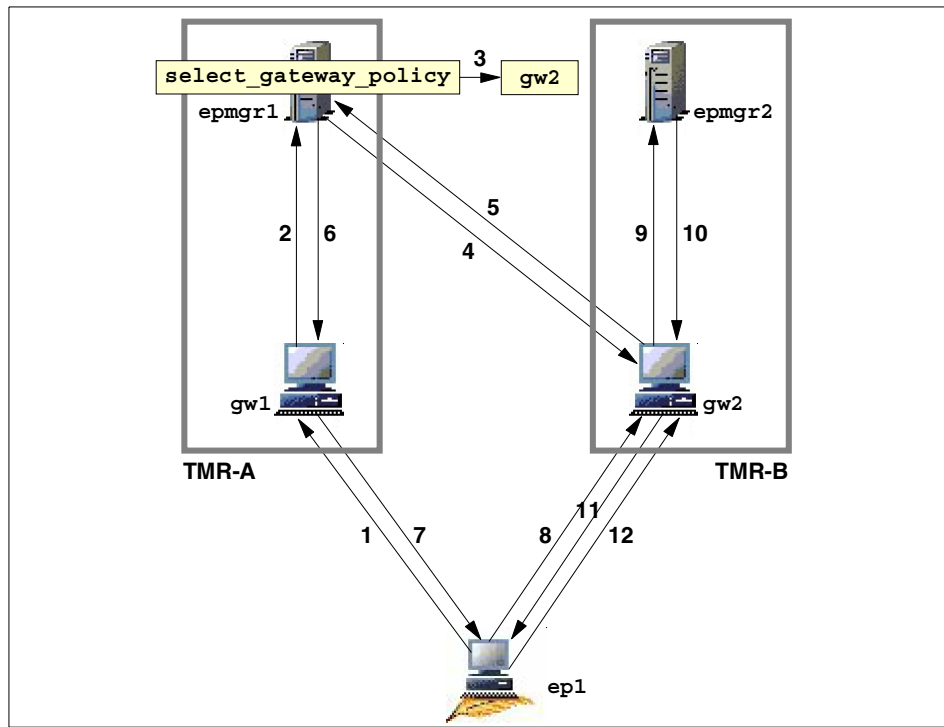


Figure 27. TMR redirection

Perform the following steps to carry out a TMR redirection:

1. The Endpoint (ep1) attempts to perform the initial login to the Endpoint Gateway (gw1) specified first by the AS/400 Endpoint option (LGNINTRFC).
2. The Endpoint Gateway (gw1) forwards the login request to the Endpoint Manager (epm1).
3. The Endpoint Manager (epm1) receives the login request from the Endpoint (ep1), then refers to the `select_gateway_policy` defined in TMR-A and gets the candidate for the assigned Gateway. The Endpoint Gateway (gw2) that the Endpoint Manager (epm1) finds belongs to TMR-B so that the Endpoint Manager (epm1) then recognizes this login process as TMR redirection.

4. The Endpoint Manager (epm1) sends the request for retrieving the interface information of the Endpoint Gateway (gw2).
5. The Endpoint Manager (epm1) retrieves the Endpoint Gateway's (gw2) interface information.
6. The Endpoint Manager (epm1) sends the network interface information of the Endpoint Gateway (gw2), with directions to use it as the intercepting Gateway (gw2), to the original intercepting Gateway (gw1).
7. The Endpoint Gateway (gw1) relays the information to the Endpoint (ep1).
8. The Endpoint (ep1) also recognizes that the login request is redirected and begins the initial login process again with the new information. The login request is intercepted by the Endpoint Gateway (gw2) as designated in the `select_gateway_policy` script in the TMR-A.
9. The login request is forwarded to the Endpoint Manager (epm2) in TMR-B.
10. Having no defined Endpoint policy in TMR-B, the Endpoint Manager (epm2) assigns the Endpoint (ep1) to the intercepting Gateway (gw2) and sends the login information to the intercepting Gateway (gw2).
11. The intercepting Gateway (gw2) relays the login information to the Endpoint (ep1).
12. As a result, the Endpoint (ep1) logs into its assigned Gateway (gw2).

Refer to the redbook, *All About Tivoli Management Agents*, SG24-5134, for more information about the TMR redirection.

Chapter 5. AS/400 Endpoint considerations and tips

Normally, once you have installed Tivoli Management Agents on managed systems, most management operations and configurations are performed in the same manner on all supported platforms. However, some operations depend on each platform of the managed system. This chapter introduces AS/400 Endpoint-specific considerations for configurations and operations, and also provides hints and tips for AS/400 Endpoint configuration.

5.1 AS/400 Endpoint considerations

This section introduces some considerations in AS/400 Endpoint management environment.

5.1.1 Using tasks on an AS/400 Endpoint

In patch 3.6.1-TMF-0002, Tivoli provides an AS/400 Task Library to assist in automating frequently-performed tasks. These tasks are also designed to help automate Tivoli Enterprise Console actions. The tasks provided in the AS/400 Task Library are listed in Table 13.

Table 13. AS/400 Endpoint task library

AS/400 task	Description
Command	This command enables you to enter any AS/400 CL command, valid in batch mode, for which all parameters are known. Prompting is not supported.
Send_Reply	This command invokes the AS/400 command, <code>SNDRPLY</code> , to send a reply to a message on a specific message queue.
Run_Backup	This command invokes the AS/400 command, <code>RUNBKUP</code> , to start saving information on your AS/400.
Start_Cleanup	This command invokes the AS/400 command, <code>STRCLNUP</code> , to start cleaning up information on your AS/400, such as spool files or temporary libraries.
Power_Down_System	This command invokes the AS/400 command, <code>PWRDWN SYS</code> , to end all subsystems, power down the system, and, optionally, re-IPL a remote AS/400 system.
Vary_Configuration	This command invokes the AS/400 command, <code>VRRCFG</code> , to reset lines, controllers, and devices attached to an AS/400.

Note that these tasks must be run under valid OS/400 user profiles. Additionally, these user profiles must have the necessary authority needed to

run the OS/400 commands corresponding to the task. By default, the User Name specified in the task definition is “*”. This means that the command will run as the user ID running the desktop. If the user running the desktop is root, this will automatically be mapped to QTIVROOT on the AS/400. If the user does not exist on the AS/400 or if the user does not have the necessary authority, the QLCFD job will be unable to spawn the job to perform the task.

5.1.1.1 AS/400 task scenario

Let us suppose that users from one AS/400 endpoint, KONG, are receiving messages indicating that they are not authorized to run a program called UPDATES. Meanwhile, users on another AS/400 endpoint, SHAFT, are running this same program with no errors. We decide to run the following Display Object Authority (DSPOBJAUT) command on both endpoints:

```
DSPOBJAUT OBJ (LPVLIB/UPDATES) OBJTYPE (*PGM)
```

We will compare the results to see if we can determine the cause of the problem. We begin by opening the AS/400 Task Library.

Because the UPDATES program is part of an application owned by the LORRAINE user profile, we want to run this command as her user profile to assure that we have all the necessary authority; so, we right click on the **Command** task and select **Edit Task**.

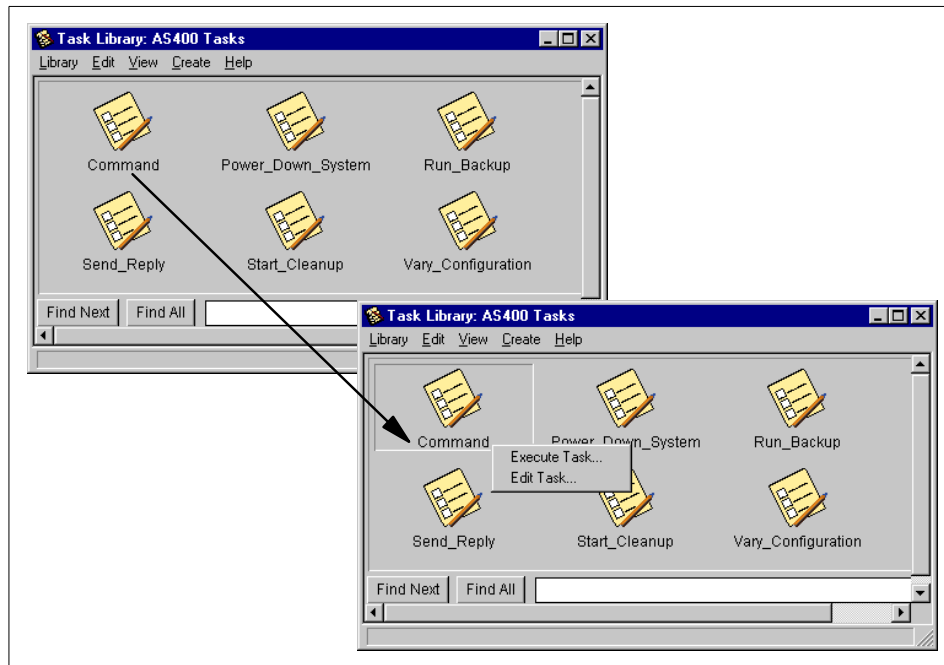


Figure 28. AS/400 task library

This brings up the following Edit Task panel where we specify this user profile as the User Name as shown in Figure 29 on page 102.

Figure 29. Editing AS/400 task

We select **Change & Close**, which returns us to the AS/400 Tasks Library. Now, we double click on **Command Task** to select to execute the task. We fill in the values as shown in Figure 30 on page 103.

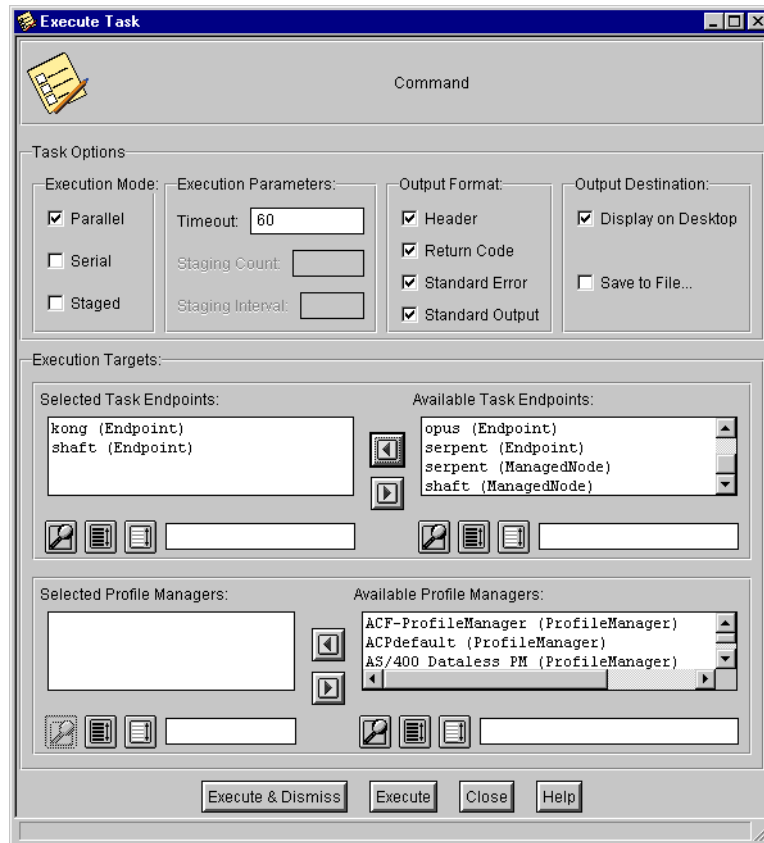


Figure 30. Executing AS/400 task

Next, we press **Execute & Dismiss**, which brings up the Configure Tasks Arguments panel for the Command task. As shown in Figure 31 on page 104, in this panel, we enter the Display Object Authority command as desired.

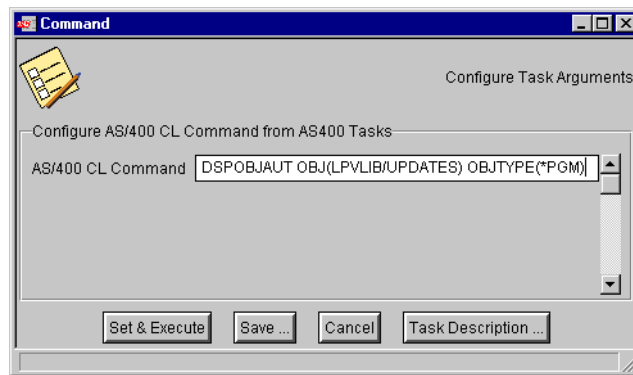


Figure 31. Configuring AS/400 task arguments

We then select **Set & Execute** to run the command on these AS/400 endpoints. The Command Output screen shown in Figure 32 shows our results.

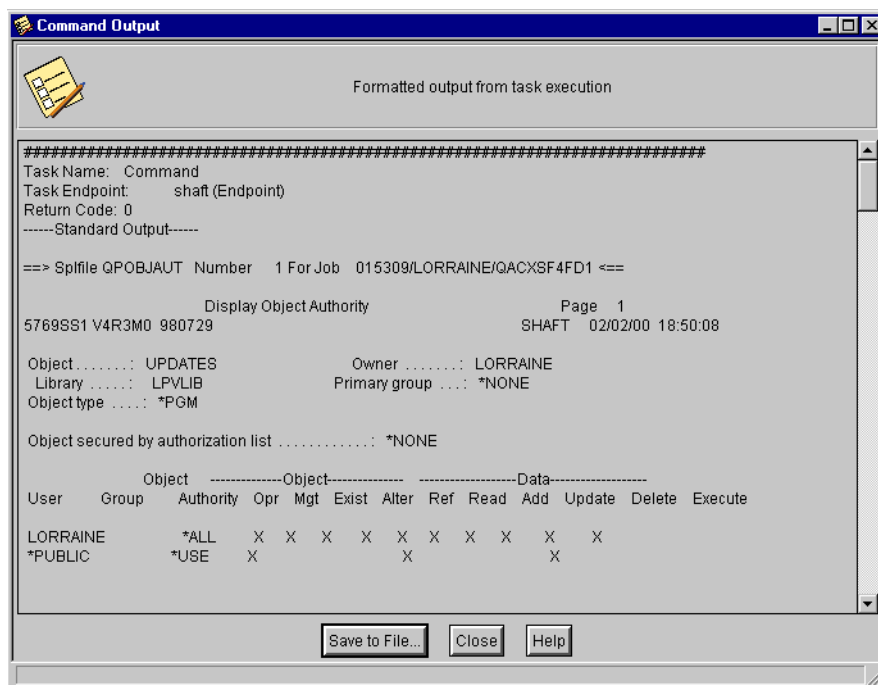


Figure 32. The command task output from AS/400 (shaft)

Notice that from the output provided from SHAFT, we can see that the *PUBLIC authority on the UPDATES program is *USE. We window down to find the following output from this command run on KONG as shown in Figure 33.

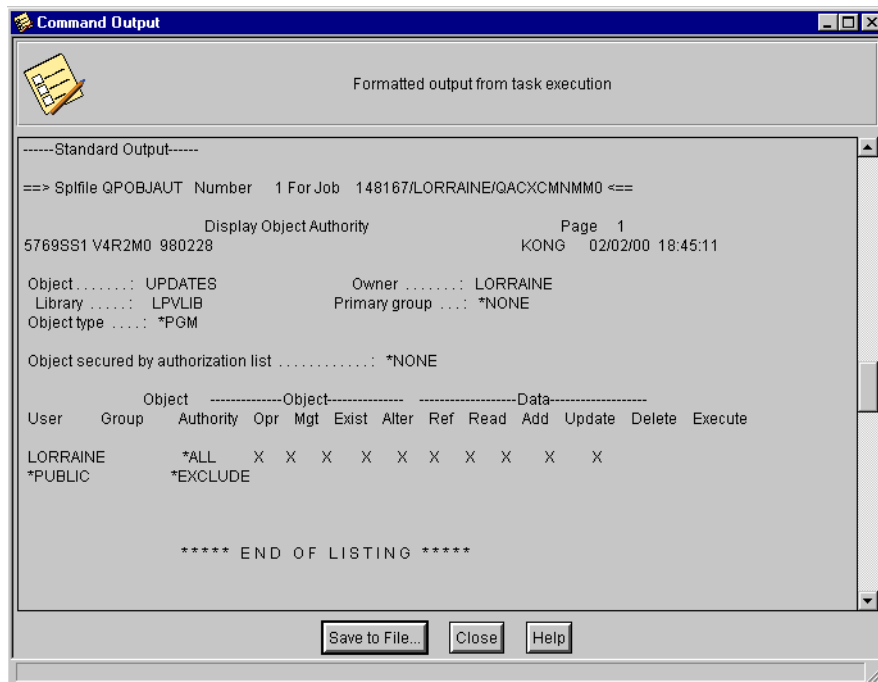


Figure 33. The command task output from AS/400 (kong)

Clearly, on KONG, the *PUBLIC authority for this same program is *EXCLUDE. This is quite possibly the cause of the authority errors being encountered. At this point, using this same Command task, we would issue the Edit Object Authority (`EDTOBJAUT`) command with the appropriate parameters on the KONG endpoint to set the *PUBLIC authority to *USE to resolve the problem.

Note

- If you wanted to run this same command, with the same parameter several times, you could use the Save button on the Configure Task Arguments to save this specification. An example of this is provided in Chapter 9, “AS/400 management scenarios” on page 437. This scenario also illustrates the use of the task as an automated TEC response.
- If, however, you wanted to be able to run the Display Object Authority command using different parameters each time, you could create your own task for this. In Chapter 6, “AS/400 security management” on page 139, for example, we create a sample “AS/400 User Administration” task library, which performs the following OS/400 commands:
 - Add Directory Entry (ADDDIRE)
 - Remove Directory Entry (RMVDIRE)
 - Display User Profile (DSPUSRPRF)
 - Change Object Owner (CHGOBJOWN)

The sample file for this task library is provided in Appendix B, “AS/400 User Administration task library definition file” on page 511.

- Additionally, in Chapter 7, “AS/400 availability management” on page 217, we will illustrate adapting the Send_ Reply task to run as a TEC task.

For complete details of OS/400 commands, refer to the *AS/400 Prog: Control Language Ref V2R3MO*.

5.1.2 Working with Tivoli Files on the AS/400

The majority of Tivoli files on the AS/400 are stored in the IFS file system in the /QIBM/UserData/Tivoli/LCF directory. As such, these files are not viewable or editable using native AS/400 commands. Nonetheless, there are commands provided with which you can copy IFS files to physical files and vice versa. In this section, we will show a few simple examples of doing this. Additionally, since joblogs on the AS/400 are spooled files, it is often helpful to copy these to physical files so that they can be FTP to support. We will illustrate this and provide some concluding remarks on FTP and the AS/400.

Note

OS/400 V4R2 and higher include Qshell, which enables you to view IFS files. The Client Access/400 might be the easiest way to display files in the /QIBM/UserData/Tivoli/LCF directory if you have installed Client Access/400 in your environment.

5.1.2.1 Copying IFS files to physical files

In certain cases, you may be able to specify that Tivoli write to native AS/400 files rather than to an IFS file. Of course, if this can be done, it removes the need to copy the IFS file to a physical file. Nonetheless, we wish to illustrate this here for your use as well.

By default, the last.cfg file is an IFS file. However, the Start TMA Endpoint (`STRIMEEPT`) command allows you to overwrite the default value and, if desired, specify a native physical file. For example, suppose that we wish to write the configuration file into a physical file called LASTCFG in the LPVLIB library. The LPVLIB library already exists on the AS/400, but there is no file by this name. We will specify to write to a member named LASTCFG in the LASTCFG file. This can be achieved by performing the following steps:

1. Since the QLCFD job will need to access this file, we use the Grant Object Authority commands, as shown in Figure 34 on page 108, to grant QTIVOLI authority to the LPVLIB library in which we will specify to write the configuration file (last.cfg).

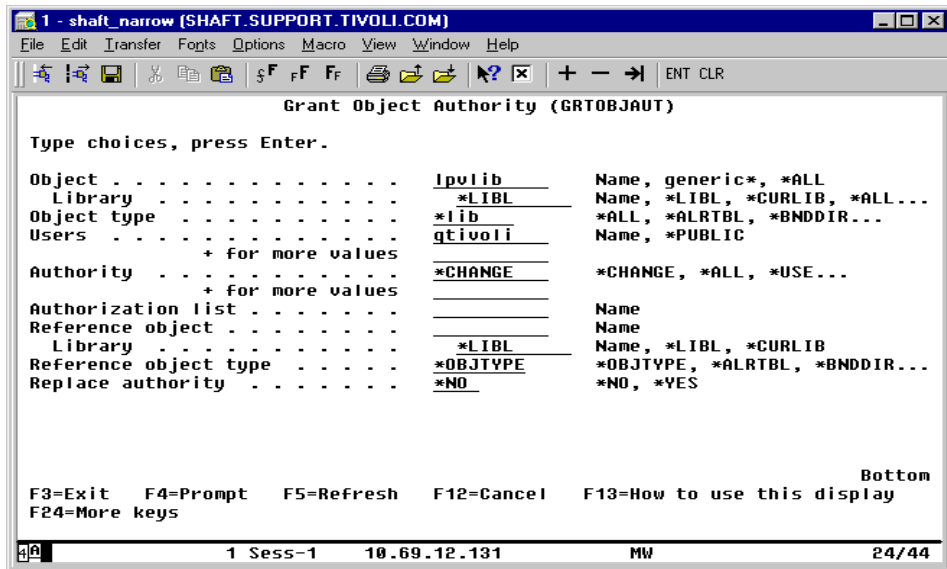


Figure 34. Grant Object Authority

- Now, when we issue the Start TMA Endpoint command, we specify the Configuration File parameter using IFS notation as shown in Figure 35.

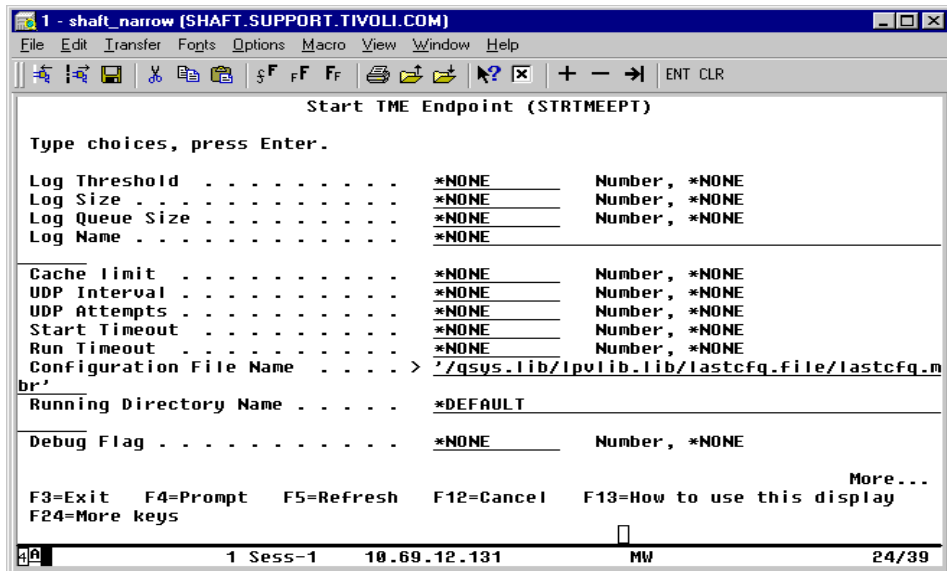


Figure 35. Start TMA Endpoint (STRTEEPT)

3. To verify that a file has been created and written to as expected, we issue the following Display Physical File Member command:

```
DSPPPFM FILE(LPVLIB/LASTCFG) MBR(LASTCFG)
```

The result is shown in Figure 36.

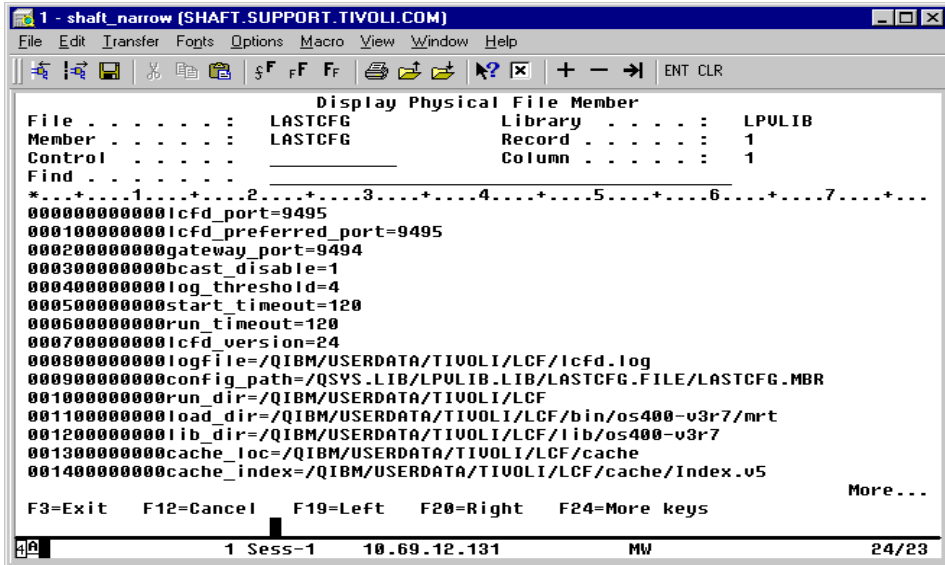


Figure 36. Display Physical File Member (LASTCFG)

As in the previous case, there was no need to actually copy an IFS file to a physical file. We now show an example where this is necessary. Let us say that we wish to view the `lcfld.log` file on the AS/400 itself. Since this file must be an IFS file, we will copy it to a physical file with the following steps:

1. Use the Create Physical File command to create a file called `LCFDLOG` in the `LPVLIB`. The command specification we used is shown in Figure 37 on page 110.

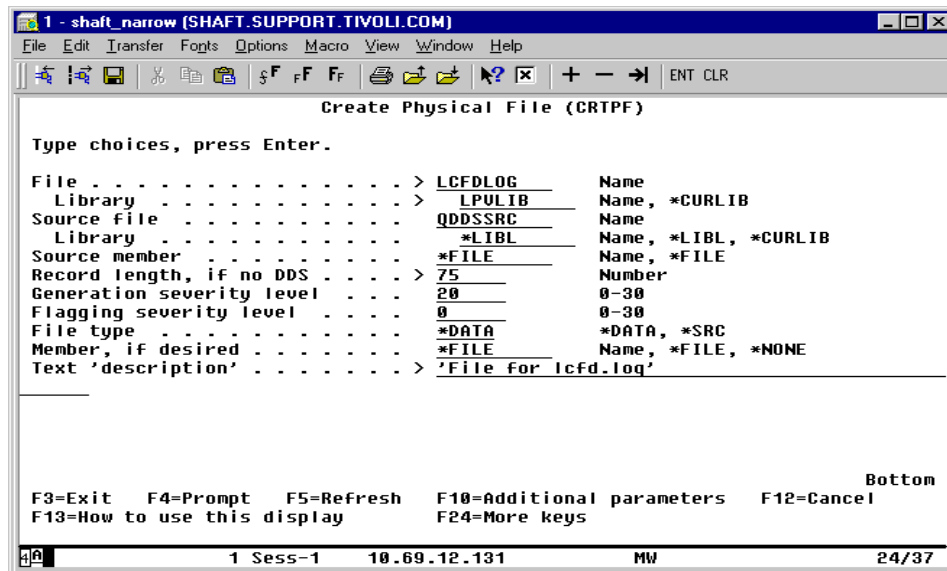


Figure 37. Create Physical File (LCFDLOG)

We select a Record Length of 75 because we wish to keep the file viewable on one screen without windowing. But, this value can be altered as desired. Also, note that, by default, this command will create a member in the file with the same name as the file name and set the Maximum members value to 1. If you wish to add additional members, you may wish to alter this Maximum members value.

2. You will not be able to copy this file while the QLCFD job is active; so, if necessary, end this job by issuing the End TMA Endpoint (ENDTIMEPT) command.
3. Issue the Copy From Stream File (CPYFRMSTRMF) command as shown in Figure 38 on page 111.

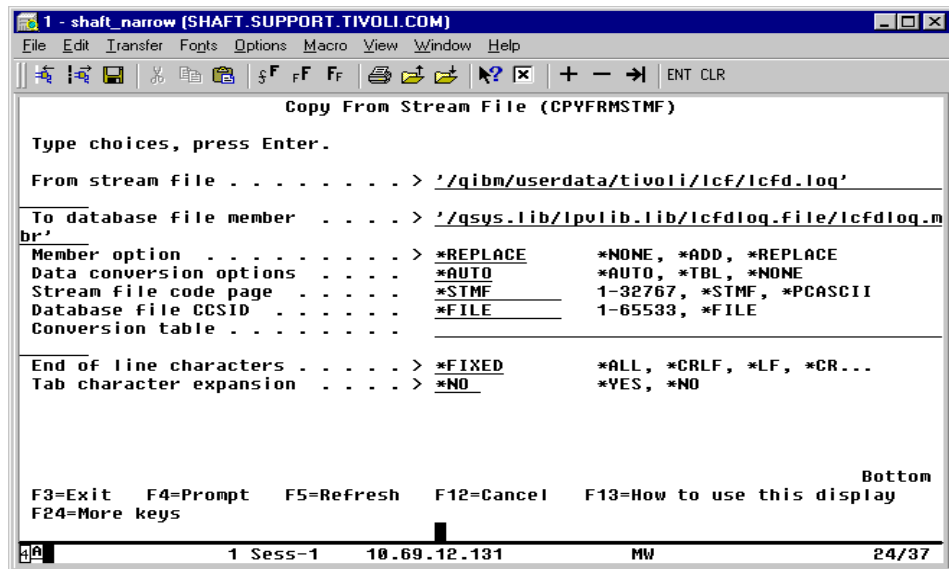


Figure 38. Copy From Stream File (lcfd.log to LCFDLOG)

Notice that specifying *FIXED as the End of line characters value will cause the data to wrap at the record length of our physical file. Otherwise, the data will be truncated if it extends passed this value.

4. To view the file, use the following Display Physical File Member command.

DSPPFM FILE (LPVLIB/LCFDLOG) MBR (LCFDLOG)

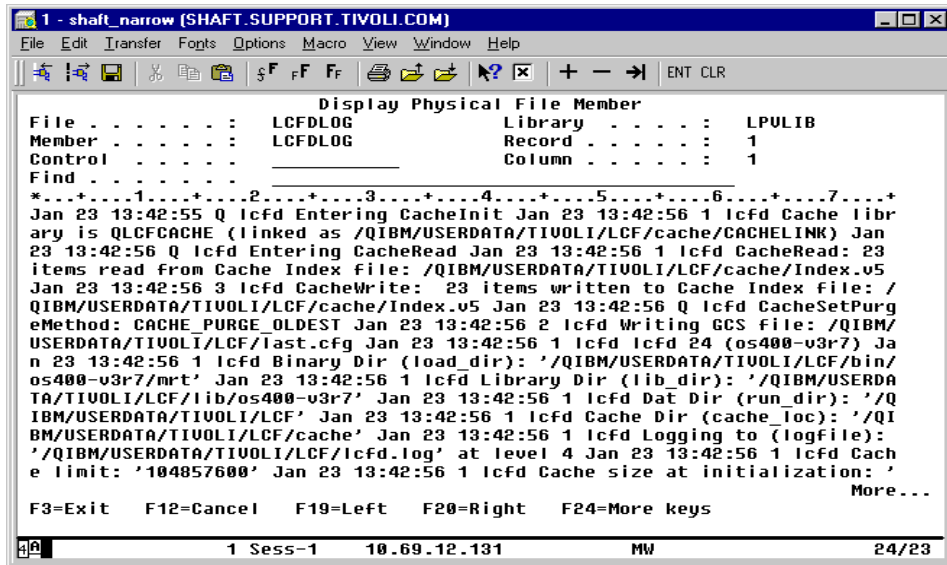


Figure 39. Display Physical File Member (LCFDLOG)

5.1.2.2 Copying Physical Files to IFS Stream Files

Now, let us suppose that after copying the lcfid.log file, we restarted the QLCFD job. Now, the lcfid.log file has been overwritten with new information. After reviewing our LCFDLOG physical file, we decide that we like to keep it for awhile. For the sake of discussion, let's say that we decide now that we wish to keep this data in an IFS file in the /tmp directory. We can use the following Copy to Stream File command to do this.

```
CPYTOSTMF FROMMBR('qsys.lib/lpvlib.lib/lcfidlog.file/lcfidlog.mbr')
TOSTMF('tmp/lcfid_02122000.log')
```

Note

We have intentionally left out any example in which a Tivoli file is copied from an IFS file, edited using native OS/400 commands, and then copied back. We do this to emphasize that this is not recommended. Using supported Tivoli interfaces to access these files is the best way to edit them. If, for some reason, you decide to edit a file using the techniques shown here to copy and replace files, use extreme care that the data is not truncated or damaged in the process.

5.1.2.3 Copying spooled files to physical files or IFS stream files

Along with copying between physical files and stream files, you may have occasion to wish to copy a spooled file to a physical file. Let us suppose, for example, that we have a joblog of a DMEPEENGINE job, which has failed. Tivoli support has asked us to put this on their FTP site for them to review. However, the joblog is actually a spooled file (formatted for a printer) on the AS/400. This is not an object that can be FTPd; so, we need to perform the following steps to copy this spooled file into a physical file.

1. Create a physical file into which to copy the spoolfile. This can be done using the Create Physical File (CRTPF) command; so, we issue the command shown here to create a physical file named JOBLOG in the LPVLIB library. Note that the LPVLIB library already exists on the system. By default, the file will be created with one member with the same name as the file.

```
CRTPF FILE(LPVLIB/JOBLOG) RCDLEN(132) MBR
```

Note

We specified a record length of 132 because this is the width of the QPJOBLOG spoolfile. You may need want to adjust this value higher if you are copying a wider spoolfile.

The LPVLIB library already exists on our system. You can use any library you desire on your system in the above command. If you want to create a new library, this can be done easily by issuing the Create Library (CRTLIB) command as follows:

```
CRTLIB LIB(libraryname) ***
```

2. As you will see in step 3, in order to use the Copy Spooled File command, we need the following information.
 - Spooled file name
 - Qualified job name that produced the spooled file including:
 - Job name
 - User
 - Number
 - Spooled file number

One method of obtaining this information is by using option 8=Attributes beside the spooled file from the Work with Spooled Files screen. To get to the

Work with the Spooled Files panel showing only QTIVROOT's spooled files, we issue the following command:

```
WRKSPLF SELECT(QTIVROOT)
```

Now, beside the joblog we wish to copy, we select option 8 as shown in Figure 40.

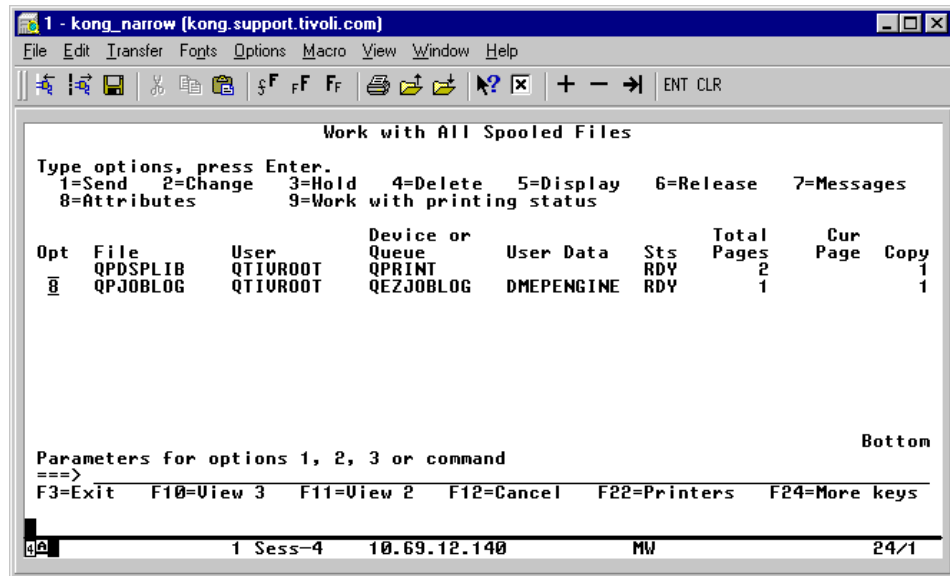


Figure 40. Work with All Spooled Files: Option 8=attributes

From the Work with Spooled File Attributes panel, locate and record the necessary values as shown in Figure 41 on page 115.

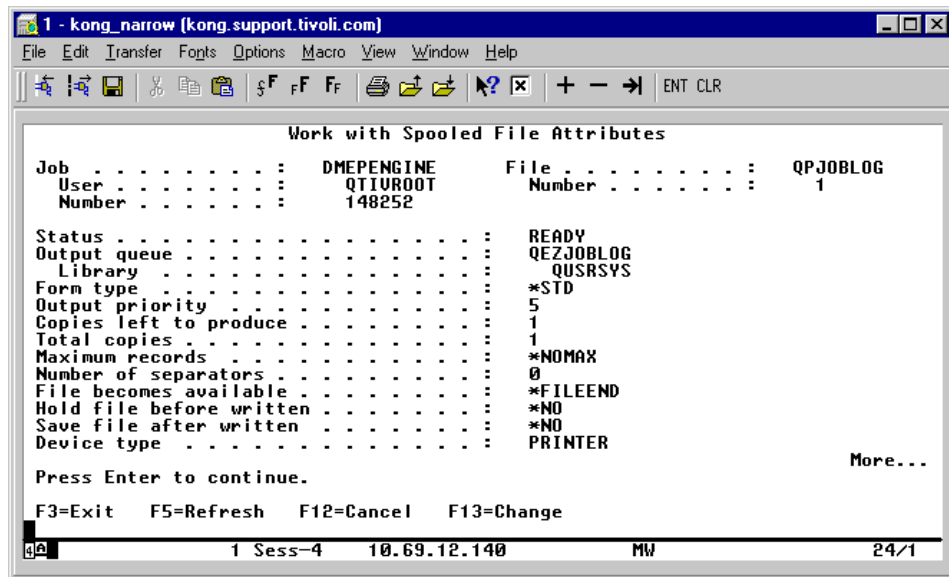


Figure 41. Work with Spooled File Attributes

The necessary values are:

- Spooled file name: QPJOBLOG
- Qualified job name which produced the spooled file including:

Job name	DMEPENGINE
User	QTIVROOT
Number	148252
- Spooled file number: 1

3. Next, we issue the Copy Spooled File (CPYSPLF) command to copy the joblog in the physical file. We issue the command as follows:

```
CPYSPLF FILE(QPJOBLOG) TOFILE(LPVLIB/JOBLOG)
JOB(148252/QTIVROOT/DMEPENGINE) SPLNBR(1) TOMBR(JOBLOG)
```

Note

When issuing a command with numerous parameters, such as this, many people prefer to type only the command name (`CPYSPLF`) and then press the F4=Prompt key to see a listing of all possible parameters. Pressing F11=Keywords will then show you the exact parameter keywords; so, you can easily identify where to supply the values.

The Display Physical File Member (`DSPPFM`) command can now be used to view the physical file member and verify that the member contains the data as expected.

5.1.2.4 Using FTP with an AS/400

As a final topic regarding the handling of Tivoli files on the AS/400, we wish to briefly discuss the topic of using FTP to move files between an AS/400 and other systems. In our specific examples, we will use an AS/400 and a UNIX machine.

FTP on the AS/400 functions in the same way as it does on other platforms. That said, the existence of two naming conventions, the library file system format and the IFS directory format, often creates confusion when using FTP. Consider, for example, the following two Tivoli files.

- The Endpoint logfile called `lcfd.log`
Type: Stream File
Location: `/QIBM/UserData/Tivoli/LCF/lcfd.log`
- The TEC AS/400 message adapter configuration file
Type: Physical File
Location: `QUSRSYS/QAYMACFG`

Notice the different formats used above for these file locations. The first location is a directory path type of format while the second is the standard AS/400 library file format.

To allow users the flexibility to use either of these formats, FTP on the AS/400 provides a naming indicator called `NAMEFMT`:

- `NAMEFMT 0` is the original library file system format with which many AS/400 users are accustomed. A fully-qualified file name in this format is:
`library/filename.membername`
- `NAMEFMT 1` indicates an IFS pathname of the form:

/filesystem/directory_path/filename

Notice that it is possible to specify physical files using NAMEFMT 1 as follows:

/QSYS.LIB/library.LIB/filename.FILE/membername.MBR

With this introductory information complete, we will now illustrate several examples of using the NAMEFMT indicator in some actual FTP sessions.

For our first example, suppose we want to FTP from an AS/400 to a UNIX machine and then use the `put` command to move the `lcfcd.log` and the `QAYMACFG` files. Examine the following FTP session shown in Figure 42.

```
3 - kong_narrow (kong.support.tivoli.com)
File Edit Transfer Fonts Options Macro View Window Help

File Transfer Protocol

Previous FTP subcommands and messages:
230 User root logged in.
UNIX Type: L8
> put /QIBM/UserData/Tivoli/lcf/lcfcd.log
File name format not valid. Use the format: library/file.member or
file.member.
> namefmt 1
500 'SITE NAMEFMT 1': command not understood.
Client NAMEFMT is 1.
> put /QIBM/UserData/Tivoli/lcf/lcfcd.log
227 Entering Passive Mode (10,69,11,110,6,248)
150 Opening ASCII mode data connection for lcfcd.log.
226 Transfer complete.
38857 bytes transferred in 0.060 seconds. Transfer rate 652.288 KB/sec.
Enter an FTP subcommand.
===>

F3=Exit      F6=Print      F9=Retrieve
F17=Top      F18=Bottom    F21=CL command line

3 Sess-4    10.69.12.140    24/1
```

Figure 42. FTP from AS/400 to UNIX (1 of 2)

Notice that, after logging into the UNIX system, our initial attempt at using the `put` command to move the `lcfcd.log` file failed. However, after issuing `NAMEFMT 1`, the same “`put`” command succeeds.

Now, consider the following actions, shown in Figure 43 on page 118, performed immediately following the above in the same FTP session.

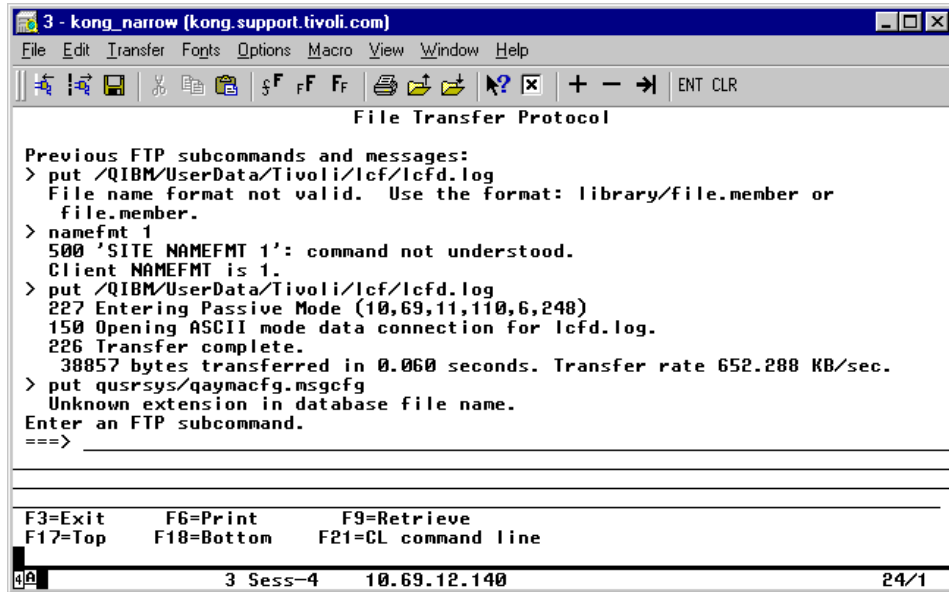


Figure 43. FTP from AS/400 to UNIX (2 of 2)

Because we specified NAMEFMT 1, now our use of the library file format with the `put` command is not correct. As shown in Figure 43, after we issue NAMEFMT 0, the same `put` command is now successful.

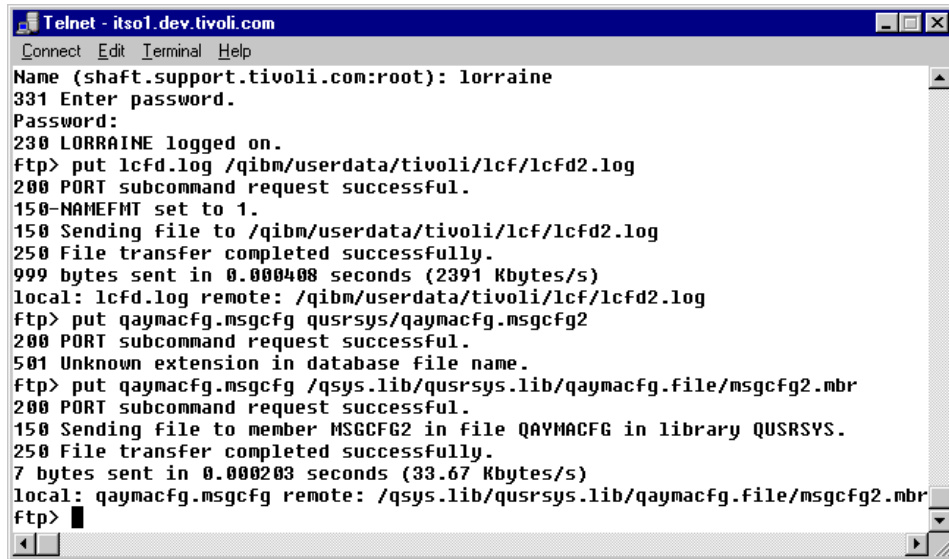
Note that, rather than issue NAMEFMT 0, we could have successfully FTPd the file using the following:

```
put /qsys.lib/qusrsys.lib/qaymacfg.file/msgcfcg.mbr
```

After the successful move, the following files are on the UNIX machine:

- `lcfd.log`
- `qaymacfg.msgcfcg`

We now show an example of an FTP session from the UNIX machine to the AS/400. Again, we will use the `put` command. In this case, we will move the `lcfd.log` and `qaymacfg.msgcfcg` files back to the AS/400. However, since we really do not wish to replace the existing files, we modify the filenames slightly. Review the following FTP session shown in Figure 44 on page 119.



```
Telnet - itso1.dev.tivoli.com
Connect Edit Terminal Help
Name (shaft.support.tivoli.com:root): lorraine
331 Enter password.
Password:
230 LORRAINE logged on.
ftp> put lcfd.log /qibm/userdata/tivoli/lcf/lcfd2.log
200 PORT subcommand request successful.
150 NAMEFMT set to 1.
150 Sending file to /qibm/userdata/tivoli/lcf/lcfd2.log
250 File transfer completed successfully.
999 bytes sent in 0.000408 seconds (2391 Kbytes/s)
local: lcfd.log remote: /qibm/userdata/tivoli/lcf/lcfd2.log
ftp> put qaymacfg.msgcfg qusrsys/qaymacfg.msgcfg2
200 PORT subcommand request successful.
501 Unknown extension in database file name.
ftp> put qaymacfg.msgcfg /qsys.lib/qusrsys.lib/qaymacfg.file/msgcfg2.mbr
200 PORT subcommand request successful.
150 Sending file to member MSGCFG2 in file QAYMACFG in library QUSRSYS.
250 File transfer completed successfully.
7 bytes sent in 0.000203 seconds (33.67 Kbytes/s)
local: qaymacfg.msgcfg remote: /qsys.lib/qusrsys.lib/qaymacfg.file/msgcfg2.mbr
ftp>
```

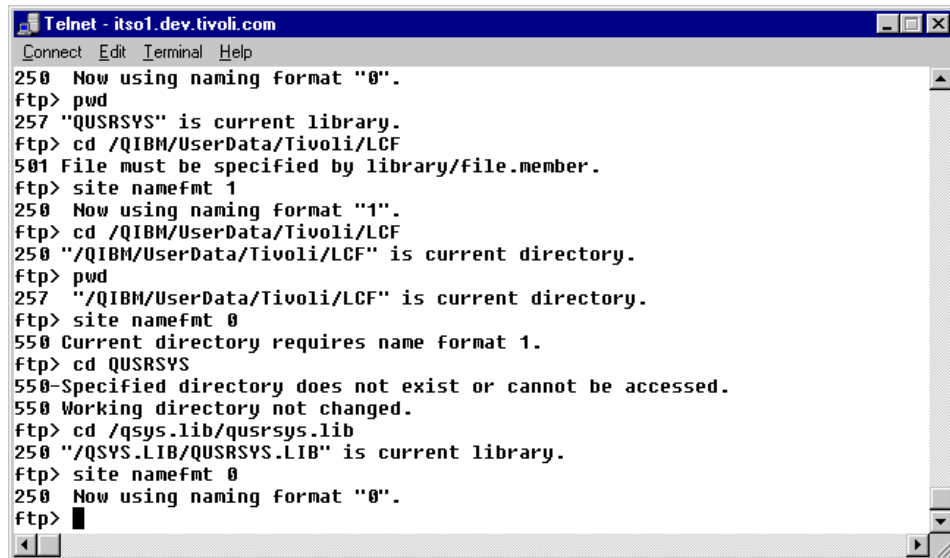
Figure 44. FTP from UNIX to AS/400 (1 of 2)

Notice that the initial `put` command in the IFS directory automatically set the NAMEFMT 1 for us. Following this, however, our attempt to put the `qaymacfg.msgcfg` file using the library file naming convention fails. In this case, we illustrate using the NAMEFMT 1 structure to successfully put the file in the QAYMACFG physical file. Of course, if we really insisted on using the library file name format to accomplish this, we could have issued the following commands:

```
site namefmt 0
put qaymacfg.msgcfg qusrsys/qaymacfg.msgcfg
```

Note the use of the SITE indicator here to perform the NAMEFMT 0 on the AS/400.

Finally, examine the screen, shown in Figure 45 on page 120, which is a continuation of this same FTP session.



```
Telnet - itso1.dev.tivoli.com
Connect Edit Terminal Help
250 Now using naming format "0".
ftp> pwd
257 "QUSRSYS" is current library.
ftp> cd /QIBM/UserData/Tivoli/LCF
501 File must be specified by library/file.member.
ftp> site namefmt 1
250 Now using naming format "1".
ftp> cd /QIBM/UserData/Tivoli/LCF
250 "/QIBM/UserData/Tivoli/LCF" is current directory.
ftp> pwd
257 "/QIBM/UserData/Tivoli/LCF" is current directory.
ftp> site namefmt 0
550 Current directory requires name format 1.
ftp> cd QUSRSYS
550-Specified directory does not exist or cannot be accessed.
550 Working directory not changed.
ftp> cd /qsys.lib/qusrsys.lib
250 "/QSYS.LIB/QUSRSYS.LIB" is current library.
ftp> site namefmt 0
250 Now using naming format "0".
ftp>
```

Figure 45. FTP from UNIX to AS/400 (2 of 2)

Notice that, when we are in NAMEFMT 0, it is not possible to `cd` in the IFS directory file system because this requires the NAMEFMT 1 format. For this same reason, once in the directory file system, issuing NAMEFMT 0 is not allowed; so, if you wish to switch back to NAMEFMT 0, you must first `cd` into the library file system. Notice, as illustrated, that in order to do so requires you to perform the `cd` command using the current NAMEFMT 1 naming structure.

5.1.3 Dataless profile manager and database profile manager

The dataless profile manager is a new profile manager type associated with the dataless client (Endpoint). The dataless profile manager can have the following subscribers:

- Endpoints
- Managed Nodes
- PC Managed Nodes
- NIS Domains
- NetWare Managed Sites

As you can see in the list above, other profile managers cannot be subscribers to the dataless profile manager. Therefore, the dataless profile manager cannot be a branch node in a profile manager hierarchy. A dataless

profile manager can only be a leaf node, that is, it can only have managed systems as subscribers.

5.1.3.1 What is the difference between dataless and database?

The Endpoint (Tivoli Management Agent) does not have an object database (.bdb file) locally; the information in the profile created in the dataless profile manager is not written to the database associated with the next level, even if it is a Managed Node (refer to Figure 46). Therefore, if you distribute the profile using the dataless profile manager to full Managed Nodes or Endpoints, the data is applied directly to the system.

The profile created in the database profile manager can be locally modified because the information in the profile is written to the local database of the full Managed Node. However, the local modification is not a recommended customization because it makes profile management more complicated, and it also makes future migration from a full Managed Node to the Endpoint (TMA) more difficult.

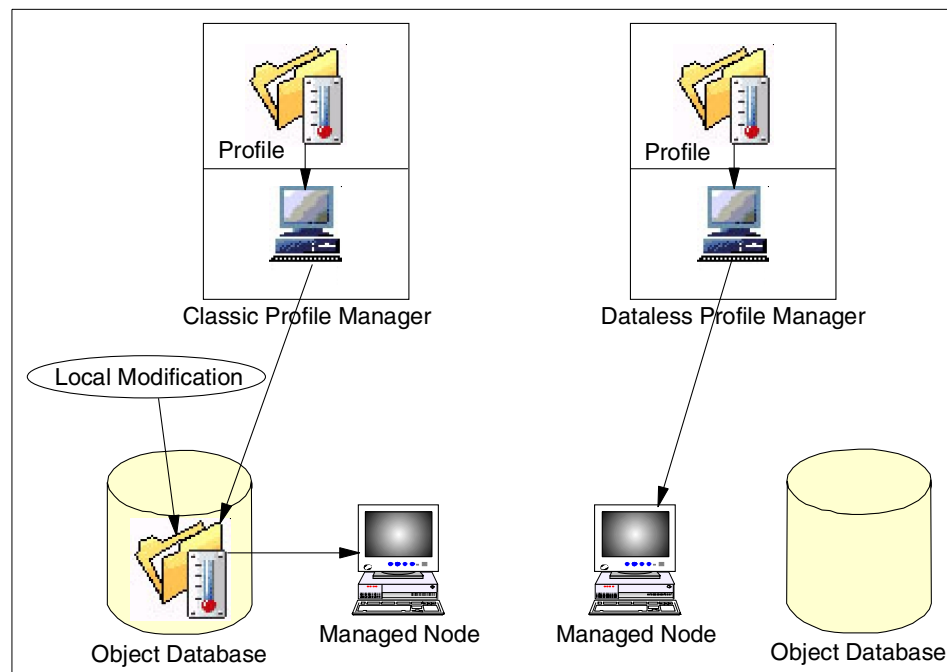


Figure 46. Difference between dataless and database profile managers

5.1.3.2 AS/400 Endpoint and dataless profile manager

The dataless profile manager was developed to support Endpoint subscribers. However, almost all features, such as defining the profile, handling the profile, and so on (except writing information to the database), are the same as the database profile manager.

At present, AS/400 supports only Endpoint (TMA). When you manage AS/400 Endpoints with Tivoli Enterprise, you must use the dataless profile manager to perform management operations. In any case, we strongly recommend that you use the dataless profile manager as follows:

- You should not create any profiles in the dataless profile manager.
- The dataless profile managers should be used as subscribers to classic profile managers for distributing profiles to the AS/400 Endpoints.

This is the best way to avoid trouble regarding the dataless profile manager and the AS/400 Endpoints.

5.1.4 AS/400 Endpoint method management

At present, AS/400 supports only Endpoint. To support Endpoint on AS/400 and run Tivoli Enterprise applications on AS/400 Endpoints, Tivoli for AS/400 Endpoints provides Endpoint methods as well as other Tivoli management applications. These Endpoint methods that are used for AS/400 Endpoints are managed by the dependency manager. The following sections introduce how Endpoint methods work on AS/400 Endpoints in your environment.

5.1.4.1 AS/400 Endpoint method cache management

Endpoint method cache management is very important in understanding Endpoint implementation on AS/400 systems. The Endpoint gives you the Tivoli Management Framework functions that are necessary to perform management operations, such as those performed by Tivoli core applications, and these management operations are processed by calling a method on a managed resource. In the Endpoint environment, this design has not changed; the Endpoint invokes an Endpoint method for performing management operations on the Endpoint platform.

The Endpoint methods that will be used by the AS/400 Endpoint are stored in the Endpoint Gateway. When the AS/400 Endpoint performs an operation, the AS/400 Endpoint automatically determines what is needed to perform the given management operation. If that Endpoint method already resides on the AS/400 Endpoint, it immediately proceeds with the operation. If it does not, the AS/400 Endpoint downloads the appropriate Endpoint method from the Endpoint Gateway to the AS/400 Endpoint with no operator intervention. In

addition, the Endpoint downloads newer versions as updates are loaded on the Endpoint Gateway. You can gain significant productivity advances with these management features because Tivoli applications are installed only once on the Endpoint Manager (TMR server) and Endpoint Gateways, with updates performed automatically thereafter. Figure 47 shows how the Endpoint Gateway and AS/400 Endpoint handle Endpoint methods.

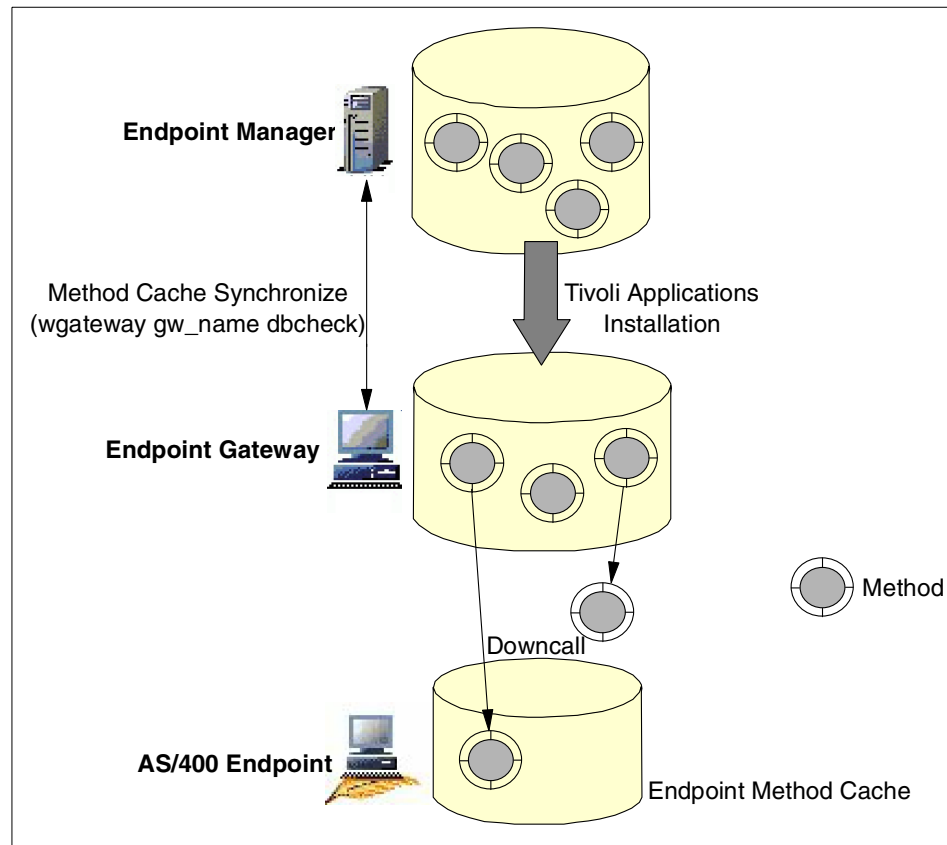


Figure 47. Endpoint method cache management

By default, the AS/400 Endpoint method cache exists under the /QIBM/UserData/Tivoli/lcf/cache directory. The Endpoint Gateway stores the Endpoint methods under the /usr/local/Tivoli/bin/lcf_bundle directory and also stores Endpoint method information in the Endpoint Gateway database (\$DBDIR/gwdb.bdb file). This information can be synchronized with the information on the Endpoint Manager (\$DBDIR/imdb.bdb file) by using the following command:

```
wgateway gw_label dbcheck
```

The following summary describes how the AS/400 Endpoint manages the Endpoint method cache.

- Once the Endpoint method is stored in the Endpoint method cache, the AS/400 Endpoint uses it even if the AS/400 system is rebooted.
- Once the Endpoint method is downloaded to the Endpoint method cache, the Endpoint does not download the same version of the Endpoint method.
- If the AS/400 Endpoint detects that the available version is greater than the current version of the Endpoint method, the AS/400 Endpoint downloads the newer method automatically.
- When the AS/400 Endpoint downloads the Endpoint method, the related methods that are defined in the dependency set are also downloaded at the same time. The dependency set is a list of other files, modules, or commands that are required for the correct operation of the method.

Note

This Endpoint method management is used for all Endpoints that run on other platforms, such as Windows NT or UNIX. This makes application development productive and keeps consistency across all supported platforms.

As we mentioned, Endpoint methods are automatically upgraded if you upgrade the version of Tivoli management applications on the Endpoint Gateway. However, the AS/400 Endpoint daemon is not upgraded automatically. We will introduce how to upgrade the AS/400 Endpoint daemon and when you need to upgrade. Please refer to Section 5.2.3, “Upgrading AS/400 Endpoint daemon” on page 133 for more detailed

5.2 Hints and tips in AS/400 Endpoint configuration

This section provides useful information for configuring AS/400 Endpoint in your environment. It enables you to efficiently manage your AS/400 systems.

5.2.1 Starting the AS/400 Endpoint daemon after IPL

In this section, we introduce two ways of automatically starting the AS/400 Endpoint daemon after IPL.

5.2.1.1 Adding the STRTIMEEPT command to the startup Job

To automatically start the AS/400 Endpoint daemon after IPL, you must modify the program QSYS/QSTRUP. To enable auto start, perform the following procedure:

1. Retrieve the CL Source into a member of a source file by running the following command:

```
RTVCLSRC PGM(QSYS/QSTRUP) SRCFILE(QGPL/QCLSRC)
```

2. Start the Source Entry Utility using the following command:

```
STRSEU SRCFILE(QGPL/QCLSRC) SRCMBR(QSTRUP) TYPE(CLP)
```

3. Add the following lines as the last job to start:

```
QSYS/STRIMEEPT  
MONMSG MSGID(CPF0000)
```

4. Optionally, save the original program.

5. Save the new program by compiling it using the following commands:

```
CRTCLPGM PGM(QSYS/QSTRUP) SRCFILE(QGPL/QCLSRC)  
CHGOBJOWN OBJ(QSYS/QSTRUP) OBJTYPE(*PGM) NEWOWN(QSYS)
```

5.2.1.2 Start the AS/400 Endpoint as an AutoStart job

To start the Endpoint as an AutoStart job, perform the following steps:

1. Create a Job Description to run the STRIMEEPT command:

```
CRTJOB JOB(library/jobname) USER(userprofile)  
RQSDTA('STRIMEEPT .....')
```

where:

- library is an existing library on the AS/400.
- jobname is any name desired.
- userprofile is any userprofile authorized to both the STRIMEEPT command and the QLCFSTAR command processing program as documented in the *Tivoli Framework AS/400 3.6.2 Endpoints User's Guide*, GC32-0279.

2. Add the job description created in Step 1 as an AutoStart job in the desired subsystem. QSYSWRK would be a good subsystem to choose since it is automatically started at each IPL.

```
ADDAJE SBS(QSYS/QSYSWRK) JOB(jobname) JOB(library/jobname)
```

where:

- jobname is any name desired.
- library/jobname matches what was specified above in Step 1.

Note

Because TCP/IP needs to be running prior to the issuance of the `STRTIMEPT` command, the user must ensure that TCP/IP is also set to start automatically before the Endpoint AutoStart job is started.

Most customers will prefer to start the Endpoint as an AutoStart Job using this instruction.

5.2.2 Using the AS/400 Endpoint Web interface

The AS/400 Endpoint daemon includes an integrated HTTP daemon that allows administrators to perform management operations through a Web browser interface, such as Netscape Navigator or Microsoft Internet Explorer. In this section, we introduce the AS/400 Endpoint's Web server function.

5.2.2.1 Accessing the AS/400 Endpoint Web interface

To access the HTTP daemon included in the AS/400 Endpoint, simply use the URL that consists of the hostname (or IP address) and the port number that the AS/400 Endpoint uses for communication with the Endpoint Gateway. Normally, the default port number is 9495. Therefore, to access the AS/400 Endpoint (known as shaft) we use our Web browser to access the following URL:

`http://shaft.support.tivoli.com:9495`

Figure 48 on page 127 shows the first page of the AS/400 Endpoint Web interface.

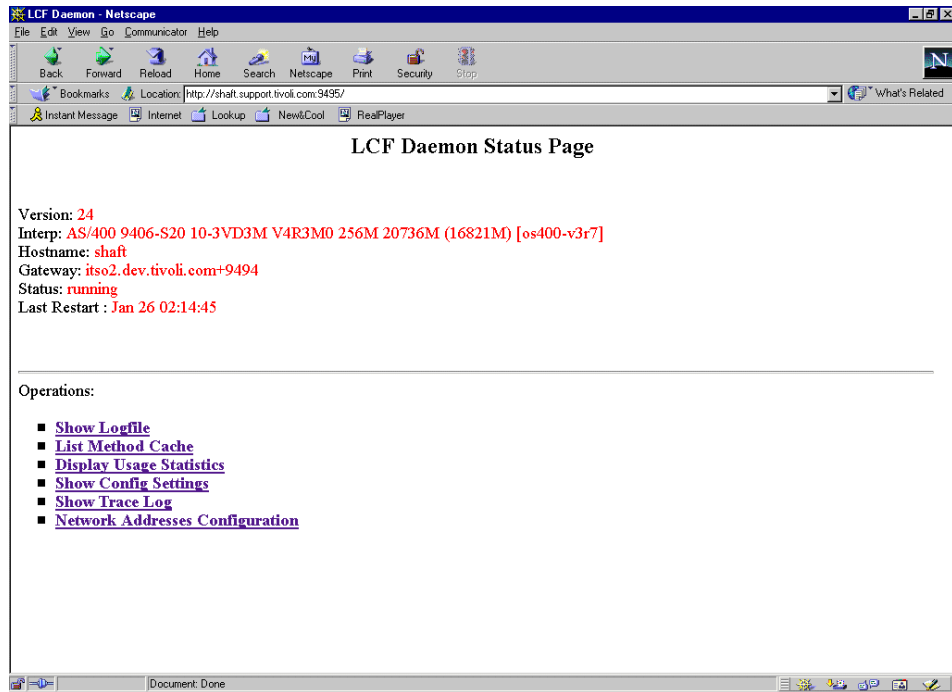


Figure 48. AS/400 Endpoint Web interface

5.2.2.2 AS/400 Endpoint Web interface functions

The AS/400 Endpoint Web interface provides very specific and useful information and services to administrators. The administrator can use a Web browser to query information about the AS/400 Endpoint, as well as use it to change the AS/400 Endpoint's configuration options. There are seven specific pages that display information and allow an administrator to alter the AS/400 Endpoint's configuration options. The following describes the information and operations provided by each page.

LCF Daemon Status page:

This is the primary page on the AS/400 Endpoint Web interface. The top half of this page provides information regarding the Endpoint including:

Version	The version of the AS/400 Endpoint daemon
Interp	The operating system on which the AS/400 Endpoint resides
Hostname	The AS/400 Endpoint's hostname
Gateway	The Endpoint Gateway's address and port number
Status	The current status of the AS/400 Endpoint

Last Restart The date and time that this Endpoint was last started

Show Logfile page

This page simply displays the contents of the AS/400 Endpoint's lcf.d.log file. Since AS/400 does not provide a standard editor, such as vi editor for UNIX, this function is quite useful for problem determination of AS/400 Endpoint.

List Method Cache page

It is sometimes useful to be able to see a listing of the Endpoint methods that are currently available in the AS/400 Endpoint method cache. The methods that displayed on this page would be stored in the cache directories on the AS/400 Endpoint.

Display Usage Statistics page

This page, as its name implies, provides statistics related to the number of downcalls that have been issued and the hit and miss rate for the AS/400 Endpoint method cache.

Show Config Settings page

This page simply displays the current configuration setting for the AS/400 Endpoint. The information presented is made up of information contained in the last.cfg and lcf.dat files. More information on these files is presented in Chapter 4, "AS/400 Endpoint configuration" on page 61.

Show Trace Log page

This page allows the administrator to view the trace log of messages sent and received at the AS/400 Endpoint.

Network Address Configuration page

The Network Address Configuration page displays information about the current AS/400 Endpoint settings, but, more importantly, it allows the administrator to change the configuration of the AS/400 Endpoint (see Figure 49 on page 129). In the Additional configuration options field, you can specify any parameters supported by the Endpoint daemon. Refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more detailed information about the Endpoint options.

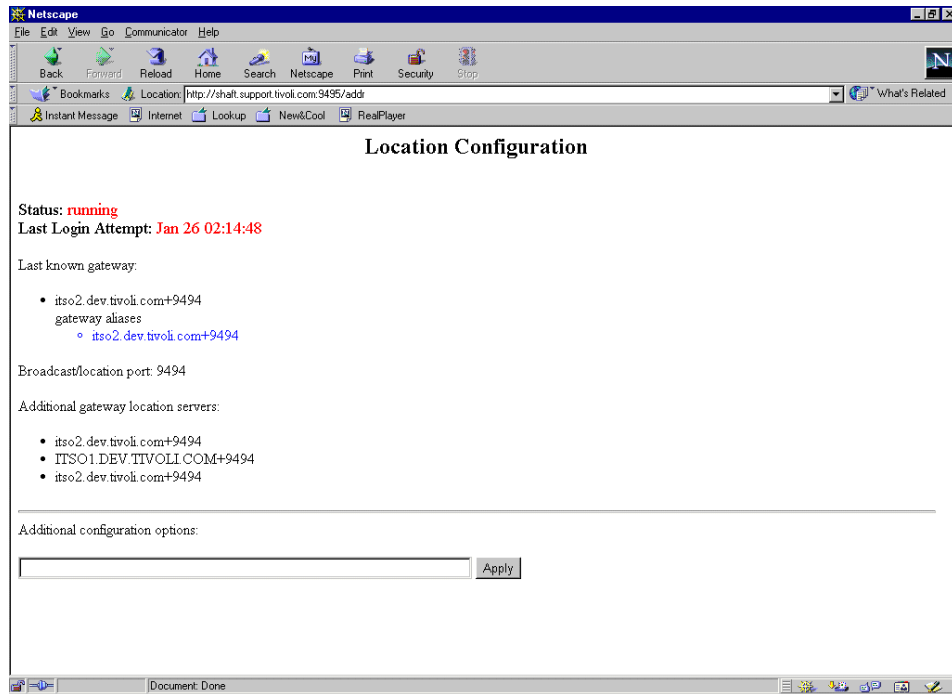


Figure 49. Location Configuration

For instance, you can use this page to alter the default Endpoint Gateway for the AS/400 Endpoint. Once you have entered the configuration options and selected **Apply** button, you will be prompted for a user ID and password. To obtain the proper http password, the TMR administrator (who needs senior or super authority) can issue the `wep` command as follows:

```
# wep shaft get httpd
tivoli:zAs[U?EH
#
```

This will display the current user ID and password for the specified Endpoint. The default user ID is *tivoli*. The administrator may also change the current user ID and password by issuing the following command:

```
wep shaft set httpd userid:password
```

To go back to changing the configuration through this Web page, once the user ID and password are properly entered, the configuration changes are

applied to the AS/400 Endpoint, and the Endpoint is restarted with the new configuration.

Note

The AS/400 Endpoint HTTP is not a general purpose HTTP server. Applications cannot extend it to do other things. It can only make changes to the AS/400 Endpoint daemon setting. It cannot run any CGI scripts or other programs, such as Java applets. In other words, the security implication is to prevent it from sending an AS/400 Endpoint daemon to another Endpoint Gateway. The `http_disable` option of the AS/400 Endpoint daemon can make these pages read-only.

5.2.2.3 Modifying AS/400 Endpoint settings by Web interface

Since AS/400 does not have a standard editor, such as vi editor for UNIX, to modify the `last.cfg` file, using Endpoint Web interface is easier than editing the `last.cfg` file directly. In this section, we introduce how to change the AS/400 Endpoint option setting using an Endpoint Web interface.

Opening the Network Address Configuration page

To use the Endpoint Web interface to reconfigure the Endpoint, you need to know the hostname, port, user ID, and password for the Endpoint. These are all obtainable through the `wep` command. Use the URL `http://hostname:port` to connect the Endpoint Web interface. In this example, we use the same AS/400 Endpoint as the previous section. Once you have connected to the AS/400 Endpoint, click on the **Network Address Configuration** menu and the Location Configuration page will appear (see Figure 49 on page 129).

Reconfiguring the AS/400 Endpoint setting

On the Location Configuration page, you can enter Endpoint options in the Additional configuration options field. In this example, we will change the `login_interval` option as shown in Table 14.

Table 14. Reconfiguring the Endpoint `login_interval` option

Case	<code>login_interval</code>
Default	1,800 (seconds)
After reconfiguring	900 (seconds)

To change the `login_interval` value, type in the `login_interval` option with the appropriate syntax as shown in Figure 50 on page 131. Refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more detailed information about Endpoint options.

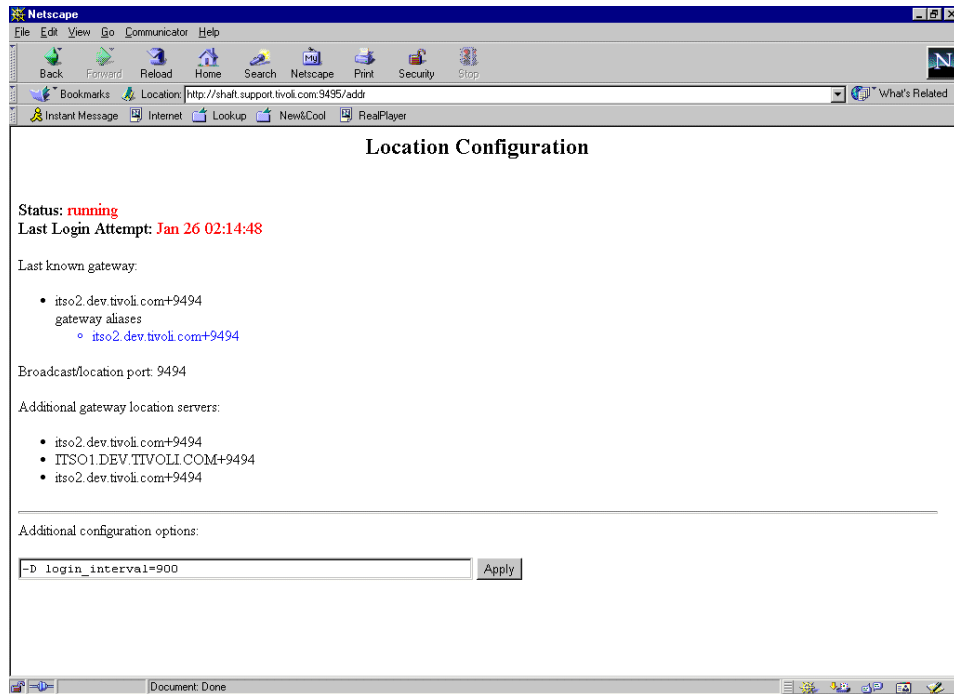


Figure 50. Reconfiguring Endpoint option setting

Once you have typed the appropriate Endpoint options in the Additional configuration options field and pressed the **Apply** button, the dialog box shown in Figure 51 will be displayed.

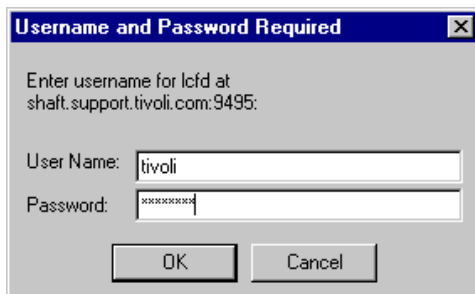


Figure 51. Password dialog box

In this case, the AS/400 Endpoint has already logged into the Endpoint Gateway; the password can be obtained by issuing the `wep` command (refer to the previous section for more information). Enter the appropriate user ID and

password in the dialog box and press **OK**. The AS/400 Endpoint will be restarted with the modified configurations. The reset message will be shown as shown in Figure 52.

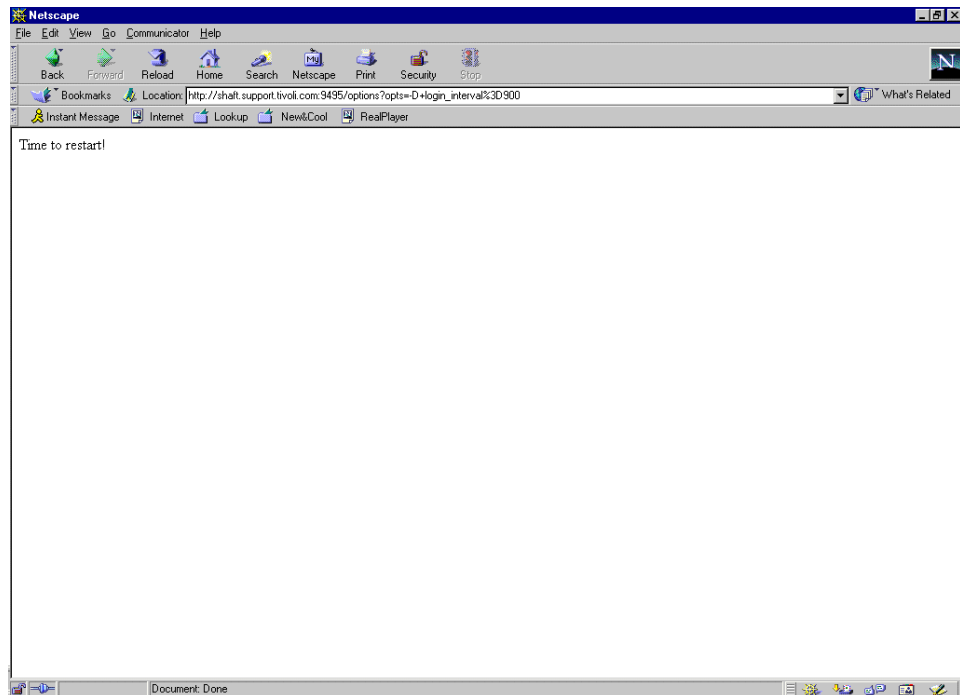


Figure 52. AS/400 Endpoint reset message

After this operation, go back to the Show Config Settings page and check the login_interval option setting. It should be set to 900 (seconds) as shown in Figure 53.

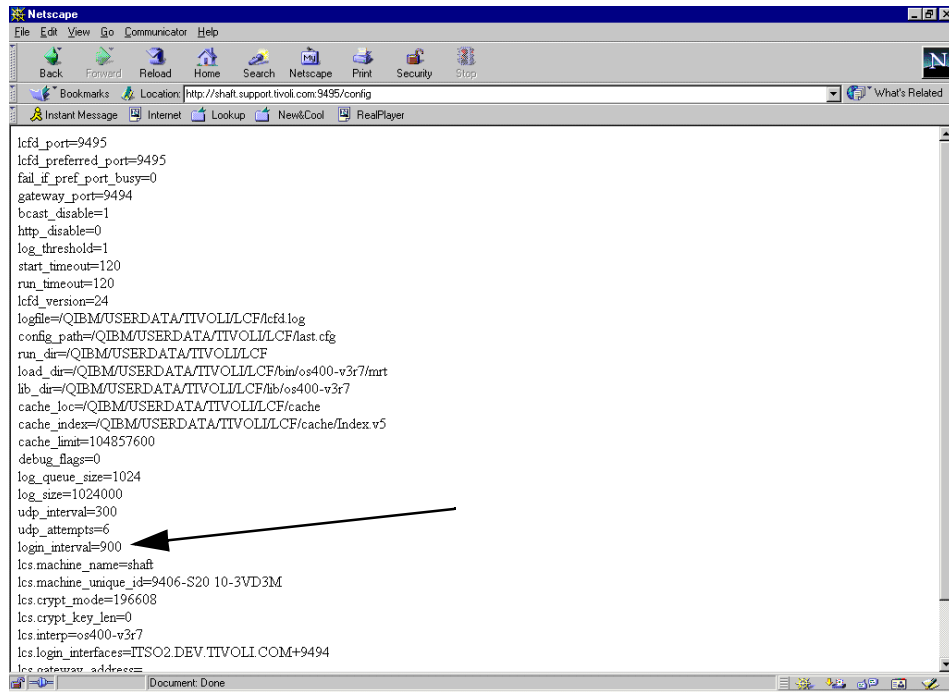


Figure 53. Reconfiguration result

Note

By default, the user ID for the Endpoint Web interface is *tivoli*, and the initial password until completion of the initial login is *boss*. After the completion of the initial login, the password can be obtained by issuing the `wep` command as shown in the previous section.

Normally the default user ID and password are used for changing the `login_interface` setting when the Endpoint failed to login to some Endpoint Gateways.

5.2.3 Upgrading AS/400 Endpoint daemon

As we described in Section 5.1.4, “AS/400 Endpoint method management” on page 122, the Endpoint method is automatically upgraded when you upgrade the version of Tivoli management application on the Endpoint Gateway. What about the AS/400 Endpoint daemon? Is the AS/400 Endpoint daemon upgraded automatically as well as the Endpoint methods? The answer to this question is *no*; the AS/400 Endpoint daemon should be upgraded manually

when you upgrade Tivoli Management Framework. Normally, you can use the following two methods to upgrade the AS/400 Endpoint daemon.

- Using the `wadminep` command
- Using Endpoint policy (`login_policy`)

The following sections introduce how to upgrade the AS/400 Endpoint daemon by using these methods.

5.2.3.1 Using the `wadminep` command

Using the `wadminep` command is the most simple and common way of upgrading the AS/400 Endpoint daemon. The following is the `wadminep` command syntax.

```
wadminep ep_label upgrade
```

Refer to the *TME 10 Framework 3.6 Reference Manual*, SC31-8434, for more information about the `wadminep` command.

Table 15 shows the relationship between the Endpoint version and Tivoli Management Framework patch level.

Table 15. Endpoint version and Tivoli Management Framework patch

Endpoint version	Framework patch
5	Framework 3.6 Install CD-ROM
	Patch 3.6-TMF-0013
6	Patch 3.6-TMF-0020
	Patch 3.6-TMF-0047
8	Patch 3.6-TMF-0024
9	Patch 3.6-TMF-0034
10	Patch 3.6-TMF-0040
11	Patch 3.6-TMF-0046
20	Framework 3.6.1 Maintenance Release
	Patch 3.6.1-TMF-0034
21	Patch 3.6.1-TMF-0004
	Patch 3.6.1-TMF-0055
22	Patch 3.6.1-TMF-0011
23	Patch 3.6.1-TMF-0016

Endpoint version	Framework patch
24	Patch 3.6.1-TMF-0025
40	3.6.2 Framework Maintenance Release

When you apply the above Tivoli Management Framework patches in your TMR, especially the Endpoint Gateways, you should upgrade the AS/400 Endpoint daemons that are managed by the Endpoint Gateways.

Checking the version of AS/400 Endpoint daemon

Before upgrading the AS/400 Endpoint daemon, we strongly recommend that you check the current version of the AS/400 Endpoint daemon. You can check the version of the AS/400 Endpoint daemon using the `wadminep` command as follows:

```
wadminep ep_label view_version
```

When you perform this command, the following output will be displayed:

```
# wadminep ishi11 view_version
Performing browse mode 'view_version' on endpoint 'ishi11'
24
#
```

This means that the version of the AS/400 Endpoint daemon (ishi11) is 24. You can also use the Endpoint Web interface to check the Endpoint level.

Note

The `wadminep` command provides many useful functions for managing Endpoints. Refer to the redbook, *All About Tivoli Management Agents*, SG24-5134, for more detailed information about the `wadminep` command. However, use this command with caution. Most of options are not officially supported by Version 3.6.1 of Tivoli Management Framework. Only the *upgrade* option is officially supported.

Upgrading the AS/400 Endpoint daemon

If you need to upgrade the AS/400 Endpoint daemon, the *upgrade* option of the `wadminep` command allows you to perform upgrading. The following example shows how to upgrade the AS/400 Endpoint daemon:

```
# wadminep ishiil upgrade
Performing administrative mode 'upgrade' on endpoint 'ishiil'
Upgrading endpoint 'ishiil'
Endpoint interpreter = os400-v3r7
WARNINGS INTERCEPTED from 1790075743.47.509+:
^
K
SUCCESSFULLY upgraded endpoint 'ishiil'
```

After completing this process, we strongly recommend that you check the version of the AS/400 Endpoint daemon again using the `wadminep` command and make sure the upgrade has completed.

5.2.3.2 Using Endpoint login_policy

The Endpoint `login_policy` allows you to automate the upgrade process. As we mentioned earlier, the Endpoint `login_policy` is run on the Endpoint Gateway at every normal login and performs any action you need. If you create a script that contains the Endpoint upgrade process, this script enables you to upgrade the AS/400 Endpoint daemon automatically. Figure 54 on page 137 shows this implementation.

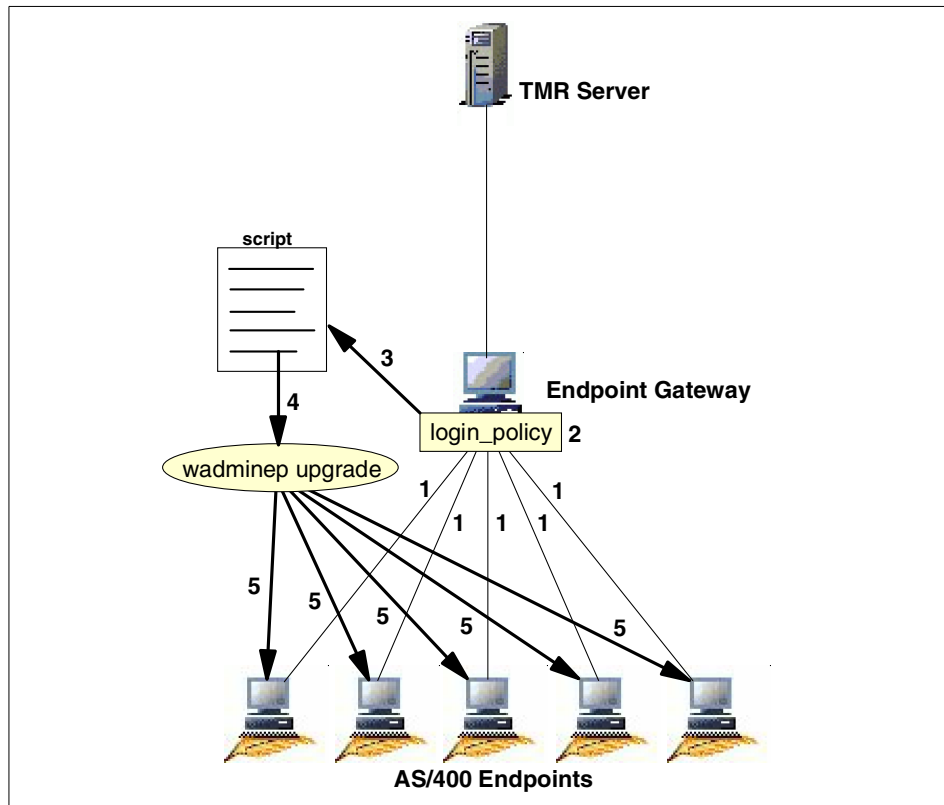


Figure 54. Using Endpoint login_policy for upgrading AS/400 Endpoint daemon

To upgrade an endpoint daemon, perform the following steps:

1. AS/400 Endpoints attempt to perform login to the assigned Gateway.
2. When AS/400 Endpoints perform login to the Endpoint Gateway, the login_policy is run on the Endpoint Gateway.
3. The login_policy executes the script that is defined.
4. This script checks the current version of the AS/400 Endpoint daemon using the wadminep view_version option.
5. If the script detects that the available version is greater than the current version of the AS/400 Endpoint daemon, the script executes the wadminep upgrade option. As a result, the AS/400 Endpoint daemon is automatically upgraded.

Note

Use this implementation with caution. The upgrade process of the `wadminep` command takes about a minute. If a single Endpoint Gateway manages several hundreds AS/400 Endpoints and they attempt to login to the Endpoint Gateway at the same time, the upgrade process in the `login_policy` may affect adversely the Endpoint Gateway performance. We strongly recommend that you test this configuration in your environment before implementation, and make sure of the performance of your Endpoint Gateway.

Chapter 6. AS/400 security management

This chapter will examine the topic of performing user administration and security management on the AS/400 using the Tivoli User Administration and Tivoli Security Management products. We will begin with a brief description of each of these products. Following this, we will consider each product in further detail specifically examining its functions in relation to the AS/400. We also suggest possible customizations and provide a sample Task Library to illustrate ways to enhance the existing product functions to meet unique environmental requirements. We conclude our discussion of each product by showing several scenarios, actually performing OS/400 user administration and security management tasks using the Tivoli products. Before proceeding however, it is extremely important to note that this chapter is, in no way, an attempt to present either a complete reference on these Tivoli products or the topic of these AS/400 system administration functions. For complete details of Tivoli User Administration, Tivoli Security Management, Tivoli for AS/400 Endpoints, and Tivoli Task Libraries, refer to the following documents.

- *User Admin 3.6.2 User's Guide*, GC32-0291
- *Security Management User's Guide 3.6*, GC31-8475
- *Tivoli Framework AS/400 3.6.2 Endpoints User's Guide*, GC32-0279
- *Tivoli for AS/400 Endpoints Release Notes*, GI10-8016
- *Tivoli User Administration Design Guide*, SG24-5108
- *Tivoli Security Management Design Guide*, SG24-5101
- *Tivoli Enterprise Internals and Problem Determination*, SG24-2034
- *TME 10 Task Library 3.6 Developer's Guide*, SC31-8436

For further information on AS/400 user administration, security management, and complete details on the OS/400 commands we will be discussing, refer to the following publications as appropriate to your OS/400 version.

- *AS/400 System Operation V3.6.0*, SC41-4203
- *AS/400 Security - Basic V4R1*, SC41-5301
- *OS/400 Security - Reference V4R4*, SC41-5302
- *OS/400 CL Reference V3R2*, SC41-3722
- *AS/400 Programming Work Management Guide*, SC21-8078

6.1 Overview of Tivoli security management applications

With Version 3.6.x, Tivoli has extended its management ability to the AS/400 platform. The Tivoli security management solution covers AS/400 systems as well as other platforms, and it enables you to perform seamless operations in security management.

The Tivoli security management solution consists of the following Tivoli applications.

- Tivoli User Administration
- Tivoli Security Management

These two applications rely upon the Tivoli Management Framework for delivery services. The following sections introduce the overview of Tivoli security management applications.

6.1.1 Tivoli User Administration

Tivoli User Administration is a profile-based application that runs on the Tivoli Framework. The base product manages UNIX, Windows NT, and NetWare user and group accounts. Additional products extend Tivoli User Administration to manage accounts on other platforms including Lotus Domino/Notes, the OS/390 Security Server, and OS/400. By storing account information in platform-independent records, Tivoli User Administration is able to add, remove, and update users across a distributed enterprise from a centralized location.

It needs to be stressed that the most important part of a Tivoli User Administration implementation is prior planning. This is even more important when you are considering using the product to manage AS/400 Endpoints because these endpoints have the potential to add a substantially large number of user records to Tivoli user profiles. If you do not have a sound detailed design, you will have problems implementing and supporting these resources with User Administration. We highly recommend working through the *Tivoli User Administration Design Guide*, SG24-5108, listed previously before attempting to architect and implement your solution. With thorough planning, many organizations will make unique time-saving customizations to User Administration, such as modifying policies or altering the GUI operation using the Tivoli Application Extension Facility (AEF).

6.1.2 Tivoli Security Management

While Tivoli User Administration manages the creation and authentication of users, Tivoli Security Management manages the access control those users

will encounter with resources. Tivoli Security Management does not add an additional layer of authentication for the end user. Rather, Tivoli Security Management works with the access control mechanisms of each supported endpoint type to control the user's access rights.

Access is managed in a hierarchical fashion. Users are collected together into groups that fit into the company organizational structure. That group is then assigned roles that correspond to the job functions members of that group will perform. The roles define the access capabilities for sets of resources. In addition, there is a set of attributes that can be used to implement the system security policy.

All four of these security record types, groups, roles, resources, and system policy, are maintained in a system-independent format. This allows the security profiles containing them to be distributed across the various supported architectures, thereby, providing a means to centrally define and enforce a consistent security policy enterprise-wide.

6.2 Tivoli User Administration for AS/400

Tivoli User Administration stores user and group records within profile objects called user profiles and group profiles respectively. To avoid confusion, it is important to remember that these two terms (user profiles and group profiles) are not unique to the Tivoli User Administration product. In fact, user and group profiles are specific OS/400 objects. Perhaps a quick comparison of these terms is in order.

As discussed, Tivoli user profiles and Tivoli group profiles are collections of platform-independent user and group account records. A Tivoli user profile is a collection of user records, each containing general user information as well as information concerning one or more account types. Similarly, a Tivoli group profile is a collection of group records, each containing all the information needed to define UNIX group accounts.

An AS/400 user profile is a specific OS/400 architected object containing environmental and security-related information. It controls such things as:

- How the user signs on to the system
- The initial program and/or menu that runs when the user signs on
- What options the user will see on system menus
- The user's working environment
- What system functions the user will be authorized to perform

- Which objects the user owns
- What private authority the user has with the objects
- How the system audits the user's actions

An AS/400 group profile is a unique instance of an AS/400 user profile. It is used to define a group of user profiles that need to access the same applications in the same manner and with the same authority. Typically, these users work in the same department or have similar job responsibilities. In most cases, these users have the same customization needs, such as a specific initial menu, library list, and default printer. On the AS/400, these can be defined in a group profile that can then be copied when creating new individual user profiles who need these customizations. Additionally, an AS/400 group profile can be used as a security tool allowing you to define object authorities for the single group profile rather than for each user profile that is a member of the group.

Note

While Tivoli User Administration performs both user and group management, only user management is supported on an AS/400 Endpoint. Note however, that, since an AS/400 Group Profile is specifically an AS/400 User Profile object, it is still possible to manipulate these using Tivoli User Administration. Additionally, the Tivoli Security Management for AS/400 Endpoints product can be used to both create AS/400 Group Profiles and add AS/400 user profiles as members in it. Refer to the following sections for details.

Throughout this chapter, we will use fully-qualified names, Tivoli user profile or AS/400 user profile, whenever necessary to avoid ambiguity.

6.2.1 Installing Tivoli User Administration for AS/400

Tivoli User Administration for AS/400 Endpoints consists of optionally-installable patches that allow the Tivoli User Administration Version 3.6.1 product to support AS/400 Endpoints. Therefore, before installing these patches, Tivoli User Administration, Version 3.6.1 base product, and the AS/400 Endpoint must be installed. For complete details on installing Tivoli User Administration, Version 3.6.1, refer Chapter 2 of the *User Admin 3.6.2 Users Guide*, GC32-0291. Refer to Chapter 4, “AS/400 Endpoint configuration” on page 61 of this book, for AS/400 Endpoint installation details including:

- Hardware requirements

- Software requirements
- Installation methods
- A complete AS/400 Endpoint enablement patch listing

Additionally, be sure to completely review the *Tivoli AS/400 3.6.1 Release Notes*, GI10-8016, because this document provides the most current information for the product including:

- New features
- Installation notes
- Documentation changes
- Defects, workarounds, and limitations

The enablement patches necessary for Tivoli User Administration for AS/400 Endpoints are described in the following sections.

3.6.1-ADM-0001

TME 10 User Administration for AS/400 Endpoints, Version 3.6.1 contains graphical user interface extensions necessary to support AS/400 Endpoints. It is very important that this patch be installed *only* on the TMR server. Installing this patch on a managed node will corrupt the installation.

3.6.1-ADM-0002

TME 10 User Administration Gateway for AS/400 Endpoints, Version 3.6.1

This patch contains the User Administration methods which the User Administration gateway will need to support AS/400 Endpoints. This patch needs to be installed on any User Administration gateway that will support AS/400 Endpoints.

Note

Just as when managing TMA Endpoints it is a good rule of thumb to install the User Administration gateway on all of the endpoint gateways, it is also recommended that you install this TME 10 User Administration Gateway for AS/400 Endpoints patch on all of your User Administration Gateways. By doing this, you are assured that the AS/400 endpoints will remain supported should you reconfigure your gateways or move the AS/400 endpoints.

6.2.2 Understanding Tivoli User Administration for AS/400

Tivoli User Administration for AS/400 Endpoints provides the tools needed to manage user accounts on OS/400 systems and, at the same time, integrates this support with other operating system types in the network. By extending the User Administration functions, the product provides unique features, specifically, in support of the AS/400 Endpoint.

6.2.2.1 Profile Population

Populating Tivoli user profiles imports user account information from specified endpoints into user records within the user profile. You can populate a Tivoli user profile using the Tivoli Desktop or by using the `wpopusrs` command. Populating from the desktop will put every user defined on the source system into the profile. The scenario described in Chapter 9, “AS/400 management scenarios” on page 437, illustrates the use of the desktop to populate a user profile.

On systems with large numbers of users, such as many AS/400s support, putting all the user records into one profile manager is not recommended. Instead, Tivoli recommends creating several user profiles, tailored to your business environment, to divide the user records into manageable groups. In such cases, the `wpopusrs` command can be used to populate each user profile from a list of users provided in a file. The population scenario in Section 6.2.5, “AS/400 User Administration management scenarios” on page 181, will use this method.

When using the `wpopusrs` command, the `-M` flag can be used to specify that the incoming records be merged with the user records already existing in the Tivoli user profile. Tivoli User Administration will then attempt to merge the incoming user information with existing user records using the login name as the key to determine a match. When a match is identified, the operating system types are compared. If they are different, the new information will be merged into the existing user record. When the user information is merged with a user record, the new operating-specific information is added. The general account information is merged on an attribute-by-attribute basis, with the existing user record information taking precedence.

To assist in using the `wpopusrs` command to populate a Tivoli user profile with OS/400 user profile information, a script called `w4getusrs.pl` is provided. This script can be used to produce files containing lists of AS/400 user profiles. These files can then, in turn, be used as input to the `wpopusrs -f` command.

The script is located in the `/usr/Tivoli/bin/lcf_bundle/generic` directory and allows the following syntax:


```
w4getusr.pl {keyword_options} endpoint_name
```

and the keywords:

<code>-d</code> directory	Destination directory where the output files are created
<code>-a</code> number	Maximum number of entries to put in each file

The command will create an ordered list of files named `usr0`, `usr1`, `usr2`, and so on, each containing, at most, the number specified by `-a` user profile names in the destination directory.

When you populate a Tivoli user profile, the default policy on the user profile is not used. Rather, the values provided from the source system are copied into the user record. Unspecified attributes are left empty. Validation policy, however, can be run against the populated records to verify that all the information complies with defined policy.

Since the AS/400 does not externalize passwords, when you initially populate a user profile from an AS/400 endpoint, the user's password will be initialized to the User Profile name. Passwords will be discussed in further detail in a later section.

6.2.2.2 Profile management

By default, when you create a new user record, Tivoli User Administration defines attributes for UNIX, Netware, and Windows NT user accounts. Additionally, with Tivoli User Administration for AS/400 Endpoints installed, OS/400 attributes will also be created. If the subscribers of the Tivoli user profile do not include each of these platform types, the process of creating a new user will be unnecessarily slowed by the generation of attributes that will not be used; so, one of the first tasks that you will want to perform, once you have planned your subscribers, will be to run the `wsetdefpol` command to disable any UNIX, NT, or Netware default policies if they are not applicable. For example, the following command disables Netware default policy on the DeptA user profile:

```
wsetdefpol DISABLED NW DeptA
```

For specific information on the `wsetdefpol` command, refer to Appendix D of the User Administration User and Group Management Guide.

In order to support the AS/400 Endpoint, Tivoli User Administration provides a large number of AS/400 attributes with Tivoli user records. The Tivoli desktop helps manage these new attributes by organizing them into 10 OS/400 subcategories. For each subcategory, we provide an illustration of the

panel and a chart showing each attribute's name, default value, valid value, and the OS/400 command parameter with which it is associated.

The attribute names are significant because they are the means by which the attribute values can be accessed via the `wgetusr` and `wsetusr` commands.

OS/400 Auditing Information

The following OS/400 Auditing Information panel allows you to set up or change the auditing that OS/400 will perform for a user.

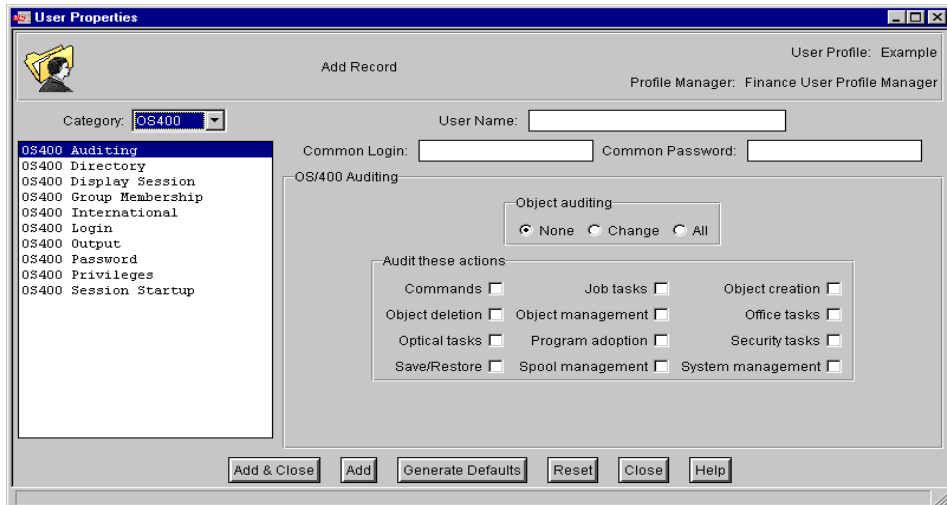


Figure 55. OS/400 auditing attributes

Table 16 lists the attributes shown in Figure 55.

Table 16. OS/400 auditing attributes

AS/400 attributes	Default value	Valid values	OS/400 command parameter
OS400_Obj_Audit	*NONE	*NONE *CHANGE *ALL	CHGUSRAUD OBJAUD(*NONE) CHGUSRAUD OBJAUD(*CHANGE) CHGUSRAUD OBJAUD(*ALL)
OS400_Command_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*CMD...)

AS/400 attributes	Default value	Valid values	OS/400 command parameter
OS400_Job_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*JOBDA...)
OS400_Obj_Create_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*CREATE...)
OS400_Obj_Delete_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*DELETE...)
OS400_Obj_Mgmt_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*OBJMGT...)
OS400_Office_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*OFCSR...)
OS400_Optical_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*OPTICAL...)
OS400_PGM_Adopt_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*PGMADP...)
OS400_Security_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*SECURITY...)
OS400_SAVRST_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*SAVRST...)
OS400_Spool_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*SPLFDATA...)
OS400_Sys_Mgmt_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*SYSMGMT...)
OS400_Service_Audit	No	No Yes	CHGUSRAUD AUDLVL(...) CHGUSRAUD AUDLVL(*SERVICE...)

Note

In this table, “...” indicates that other values may be present. For example,

CHGUSRAUD AUDLVL(...)

Suppose that, on this panel, the following selections are made:

Object auditing Change

Audit these actions Save/Restore and Systems Management

This has the effect of setting the OS400_Obj_Audit value to *CHANGE and the OS400_SAVRST_Audit and OS400_Sys_Mgmt_Audit values to Yes.

Therefore, the same results could have been achieved by running the following command:

```
wsetusr -x OS400_Obj_Auditing *CHANGE -x OS400_SAVRST_Audit Yes -x
OS400_Sys_Mgmt_Audit Yes @UserProfile:user_profile user
```

Additionally, once these actions have been taken, the following results will be seen:

```
wgetusr -x OS400_Obj_Auditing @UserProfile:user_profile user
*CHANGE
wgetusr -x OS400_SAVRST_Audit @UserProfile:user_profile user
Yes
wgetusr -x OS400_Sys_Mgmt_Audit @UserProfile:user_profile user
Yes
wgetusr -x OS400_Obj_Create_Audit @UserProfile:user_profile user
No
```

We will discuss the OS/400 Command Parameter column in Section 6.2.2.3, “Profile distribution” on page 162.

OS/400 Directory Information

The following OS/400 Directory Information panel, shown in Figure 56 on page 149, allows you to set up or change a user’s current library and home directory settings. It also allows you to set up or change values relating to jobs and storage associated with the user.

Figure 56. OS/400 directory attributes

Table 17 lists the attributes shown in Figure 56.

Table 17. OS/400 directory attributes

AS/400 attribute	Default value	Valid value	OS/400 command parameter
OS400_Current_Lib	*CRTDFT	*CRTDFT current_library_name	CHGUSRPRF LIB(*CRTDFT) CHGUSRPRF LIB(current_lib_name)
OS400_Max_Storage	*NOMAX	*NOMAX max_kbytes	CHGUSRPRF MAXSTG(*NOMAX) CHGUSRPRF MAXSTG(max_kbytes)
OS400_Priority	3	number	CHGUSRPRF PTYLMT(number)
OS400_Accounting_Code	*BLANK	*BLANK accounting_code	CHGUSRPRF ACGCDE(*BLANK) CHGUSRPRF ACGCDE(accounting_code)

AS/400 attribute	Default value	Valid value	OS/400 command parameter
OS400_JOBID	*LIBL/QDFTJOBID	*LIBL/QDFJOBID *CURLIB/ <i>jobd lib/jobd</i>	CHGUSRPRF JOBID(*LIBL/QDFTJOBID) CHGUSRPRF JOBID(*CURLIB/ <i>jobd</i>) CHGUSRPRF JOBID(<i>lib/jobd</i>)
OS400_Directory	*USRPRF	*USRPRF <i>/home_directory_path/</i>	CHGUSRPRF HOMEDIR(*USRPRF) CHGUSRPRF HOMEDIR(* <i>/home_dir_path/</i>)

OS/400 Display Session Information

The OS/400 Display Session panel, shown in Figure 57, allows you to specify values that control the functions of the user's display session.

Figure 57. OS/400 display session attributes

Table 18 lists the attributes displayed in Figure 57 on page 150.

Table 18. OS/400 display session attributes

AS/400 attribute	Default value	Valid values	OS/400 command parameters
OS400_Attention_PGM	*SYSVAL	*SYSVAL *ASSIST *NONE *LIBL/ <i>program</i> *CURLIB/ <i>program lib</i> / <i>program</i>	CHGUSRPRF ATNPGM(*SYSVAL) CHGUSRPRF ATNPGM(*ASSIST) CHGUSRPRF ATNPGM(*NONE) CHGUSRPRF ATNPGM(*LIBL/ <i>program</i>) CHGUSRPRF ATNPGM(*CURLIB/ <i>program</i>) CHGUSRPRF ATNPGM(<i>lib</i> / <i>program</i>)
OS400_Limit_Cap	*NO	*NO *PARTIAL *YES	CHGUSRPRF LMTCBP(*SYSVAL) CHGUSRPRF LMTCBP(*PARTIAL) CHGUSRPRF LMTCBP(*YES)
OS400_Assist_Level	*SYSVAL	*SYSVAL *BASIC *INTERMED *ADVANCED	CHGUSRPRF ASTLVL(*SYSVAL) CHGUSRPRF ASTLVL(*BASIC) CHGUSRPRF ASTLVL(*INTERMED) CHGUSRPRF ASTLVL(*ADVANCED)
OS400_KBD_Buffering	*SYSVAL	*SYSVAL *NO *TYPEAHEAD *YES	CHGUSRPRF KBDBUF(*SYSVAL) CHGUSRPRF KBDBUF(*NO) CHGUSRPRF KBDBUF(*TYPEAHEAD) CHGUSRPRF KBDBUF(*YES)
OS400_Show_Parm_Keys	No	No Yes	CHGUSRPRF USROPT(...) CHGUSRPRF USROPT(*CLKWD)
OS400_Show_Details	No	No Yes	CHGUSRPRF USROPT(...) CHGUSRPRF USROPT(*EXPERT)
OS400_Change_Rollkey	No	No Yes	CHGUSRPRF USROPT(...) CHGUSRPRF USROPT(*ROLLKEY)
OS400_Show_Status_MSGS	No	No Yes	CHGUSRPRF USROPT(*NOSTMSG) CHGUSRPRF USROPT(*STMSG)
OS400_Display_Help	No	No Yes	CHGUSRPRF USROPT(...) CHGUSRPRF USROPT(*HLPFULL)
OS400_Send_Message	No	No Yes	CHGUSRPRF USROPT(...) CHGUSRPRF USROPT(*PRTMSG)

Note

In this table, “...” indicates that other values may be present. For example,
CHGUSRPRF USROPT(...)

OS/400 Group Membership information

The OS/400 Group Membership panel, shown in Figure 58, allows you to specify the OS/400 group profiles to which this user profile will belong.

The screenshot shows the 'User Properties' dialog box with the 'OS/400' category selected. The 'OS/400 Group Membership' section is active, displaying a list of OS/400 attributes on the left and a list of group names on the right. The 'Primary Group' field is empty, and the 'Supplemental Groups' list is also empty. The 'Add' button is highlighted, and the 'Remove' button is disabled. The 'Add & Close' button is at the bottom left, and the 'Add', 'Generate Defaults', 'Reset', 'Close', and 'Help' buttons are at the bottom right.

Figure 58. OS/400 group membership attributes

Table 19 lists the attributes displayed in Figure 58.

Table 19. OS/400 group membership attributes

AS/400 attribute	Default value	Valid value	OS/400 command parameters
OS400_Prim_GRP	*NONE	*NONE group_name	CHGUSRPRF GRPPRF(*NONE) CHGUSRPRF GRPPRF(group_name)
OS400_Supp_GRPs	*NONE	*NONE group1 - group15	CHGUSRPRF GRPPRF(*NONE)

OS/400 International Information

The OS/400 International Information panel, shown in Figure 59, allows you to set and change values associated with the user's language-related functions.

The screenshot shows the 'User Properties' window with the 'OS/400 International' category selected. The 'OS/400 International' section contains the following fields:

- Sort sequence: [text box]
- Language: [text box]
- Country: [text box]
- Coded character set ID: [text box]
- Locale: [text box]

The 'Job attributes to set from locale' section contains the following checkboxes:

- ☐ None
- ☐ Use system value
- ☐ Coded character set ID
- ☐ Date format
- ☐ Date separator
- ☐ Sort sequence table
- ☐ Time separator

Figure 59. OS/400 international attributes

Table 20 lists the attributes displayed in Figure 59.

Table 20. OS/400 international attributes

AS/400 attribute	Default value	Valid values	AS/400 command parameters
OS400_Sort_Sequence	*SYSVAL	*SYSVAL *HEX *LANGIDSHR *LANGIDUNQ *LIBL/table_name *CURLIB/table_name lib/table_name	CHGUSRPRF SRTSEQ(*SYSVAL) CHGUSRPRF SRTSEQ(*HEX) CHGUSRPRF SRTSEQ(*LANGIDSHR) CHGUSRPRF SRTSEQ(*LANGIDUNQ) CHGUSRPRF SRTSEQ(*LIBL/table_name) CHGUSRPRF SRTSEQ(*CURLIB/table_name) CHGUSRPRF SRTSEQ(*lib/table_name)
OS400_Language	*SYSVAL	*SYSVAL language_id	CHGUSRPRF LANGID(*SYSVAL) CHGUSRPRF LANGID(language_id)
OS400_Country	*SYSVAL	*SYSVAL country_id	CHGUSRPRF CNTRYID(*SYSVAL) CHGUSRPRF CNTRYID(country_id)

AS/400 attribute	Default value	Valid values	AS/400 command parameters
OS400_CCSID	*SYSVAL	*SYSVAL *HEX code_char_set_id	CHGUSRPRF CCSID(*SYSVAL) CHGUSRPRF CCSID(*HEX) CHGUSRPRF CCSID(<i>code_char_set_id</i>)
OS400_Locale	*SYSVAL	*SYSVAL *NONE *C *POSIX locale_path_name	CHGUSRPRF LOCALE(*SYSVAL) CHGUSRPRF LOCALE(*NONE) CHGUSRPRF LOCALE(*C) CHGUSRPRF LOCALE(*POSIX) CHGUSRPRF LOCALE(<i>locale_path_name</i>)
OS400_Loc_None	No	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*NONE)
OS400_Loc_SYSVAL	Yes	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*SYSVAL)
OS400_Loc_CCSID	No	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*CCSID)
OS400_Loc_DATFMT	No	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*DATFMT)
OS400_Loc_DATSEP	No	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*DATSEP)
OS400_Loc_SRTSEQ	No	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*SRTSEQ)
OS400_Loc_TIMSEP	No	No Yes	CHGUSRPRF SETJOBATR(...) CHGUSRPRF SETJOBATR(*TIMSEP)

Note

In this table, “...” indicates that other values may be present. For example,

CHGUSRPRF SETJOBATR(...)

OS/400 Login Information

The OS/400 Login Information options, shown in Figure 60 on page 155, allow you to set and change both values associated with the user’s login as well values associated with deleting the user profile.

The screenshot shows the 'User Properties' dialog box with the 'OS/400' category selected. The 'OS/400 Login' sub-category is highlighted in the left pane. The right pane contains the following fields:

- User Name: [Text Field]
- Common Login: [Text Field] Common Password: [Text Field]
- OS/400 Login:
 - Login: [Text Field]
 - User ID number: [Text Field]
 - Login enablement: [Dropdown Menu: Enable user for processing]
 - Action taken when profile is deleted: [Dropdown Menu: Do not delete, if user owns objects]
 - New owner login: [Text Field]
 - Text description: [Text Field]

Buttons at the bottom: Add & Close, Add, Generate Defaults, Reset, Close, Help.

Figure 60. OS/400 login attributes

Table 21 lists the attributes displayed in Figure 60.

Table 21. OS/400 login attributes

AS/400 attribute	Default value	Valid values	OS/400 command parameters
OS400_Login	login	A valid OS/400 user profiles has up to 10 characters.	CRTUSRPRF USRPRF(<i>login</i>)
OS400_UID	*GEN	uid A value from 1 to 4294967294.	CHGUSRPRF UID(*GEN) CHGUSRPRF UID(<i>uid</i>)
OS400_Login_Enablement	*ENABLED	*ENABLED *DISABLES	CHGUSRPRF STATUS(*ENABLED) CHGUSRPRF STATUS(*DISABLED)
OS400_DLT_Action	*NODLT	*NODLT *DLT *CHGOWN	DLTUSRPRF OWNNOBJOPT(*NODLT) DLTUSRPRF OWNNOBJOPT(*DLT) DLTUSRPRF OWNNOBJOPT(*CHGOWN <i>new_owner</i>)
OS400_DLT_NewOwn		<i>new_owner</i> A valid OS/400 user profile name.	DLTUSRPRF OWNNOBJOPT(*CHGOWN <i>new_owner</i>)

AS/400 attribute	Default value	Valid values	OS/400 command parameters
OS400_Login_Text	*BLANK	*BLANK description (l) Up to 50 characters.	CHGUSRPRF TEXT(*BLANK) CHGUSRPRF TEXT(description)

OS/400 Output Information

The OS/400 Output Information panel, shown in Figure 61, allows you to set the user's printer, output queue, and message queue options.

The screenshot shows the 'User Properties' dialog box with the 'OS/400' category selected. The 'OS/400 Output' panel is active, displaying the following fields and options:

- User Name:** [Text input field]
- Common Login:** [Text input field]
- Common Password:** [Text input field]
- Print device:** [Text input field]
- Output queue:** [Text input field]
- Message queue:** [Text input field]
- Message delivery:** [Dropdown menu set to 'Default']
- Message severity level:** [Text input field]

At the bottom of the dialog are buttons: 'Add & Close', 'Add', 'Generate Defaults', 'Reset', 'Close', and 'Help'.

Figure 61. OS/400 output attributes

Table 22 lists the attributes displayed in Figure 61.

Table 22. OS/400 output attributes

AS/400 attribute	Default value	Valid values	OS/400 command parameters
OS400_Printer_Device	*WRKSTN	*WRKSTN *SYSVAL printer_device	CHGUSRPRF PRTDEV(*WRKSTN) CHGUSRPRF PRTDEV(*SYSVAL) CHGUSRPRF PRTDEV(<i>printer_device</i>)
OS400_OUTQ	*WRKSTN	*WRKSTN *DEV output_queue	CHGUSRPRF OUTQ(*WRKSTN) CHGUSRPRF OUTQ(*DEV) CHGUSRPRF OUTQ(<i>output_queue</i>)
OS400_MSGQ	*USRPRF	*USRPRF *LIBL/ <i>message_queue</i> *CURLIB/ <i>message_queue</i> <i>lib/message_queue</i>	CHGUSRPRF MSGQ(*USRPRF) CHGUSRPRF MSGQ(LIBL/ <i>message_queue</i>) CHGUSRPRF MSGQ(*CURLIB/ <i>message_queue</i>) CHGUSRPRF MSGQ(<i>lib/message_queue</i>)
OS400_MSG_Delivery	*NOTIFY	*NOTIFY *BREAK *HOLD *DFT	CHGUSRPRF DLVRY(*NOTIFY) CHGUSRPRF DLVRY(*BREAK) CHGUSRPRF DLVRY(*HOLD) CHGUSRPRF DLVRY(*DFT)
OS400_MSG_Sev_Level	0	0 severity_code A value from 00 to 99.	CHGUSRPRF SEV(0) CHGUSRPRF SEV(<i>severity_code</i>)

OS/400 Password Information

The OS/400 Password Information options, shown in Figure 62 on page 158, allow you to specify the user's password, set the password as expired, and specify the password expiration interval.

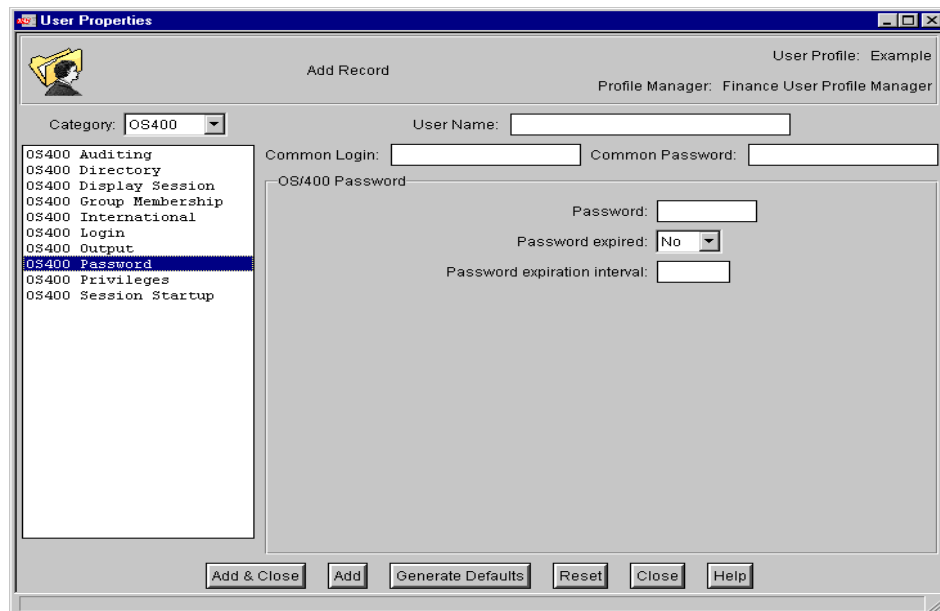


Figure 62. OS/400 password attributes

Table 23 lists the attributes displayed in Figure 62.

Table 23. OS/400 password attributes

AS/400 Attribute	Default Value	Valid Values	OS/400 Command Parameters
OS400_PassW	Sets the user's password to the user's login name.	*NONE user_password	CRTUSRPRF/CHGUSRPRF PASSWORD(*NONE) CRTUSRPRF/CHGUSRPRF PASSWORD(user_password)
OS400_PassW_Expired	*NO	*NO *YES	CHGUSRPRF PWDEXP(*NO) CHGUSRPRF PWDEXP(*YES)
OS400_PassW_ExpInv	*SYSVAL	*SYSVAL *NOMAX expiration_interval A value between 1 and 366.	CHGUSRPRF PWDEXPITV(*SYSVAL) CHGUSRPRF PWDEXPITV(*NOMAX) CHGUSRPRF PWDEXPITV(expiration_interval)

OS/400 Privileges Information

The OS/400 Privileges Information options, shown in Figure 63 on page 159, allow you to set the user's class and special authorities. On the AS/400, the

user's special authority values default to a keyword called *USRCLS. This means that the user's special authorities will be set based on the user class specified. The action of this panel mimics this default behavior. That is, once you set the Privilege class, the system privileges defined by default on the AS/400 for that specified class will be selected for you. However, since these are simply default settings, this can be changed as desired.

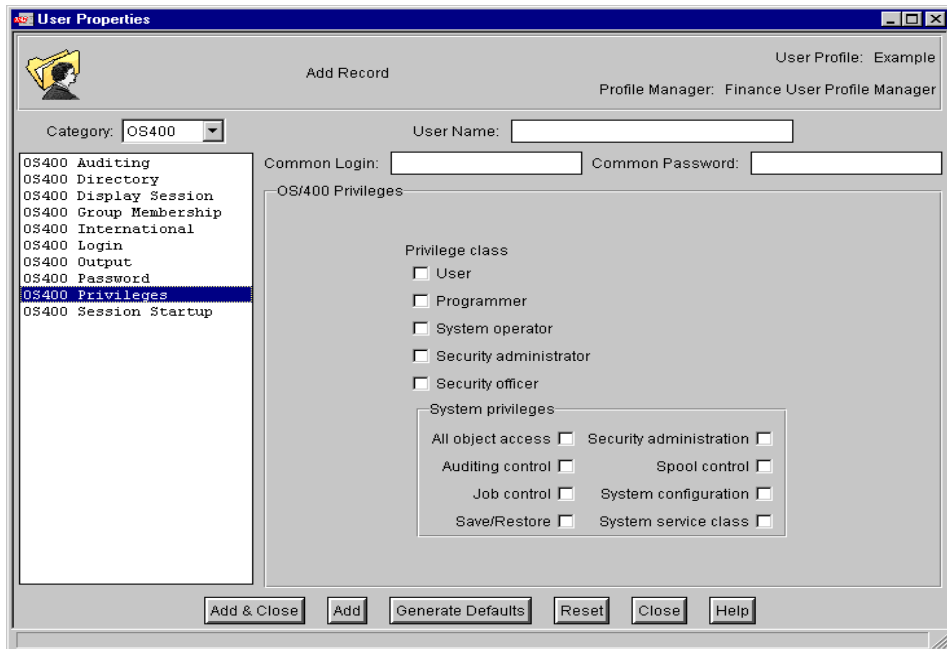


Figure 63. OS/400 privileges attributes

Table 24 lists the attributes displayed in Figure 60.

Table 24. OS/400 privileges attributes

AS/400 attribute	Default value	Valid values	OS/400 command parameters
OS400_Class	*USER	*USER *SYSVAL *PGMR *SECADM *SECOFR	CHGUSRPRF USRCLS(*USER) CHGUSRPRF USRCLS(*SYSVAL) CHGUSRPRF USRCLS(*PGMR) CHGUSRPRF USRCLS(*SECADM) CHGUSRPRF USRCLS(*SECOFR)

AS/400 attribute	Default value	Valid values	OS/400 command parameters
OS400_Priv_ALLOBJ	No	No Yes	CHGUSRPRF SPCAUT(...) CHGUSRPRF SPCAUT(*ALLOBJ)
OS400_Priv_AUDIT	No	No Yes	CHGUSRPRF SPCAUT(...) CHGUSRPRF SPCAUT(*AUDIT...)
OS400_Priv_JOBCTL	No	No Yes	CHGUSRPRF SPCAUT(...) CHGUSRPRF SPCAUT(*JOBCTL...)
OS400_Priv_SAVSYS	No	No Yes	CHGUSRPRF SPCAUT(...) CHGUSRPRF SPCAUT(*SYSVAL...)
OS400_Priv_SECADM	No	No Yes	CHGUSRPRF SPCAUT(...)
OS400_Priv_SPLCTL	No	No Yes	CHGUSRPRF SPCAUT(...)
OS400_Priv_IOSYSCFG	No	No Yes	CHGUSRPRF SPCAUT(...)
OS400_Priv_Service	No	No Yes	CHGUSRPRF SPCAUT(...)

Note

In this table, “...” indicates that other values may be present. For example,
CHGUSRPRF SPCAUT(...)

OS/400 Session Startup Information

The OS/400 Session Startup Information panel options, shown in Figure 64 on page 161, allow you to specify the user’s initial program and menu whether the sign-on information is displayed and whether the user is limited to a certain number of device settings.

The screenshot shows the 'User Properties' window with the 'OS/400' category selected. The 'OS/400 Session Startup' section is expanded, revealing the following fields:

- User Name:** [Empty text box]
- Common Login:** [Empty text box]
- Common Password:** [Empty text box]
- Initial program:** [Empty text box]
- Initial menu:** [Empty text box]
- Display sign-on information:** [Use system value (dropdown)]
- Limit device sessions:** [Use system value (dropdown)]

The left pane lists various OS/400 attributes, with 'OS/400 Session Startup' currently selected. The bottom of the window contains buttons for 'Add & Close', 'Add', 'Generate Defaults', 'Reset', 'Close', and 'Help'.

Figure 64. OS/400 session startup attributes

Table 25 lists the attributes contained in Figure 64.

Table 25. OS/400 session startup attributes

AS/400 attribute	Default value	Valid value	OS/400 command parameters
OS400_Initial_PGM	*NONE	*NONE *LIBL/ <i>program</i> *CURLIB/ <i>program lib/program</i>	CHGUSRPRF INLMNU(*NONE) CHGUSRPRF INLMNU(*LIBL/ <i>program</i>) CHGUSRPRF INLMNU(*CURLIB/ <i>program lib/program</i>)
OS400_Initial_Menu	*LIBL/MAIN	*LIBL/MAIN *SIGNOFF *LIBL/ <i>menu</i> *CURLIB/ <i>menu lib/menu</i>	CHGUSRPRF INLPGM(*LIBL/MAIN) CHGUSRPRF INLPGM(*SIGNOFF) CHGUSRPRF INLPGM(*LIBL/ <i>menu</i>) CHGUSRPRF INLPGM(*CURLIB/ <i>menu lib/menu</i>)
OS400_Signon_Info	*SYSVAL	*SYSVAL *NO *YES	CHGUSRPRF DSPSGNINF(*SYSVAL) CHGUSRPRF DSPSGNINF(*NO) CHGUSRPRF DSPSGNINF(*YES)

AS/400 attribute	Default value	Valid value	OS/400 command parameters
OS400_Limit_Dev_Sess	*SYSVAL	*SYSVAL *NO *YES	CHGUSRPRF LMTDEVSSN(*SYSVAL) CHGUSRPRF LMTDEVSSN(*NO) CHGUSRPRF LMTDEVSSN(*YES)

Note

In certain environments, you may not want these additional AS/400 attributes on every Tivoli user profile. The `wrmprop` command can be used to remove these properties if desired. Refer to Appendix C, “w4rmusprop” on page 517, for a sample script that can be used to automate this further.

6.2.2.3 Profile distribution

Now that we have looked at the AS/400 attributes provided by Tivoli User Administration for AS/400 Endpoints, we will explain how these attributes are used when the Tivoli User Profile is distributed to an AS/400 Endpoint. The *OS/400 Command Parameters* column in the above tables will be explained in detail in this section.

The AS/400 attributes illustrated in the above panels are directly related to OS/400 commands that are used to create, change, audit, and delete AS/400 user profiles. To be precise, by using the attributes shown in the preceding figures, Tivoli User Administration performs the `CRTUSRPRF` (Create User Profile), `CHGUSRPRF` (Change User Profile), `CHGUSRAUD` (Change User Auditing), and `DLTUSRPRF` (Delete User Profile) OS/400 commands upon distributing the profile to an AS/400 Endpoint based on the action performed to the user record.

There are three basic actions that can be performed on a user record within a Tivoli User Profile. Users can be added, edited, and/or deleted. For each of these actions, the following corresponding OS/400 commands result:

- For each user record added, the following OS/400 commands will be run:
 - `CRTUSRPRF USER(login) PWD(password)`
 - `CHGUSRPRF USER(login)` with associated parameters
 - `CHGUSRAUD USER(login)` with associated parameters
- For each user record modified, the following OS/400 commands will be run:
 - `CHGUSRPRF USER(login)` with associated parameters

- CHGUSRAUD USER(*login*) with associated parameters
- Finally, for each user recorded deleted, the following OS/400 command will be run:
 - DLTUSRPRF USER(*login*) with associated parameters

Therefore, in order to understand how Tivoli User Administration for the AS/400 Endpoint works, a good understanding of these commands is helpful. To this end, we will show the panels for each of these commands and then follow these with a discussion of particular aspects of importance related to Tivoli User Administration. Of course, this is not, by any means, an attempt to completely detail these commands.

Note

To see these commands as shown here, type in the command at the AS/400 command line, and then press **F4** (Prompt). This shows the most commonly-used parameters for the command. To see all parameters, press **F9** (All parameters). A keyword is provided for each parameter. To view the keywords, press **F11**.

Additionally, OS/400 provides cursor-sensitive help text. Pressing the **F1** (Help) key while the cursor is positioned on a parameter will provide a detailed description of that value. OS/400 also provides parameter prompting. Pressing the **F4** (Prompt) key while the cursor is positioned on a parameter will provide a listing of appropriate input values.

Finally, keep in mind that changes may be made to OS/400 commands between releases, and review the corresponding documentation accordingly.

In Figure 65 on page 164, we display the Create User Profile command, CRTUSRPRF.

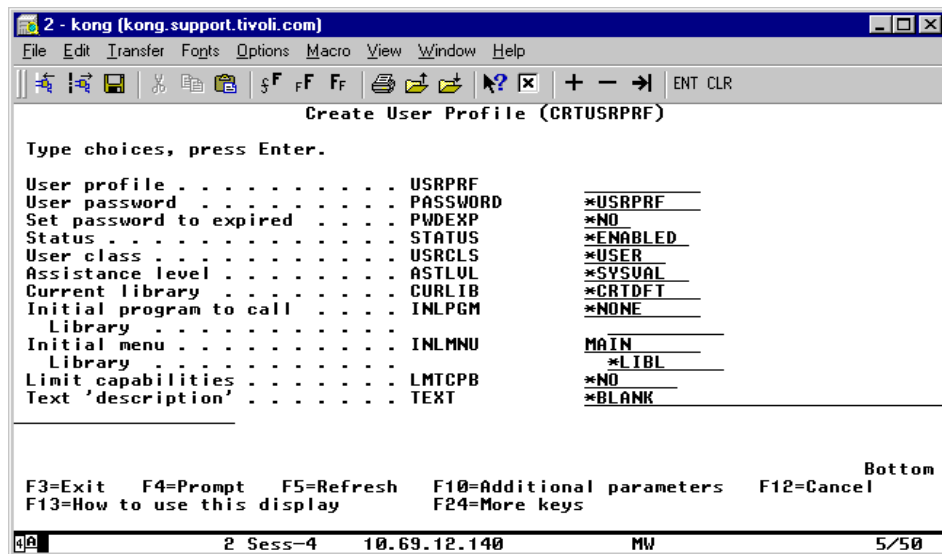


Figure 65. Create user profile (1 of 4)

As shown in Figure 65, the `CRTUSRPRF` commands issued by Tivoli User Administration only specify the `USRPRF` and `PASSWORD` parameters; so, only the first panel of options is shown. The remaining parameters of the `CRTUSRPRF` command are similar to those in the `CHGUSRPRF` command, which we will show in detail next.

The Create User Profile (`CRTUSRPRF`) command creates an OS/400 user profile. Both the user profile and the password can be 10 characters long. The password specified on the `CRTUSRPRF` command is not checked against the AS/400 password validation rules as set up in the System Values. For complete details of these system values, refer to the AS/400 Work Management Guide and the AS/400 Security Reference Guide.

Since this command will initially set the user's password when the user has been created from Tivoli, it may now be a good time to mention how Tivoli will handle a user's passwords. When a user is created from Tivoli, the password is set as specified within Tivoli. Also, any time the password is changed within Tivoli User Administration (using the Tivoli Desktop, the `wsetusr` command, the `wpasswd` command, or the OnePassword Web tool), the user's password will be updated when the user profile is distributed. However, assuming the password from within Tivoli has not been changed, the password value on the AS/400 is not changed when the profile is distributed; so, in this way, the AS/400 user is still able to manipulate his or her password locally without

Tivoli overwriting this selected password unless it has been explicitly set. The following panels illustrate the CHGUSRPRF command. Figure 66 shows page 1 of the user profile.

1 - shaft (shaft.support.tivoli.com)

File Edit Transfer Fonts Options Macro View Window Help

Change User Profile (CHGUSRPRF)

Type choices, press Enter.

User profile	USRPRF	
User password	PASSWORD	*SAME
Set password to expired	PWDEXP	*SAME
Status	STATUS	*SAME
User class	USRCLS	*SAME
Assistance level	ASTLVL	*SAME
Current library	CURLIB	*SAME
Initial program to call	INLPGM	*SAME
Library		
Initial menu	INLMNU	*SAME
Library		
Limit capabilities	LMTCPB	*SAME
Text 'description'	TEXT	*SAME

More...

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

1 Sess-1 10.69.12.131 MW 24/2

Figure 66. Change user profile (page 1 of 4)

Figure 67 on page 166 shows page 2 of the user profile.

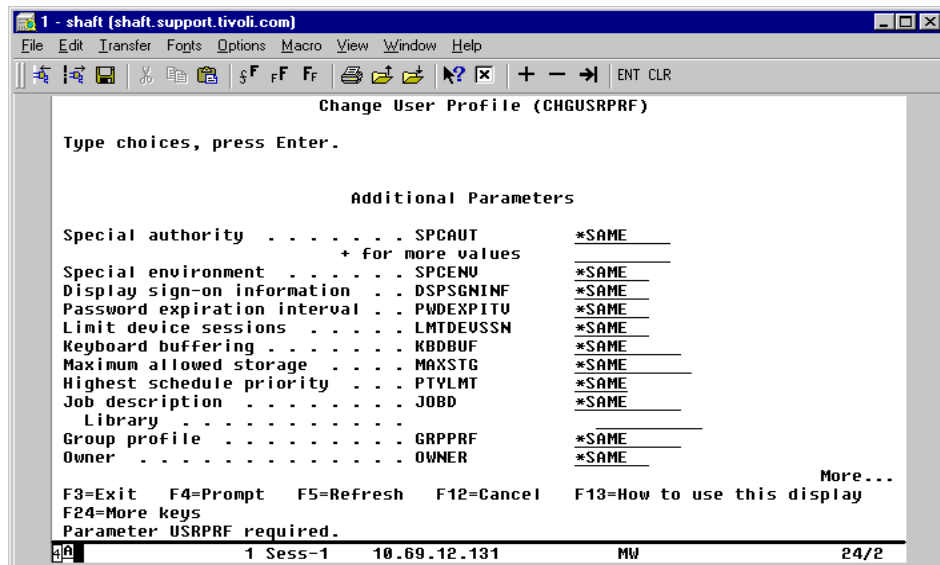


Figure 67. Change user profile (page 2 of 4)

Figure 68 shows page 3 of the user profile.

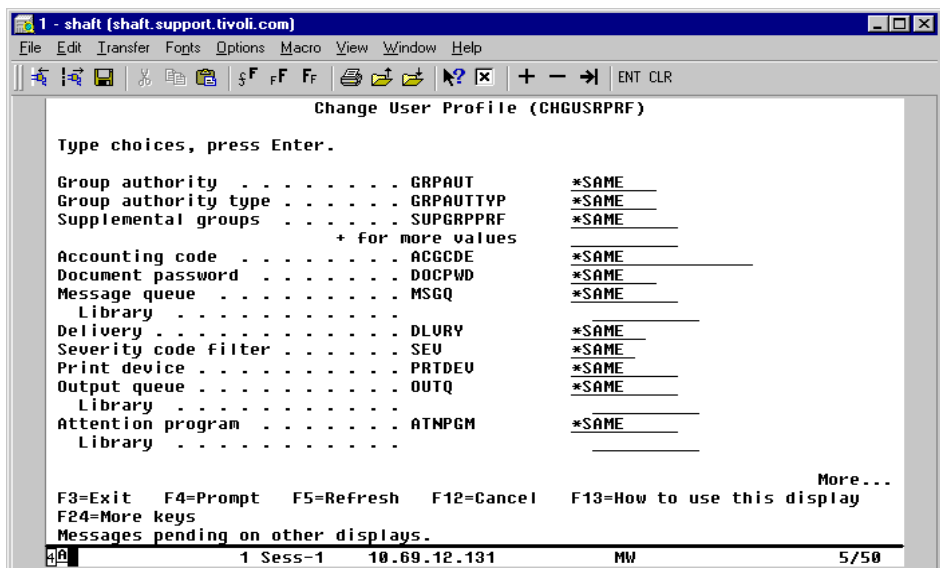


Figure 68. Change user profile (page 3 of 4)

Figure 69 on page 167 shows page 4 of the user profile.

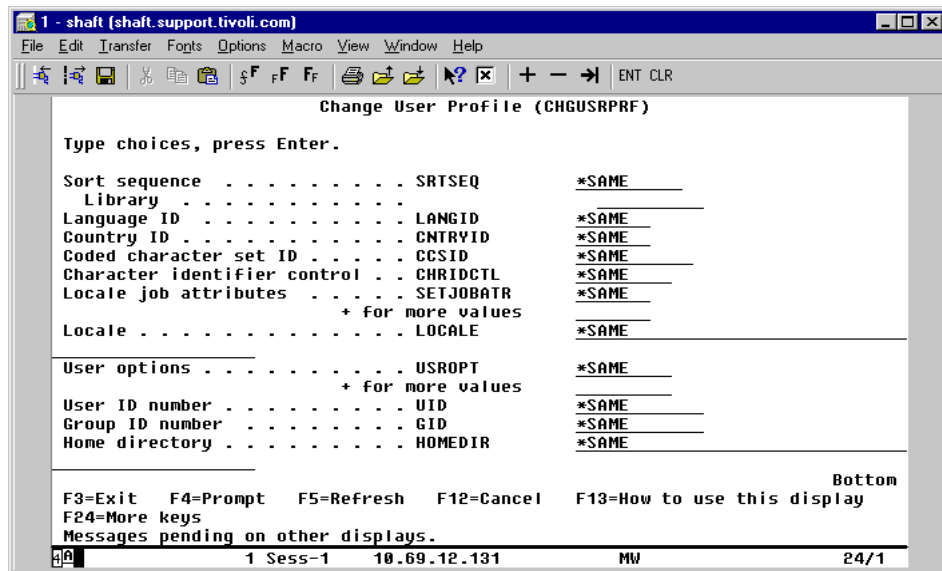


Figure 69. Change user profile (page 4 of 4)

The Change User Profile (CHGUSRPRF) command changes the values specified on an OS/400 user profile. As with the CRTUSRPRF command, the password validation rules, as specified in AS/400 system values, are not verified when a password is changed using this command. Also, the following IBM-supplied user profiles are not valid with this command:

- QDBSHR
- QDOC
- QDSNX
- QFNC
- QGATE
- QLPAUTO
- QDFTOWN
- QMSF
- QSNADS
- QSPLF
- QSPLJOB
- QSYS

- QTSTRQS

Another important thing to point out when using this command is that, while certain values may be permissible, the command may still actually fail if the value specified is not valid on the system at runtime. A couple of examples may help to clarify this. Consider the following commands:

```
CHGUSRPRF USRPRF(TOM) LMTCAP(*YES) GRPPRF(GRP1)
CHGUSRPRF USRPRF(TOM) LMTCAP(*YES) PRTDEV(PRT3812)
```

In both cases, all the supplied values are valid. However, if there is no user profile called GRP1 or if there is no printer device called PRT3812 on the system when these commands are run, they will issue the following messages:

```
CPF2259 "Group Profile GRP1 not found."
CPI2243 "Object PRT3812 in library QSYS not found."
```

In the first case, CPF2259 is a Diagnostic error. The complete message text explains:

Message	Group profile GRP1 not found.
Cause	User profile TOM was not created or changed because the user profile GRP1 specified that the GRPPRF input value does not exist.
Recovery	Either create user profile GRP1 and submit the command again or specify an existing user profile on the GRPPRF input value and try the command again.

Note the instructions to try the command again indicating that the entire CHGUSRPRF command was not performed. Therefore, not only was TOM's Group Profile (GRPPRF) not set to GRP1, but his Limit Capabilities (LMTCAP) was not changed to *YES.

However, the second case is slightly different. In this case, the message issued when the Print Device (PRTDEV) does not exist, CPI2243, is an informational message that simply lets you know that this device does not exist. Immediately following this informational message, completion message CPC2205 is issued:

```
CPC2205 Completion "User profile TOM changed."
```

And, in fact, both TOM's print device (PRTDEV) and the limit capabilities (LMTCAP) values were changed as specified.

Our main point in these examples is to illustrate the value of reviewing the User Administration joblogs for message such as these. In the first example, the Tivoli user record will be out of sync with the actual OS/400 user profile values. In the second example, Tivoli and the AS/400 Endpoint will remain in sync. Nonetheless, TOM will have difficulty printing until either his user profile is changed or a print device named PRT3812 is created. We will discuss obtaining the User Administration joblog further in Section 6.2.4, “Troubleshooting Tivoli User Administration for AS/400” on page 176.

The next OS/400 command that Tivoli will run when adding or modifying a user record is the Change User Auditing command. Its options are shown in Figure 70.

Figure 70. Change user auditing

The Change User Auditing command, `CHGUSRAUD`, allows you to change how a user is audited. It is important to note that there are also several AS/400 system values, `QAUDCTL`, `QAUDLVL`, `QAUDENDACN`, `QAUDFRCLVL`, and `QCRTOBJAUD`, which are used to control auditing on the AS/400. The `QAUDCTL` system value, for example, turns auditing on or off on the AS/400; so, while the user auditing level on the user profile may be specified, auditing is still not actually performed until this system value is set appropriately. Additionally, the `QAUDLVL` system value works in conjunction with the user profile's `AUDLVL` value. For example, if the `QAUDLVL` system value is set to `*DELETE` and the user's `AUDLVL` specifies `*CREATE`, both deleting and

creating actions performed by the user will be audited. For more information on these system values, refer to the *Application System/400 Work Management Guide*, SC21-8078. For complete information on OS/400 Auditing, refer to the *AS/400 System Management Guide*.

The final command to be discussed, Delete User Profile (`DLTUSRPRF`) is issued when a user record in a Tivoli user profile has been deleted. The panel for this command is shown in Figure 71.

```

1 - shaft (shaft.support.tivoli.com)
File Edit Transfer Fonts Options Macro View Window Help
Delete User Profile (DLTUSRPRF)

Type choices, press Enter.

User profile . . . . . USRPRF
Owned object option: . . . . . OWNBJOPT
Owned object value . . . . . *NODLT
User profile name if *CHGOWN
Primary group option: . . . . . PGPOPT
Primary group value . . . . . *NOCHG
New primary group . . . . .
New primary group authority . . . . .

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

Bottom
1 Sess-1  10.69.12.131  MW  24/46

```

Figure 71. Delete user profile

The Delete User Profile command, `DLTUSRPRF`, deletes an OS/400 user profile. This command will automatically delete the user's message queue (provided the user is the owner of his or her own message queue) and remove the user from the system directory and distribution lists. However, there are several situations that will prevent the `DLTUSRPRF` command from successfully deleting the user profile including:

- The user profile owns objects, but the `OWNOBJOPT` parameter specifies `*NODLT`.
- The user profile is the primary group for any objects, but the `PGPOP` parameter specifies `*NOCHG`.
- The user profile is enrolled in OfficeVision/400.
- The user profile is a group profile that contains members.

Because the default value for the OWNBJOPT parameter is *NODLT, the Tivoli default value for the OS400_DLT_Action is also *NODLT. This is a significant point to consider. This means that, if this default value is not changed and the user profile being deleted owns OS/400 objects, the `DLTUSRPRF` command issued by Tivoli will fail to delete the user profile from the AS/400 endpoint even though the user record has been deleted from the Tivoli user profile. As a result, the Tivoli user profile and the AS/400 Endpoint will be *out of sync* with each other. We will discuss this *out of sync* condition further in the next section.

Profile synchronization

In the preceding section, we discussed several instances where it is possible to create a situation in which the Tivoli user profile no longer accurately portrays the user profiles on the AS/400 endpoint. In a general sense, this condition results from one of two occurrences:

- Changes made directly on the AS/400 endpoint that are not visible in Tivoli.
- Changes made to the Tivoli user profile that could not be processed on the AS/400 endpoint. Several specific examples are provided in the preceding discussions of the `CHGUSRPRF` and `DLTUSRPRF` commands.

At this time, there is no function provided to synchronize the AS/400 endpoint and the Tivoli user profile. Nonetheless, while there may be no way to actually prevent all occurrences of these situations, we wish to provide a few suggestion you may wish to consider:

- Secure these OS/400 commands to prevent unauthorized use of them on the AS/400 endpoint.
- If end user access to these functions is desired, create a “Tivoli” menu that allows the users to input their desired options but, rather than running these commands, sends a message to a message queue being monitored by TEC. In this way, the request can be sent to a TEC console and processed by Tivoli administration. Refer to Section 7.3.5.6, “Creating an AS/400 Tivoli Inventory menu” on page 382, for an example of using a Tivoli menu.
- Review the Tivoli User Administration joblog on the AS/400 endpoint for messages, and correct these where appropriate. Locating this joblog will be discussed in more detail in Section 6.2.4, “Troubleshooting Tivoli User Administration for AS/400” on page 176.
- On some regular scheduled basis, write a script that runs `wpopulate` from the AS/400 endpoints into a temporary Tivoli user profile. Then, compare the user record value from this temporary user profile with the actual user

profile containing the user records updating the user record or redistributing the Tivoli user profile as desired to correct any differences. Recall that you can use the -f option to limit this to a particular set of user profiles at a time.

6.2.3 Customizing Tivoli User Administration for AS/400

In this section, we will provide suggestions on ways to supplement and enhance the functions provided by Tivoli User Administration for AS/400 Endpoints. These suggestions include creating an AS/400 User Administration Task Library, editing default and validation policies, and making AEF customizations.

6.2.3.1 An AS/400 User Administration task library

As part of user administration on the AS/400, AS/400 system administrators will often need to perform additional tasks when creating or deleting a user. Consider the following four examples:

- If the AS/400 is part of an SNA Network, they may wish to enroll the new user in the AS/400 System Directory using the Add Directory Entry (ADDDIRE) command.
- As discussed previously, a very important consideration before deleting a User Profile from the AS/400 is object ownership. Refer back to the discussion on the DLTUSRPRF command. In some cases, it may be fine to specify to delete the objects a User Profile owns or to assign ownership of these objects to another user. But, in many cases, AS/400 system administrators want to view a listing of OS/400 objects that the user owns before selecting one of these options. This can be accomplished by using the Display User Profile (DSPUSRPRF) command and specifying *OBJOWN in the TYPE parameter.
- Should it be determined that all the objects that a User Profile to be deleted owns can be deleted except for one or two, it is often preferable to change ownership of just those particular objects and then delete the User Profile specifying to delete the remainder of the objects. Changing the ownership of a specific OS/400 object can be done using the Change Object Owner (CHGOBJOWN) command.
- In some cases, you may wish to remove an OS/400 user profile from the AS/400 System Directory without deleting the user profile. Recall that, when actually deleting a user profile, the Delete User Profile (DLTUSRPRF) command will take care of removing the user from the AS/400 System Directory. However, the Remove Directory Entry command, RMVDIRE, can be used to remove the user's directory entry without removing the user profile from the system.

Tasks, such as these as well as other unique AS/400 functions, can be added as Tivoli Task. Refer to Chapter 5, “AS/400 Endpoint considerations and tips” on page 99, for a complete discussion of the supplied AS/400 Task Library and creating AS/400 Tasks.

In Section 6.2.5, “AS/400 User Administration management scenarios” on page 181, we address these four specific examples by creating and using the “AS/400 User Administration Task Library” shown in Figure 72.

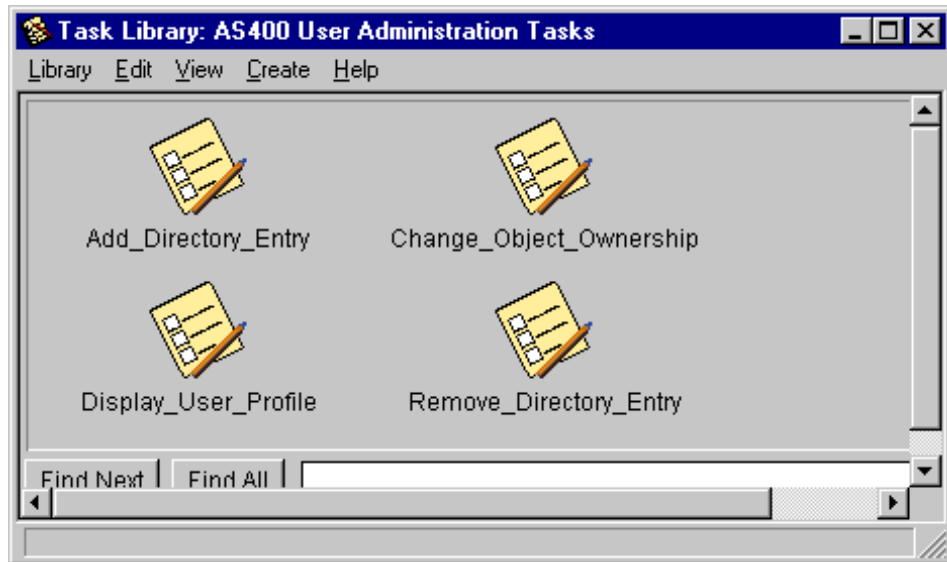


Figure 72. AS/400 User Administration task library

For your reference, Appendix B, “AS/400 User Administration task library definition file” on page 511, contains the UserAdmin.tll file that was used to create this. Note that this file is intended as a sample only and, as such, does not provide several features that we suggest adding should you wish to use it in a production environment:

- Not all CL command parameters are necessarily provided. Refer to the *OS/400 CL Reference V3R2*, SC41-3722.
- A complete HelpDescription field should be added.
- No parameter validity checking is performed.
- For use in a multinational environment, NLS enablement should be done using gencmsg.
- Finally, it is worth mentioning that the Change Object Ownership does not handle changing ownership of IFS objects. Also, the Display User Profile

command used with the *OBJOWN parameter will not show any IFS objects the user profile owns. However, it will issue message CPI220D indicating that “*n* objects were not included in this list.” in this case. You may wish to write Tasks to handle IFS objects using OS/400 system APIs.

6.2.3.2 Default and validation policy suggestions

Initially, a Tivoli user profile contains no records, but it does contain a set of Tivoli-provided default and validation policies. It is important to note that Tivoli User Administration uses profile-based policy rather than region-based policy. This means that each user profile has its own unique set of default and validation policies. While this provides an increased level of flexibility, it also adds increased complexity. Therefore, emphasis is again placed on the importance of planning and design, especially in large environments. Refer to the *Tivoli User Administration Design Guide*, SG24-5108, and the specific sections in the *User Administration 3.6.2 User's Guide*, GC32-0291, dealing with managing many users.

Clearly, the default and validation policies provided by Tivoli User Administration for AS/400 Endpoints are designed to correspond to the OS/400 command defaults and allowable values. Although much of the default and validation policy will be environment-specific, a few suggestions might include the following.

Default policies

In rare cases, an AS/400 Administrator may wish to change the defaults as they are shipped from IBM. They can accomplish this using the Change Command Defaults command, `CHGCMDDFT`. If this is done, clearly, you will want to make corresponding changes to the default policies for any altered parameter.

Another useful tool that an AS/400 Administrator may be accustomed to using is the ability to copy an existing OS/400 user profile when creating a new one. This function is very similar to creating a user profile template. Chapter 9, “AS/400 management scenarios” on page 437, provides a scenario of this. However, because the policy is profile-based, there may be environments in which editing the default policies makes sense. Perhaps, for example, an entire department wants to have the same initial menu, initial program, and printer device values. You may select to switch the default policy so that these values will automatically be generated for a new user record within this department's profile.

Validation policies

Just as it is possible for an AS/400 Administrator to change the IBM-supplied default values on the commands that Tivoli User Administration will use, they

may also choose to force a particular value into a parameter. This is generally achieved by writing a command very similar to the OS/400-supplied one but not providing prompts for values that they do not want changed. The end user will then call this customized version of the command and fill in the parameters that they desire. When this customized command runs, what really occurs is that the actual OS/400 command is called specifying both the user-specified options and the hardcode values. In cases such as this, it would make sense to change the validation policy so that only the values that were hardcoded are allowed.

Recall also that neither the Create User Profile command, `CRTUSRPRF`, nor the Change User Profile command, `CHGUSRPRF`, validated the AS/400 password rules. However, this limitation could actually be somewhat compensated for by implementing a validation policy that would functionally match the setting of the following AS/400 system values:

QPWDEXPITV	Password expiration interval
QPWDLMTAJC	Limit adjacent digits in password
QPWDLMTCHR	Limit characters in password
QPWDLMTREP	Limit repeating characters in password
QPWDMAXLEN	Maximum password length
QPWDMINLEN	Minimum password length
QPWDPOSDIF	Limit password character positions
QPWDRQDDGT	Require digit in password
QPWDRQDDIF	Duplicate password control
QPWDVLDPGM	Password validation program

Also, any preferred naming conventions for user profiles, printers, job descriptions, and so on can be enforced by updating the appropriate validation policy.

6.2.3.3 AEF customizations

Tivoli AEF is a product, which allows you to customize Tivoli applications to conform to specific environmental needs. We will not attempt to cover AEF in any detail here other than to point out its ability to customize and extend the capabilities of Tivoli products. Refer to the *TME 10 AEF 3.6 User's Guide*, GC31-8345, for more details.

6.2.4 Troubleshooting Tivoli User Administration for AS/400

Troubleshooting Tivoli User Administration for AS/400 Endpoints can be broken down into two main areas:

- Problems with the Profile Distribution
- Unexpected results of the Profile Distribution

Troubleshooting Tivoli user profile distribution problems will be very similar to troubleshooting distribution problems to other endpoint types. You will first want to verify that the endpoint is available and properly connected to the gateway and that the gateway is functioning properly. Refer to *All About Tivoli Management Agents*, SG24-5134, for further details about these activities related specifically to an AS/400 Endpoint.

Troubleshooting unexpected results of a Tivoli user profile distribution should begin with an examination of the Distribution Defaults. The redbook, *Tivoli Enterprise Internals and Problem Determination*, SG24-2034, details these Distribution Defaults as well as specific Tivoli User Administration troubleshooting techniques to follow when you are experience unexpected results.

Finally, as we mentioned several times previously, the Tivoli user administration joblog that is actually attempting to create, change, and delete the user profiles on the AS/400 endpoint should be checked. Tivoli user administration jobs will run as the QTIVROOT on the AS/400 and have a Job name of UMBO_SKEL1. These jobs are run using the Tivoli-provided job description, QTMELCF, in the QTMELCF library; so, one way to locate these joblogs after the job has run is to issue the following Work with Spoolfile command, WRKSPLF:

```
WRKSPLF SELECT(QTIVROOT *ALL *ALL UMBO_SKEL1)
```

This will bring up all the QTIVROOT's spoolfiles, which specify UMBO_SKEL1 as the User Data. More than likely, these will all be spoolfiles called QPJOBLOG indicating that these are the joblog's spoolfiles created when the job ended. These are shown in Figure 73 on page 177.

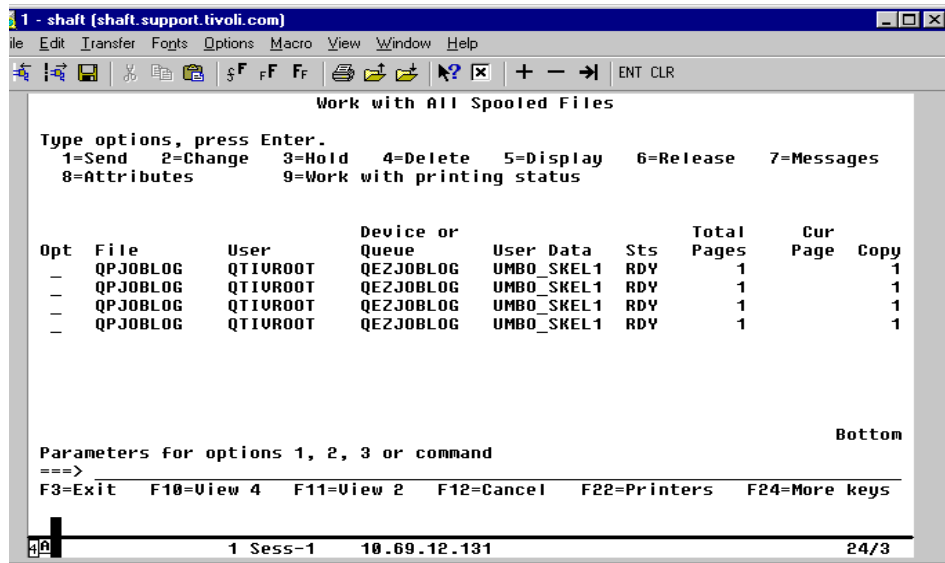


Figure 73. Work with spooled files

Note

The following Function keys on the *Work with Spooled Files* screen are very useful, particularly if there are a large number of spooled files.

- F18** Bottom will take you to the last screen of spooled files. As the spooled files general are listed from first to last, the last spoolfile is usually the one from the last running of the job.
- F11** View 1/2/3/4 will display other spoolfile attributes. We find View 2 to be particularly helpful as it provides the spoolfile creation date and time and this can often be helpful to confirm that the desired joblog is being examined.

Alternatively, you may wish to look at all QTIVROOT's spoolfiles using:

```
WRKSPLF SELECT(QTIVROOT *ALL *ALL *ALL)
```

Additionally, since the Tivoli User Administration jobs are using the QTMELCF job description, these jobs will be submitted to the QSYSNOMAX job queue, which, by default, will be allocated to the QSYSWRK subsystem. You can use the Work with Job Queues command, `WRKJOBQ`, to look at the job queues to verify that the job queue is released and allocated to the QSYSWRK subsystem as expected and that no jobs are waiting or held.

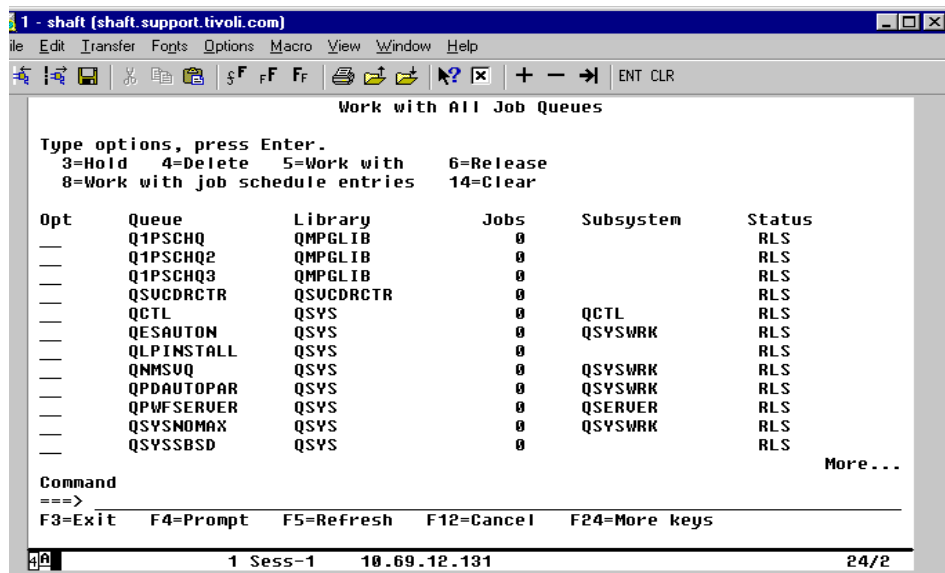


Figure 74. Work with job queues

Notice, in Figure 74, that the QSYSNOMAX job queue is released, allocated to the QSYSWRK subsystem, and has no jobs waiting to be run. If any of these were not true, the Tivoli User Administration jobs (along with any other jobs using this job queue) would not run and the Jobs column would display the number of jobs waiting.

This screen can also help determine if the Tivoli User Profile job is actually getting submitted. By specifying option 3 (Hold) for the QSYSNOMAX job queue and distributing a user profile, you should be able to catch the job on the job queue by using F5 (Refresh) on this screen. When the job is submitted, the number of jobs on the QSYSNOMAX job queue should switch from 0 to 1. After this occurs, you can choose Option 5 (Work with) to see the job. The main identifying characteristic should be the UMBO_SKEL1 job name. However, you will be able to see all the job characteristics available before it runs including its qualified job name consisting of the Job, User, and Number.

Once the fact that the job is reaching the job queue has been confirmed, you will want to release the job queue with Option 6 (Release) so that the job will be started in the QSYSWRK subsystem. Also, note that holding the job queue for a long time is not recommended because other applications may also be using this job queue to run their jobs; so, be sure to keep a close watch on the

Once the job is submitted, you may be able to see the job running by using the Work with Active Job command, `WRKACTJOB`, and looking at the active jobs in the QSYSWRK subsystem. However, in most cases, this job runs quite quickly and may complete before you can see it. Also, note that, if you obtaining the fully-qualified job name as mentioned previously, you can use the Work with Jobs command, `WRKJOB`, to see the job's status and joblog on the resulting menu:

As an example, let us examine the following UMBO_SKE1 joblog. Figure 75 shows page 1 of the UMBO_SKE1 joblog.

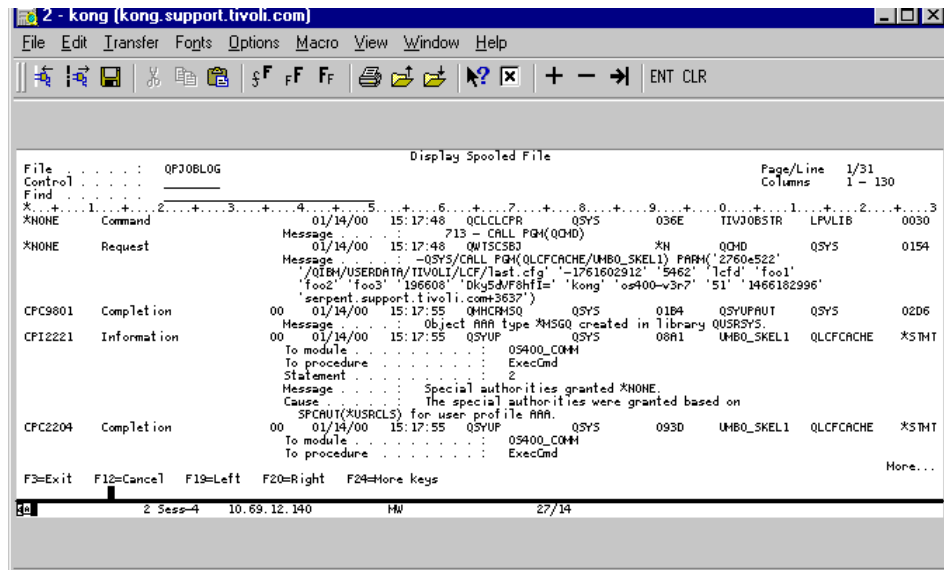


Figure 76 on page 180 shows page 2 of the UMBO_SKE1 joblog.

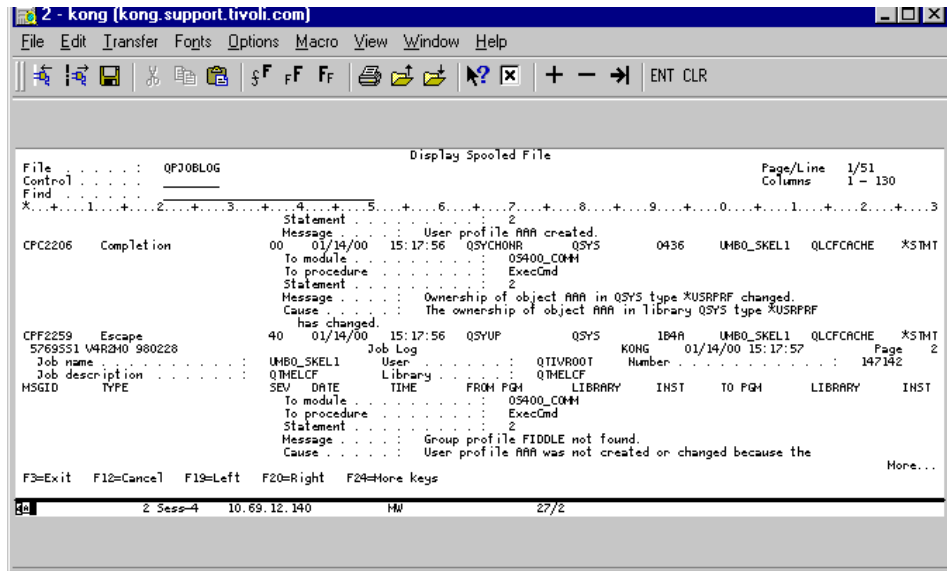


Figure 76. UMBO_SKE1 joblog (2 of 4)

Figure 77 shows page 3 of the UMBO_SKE1 joblog.

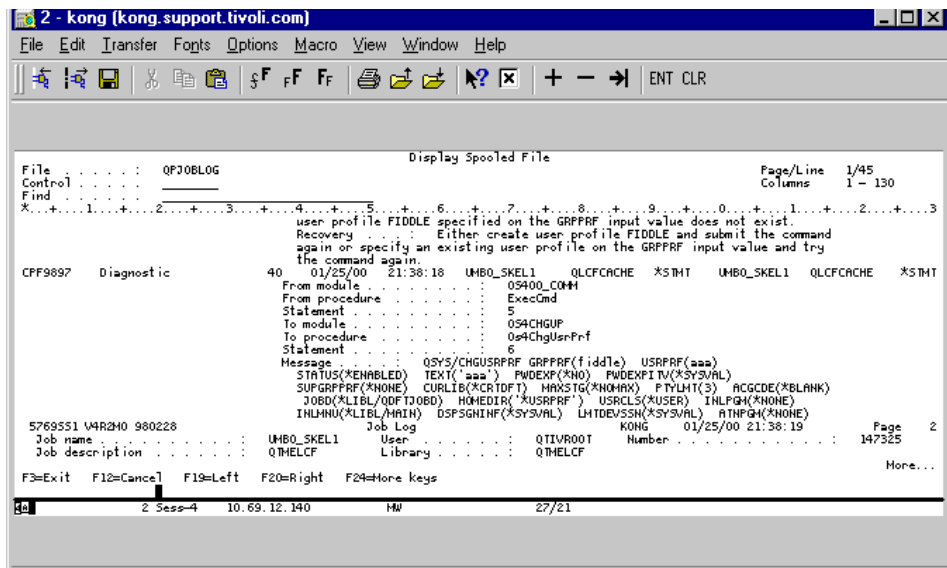


Figure 77. UMBO_SKE1 joblog (3 of 4)

Figure 78 on page 181 shows page 4 of the UMBO_SKE1 joblog.

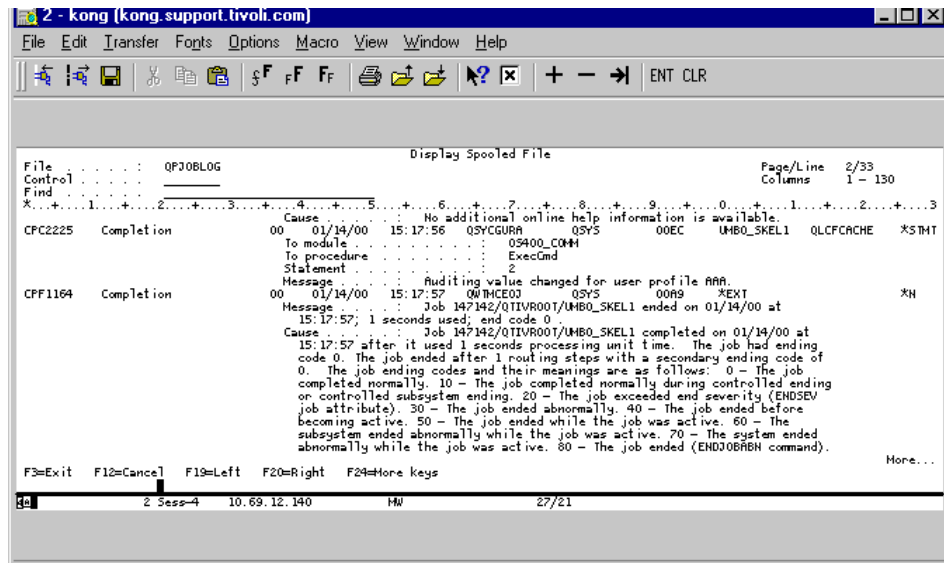


Figure 78. UMBO_SKEL1 joblog (4 of 4)

This joblog reveals that this was an attempt to create a new user profile, AAA, on the AS/400. While the Create User Profile command was successful, the Change User Profile command failed because it attempted to make the user profile, AAA, a member of the FIDDLE group profile when, in fact, there is no user profile called FIDDLE on the system.

Since we have stressed the importance of reviewing the joblogs for this process, at this point, we want to mention that you may want to consider ways of retrieving OS/400 joblogs from Tivoli. We will illustrate a possible way of achieving this for Tivoli jobs in Chapter 7, “AS/400 availability management” on page 217.

6.2.5 AS/400 User Administration management scenarios

In this section, we will show specific examples of using Tivoli User Administration for the AS/400. To facilitate this, we will consider an imaginary company consisting of two departments, Engineering and Finance, with three AS/400s: ISHII1, ISHII2, and SHAFT. ISHII1 and ISHII2 are used by the Engineering Department. Every member of the Engineering Department has identical AS/400 user profiles on both machines. SHAFT is used by the Finance Department. Additionally, there is a small set of people who provide services for both of these departments and, thus, have AS/400 user profiles on all three machines. For discussion purposes, we call these global users.

Since the importance of planning and design have been stressed several times previously, we will only briefly point out a few important guidelines to remember:

- Every subscriber in a Profile Manager must be able to accept distribution of all the profiles contained within that Profile Manager.
- Never have an Endpoint subscribed to more than one Dataless Profile Manager since the last profile distributed for a Profile Manager may overwrite changes from a prior distribution from the other Profile Manager.
- There is a recommended limit of 500 and a practical limit of 2,000 for sorting per profile. This limit is often handled by dividing the users into separate profile managers based on their user names.

Note

AS/400 user profiles may begin with the special characters, such as @, #, and \$. You will want to allow for user profiles, such as these, to be contained within the Tivoli user profiles.

For our sample environment, the implementation shown in Figure 79 on page 183 was used.

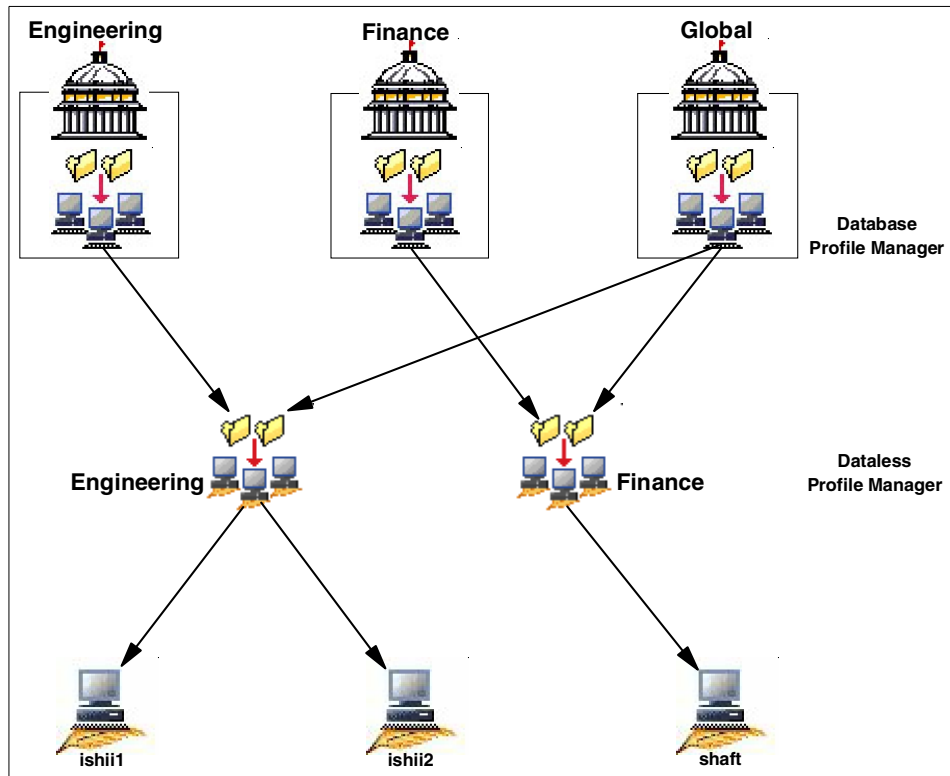


Figure 79. Profile distribution flow

Note

Notice that for global users, local changes can be in the Engineering and Finance Dataless Profile Managers. This would allow their AS/400 User Profiles to be different on SHAFT versus ISHII1 and ISHII2. Notice also that the AS/400 User Profiles on ISHII1 and ISHII2 will be identical.

To create this environment, the following steps were performed:

1. Create the Engineering, Finance, and Global profile managers within the Engineering, Finance, and Global policy regions. Recall, as mentioned previously, that User Administration provides profile-based policy. In our environment, we used separate policy regions simply to mimic the situation where separate Tivoli administrators may control these departments and, therefore, only access their appropriate policy region.
2. Create the A-F, G-L, M-R, and S-Z profiles within each profile manager.

Note

In a large environment, it is advisable to consider creating a single *master* profile defining any appropriate default and validation policies. Then, once completed, clone this master profile to create the *subset* (A-F, G-L, M-R, and S-Z) profiles. Finally, once all the profiles are created, add the specific validation policies for each so that only users that fall into the appropriate ranges are allowed.

3. Run the `w4getusr.pl` script on the AS/400 Endpoints to create a list of the User Profiles on each system. Complete details of this script will be given in the first example scenario.
4. Issue the `wpopusers` command using the `-f` option to specify the file from which to obtain the user's records and populate each profile appropriately.
5. Create the Engineering, Finance, and Global Dataless Profile Managers within the ITSO-region policy region.
6. Subscribe the appropriate AS/400 Endpoint(s) to each Dataless Profile Manager. In our environment:
 - ISHII1 and ISHII2 are subscribed to the Engineering Dataless Profile Manger
 - SHAFT is subscribed to the Finance Dataless Profile Manager
7. Subscribe the appropriate Dataless Profile Manager(s) to each Profile Manager. In our environment:
 - The Engineering Dataless Profile Manager is subscribed to the Engineering Profile Manager.
 - The Finance Dataless Profile Manager is subscribed to the Finance Profile Manager.
 - Both the Engineering Dataless Profile Manager and the Finance Dataless Profile Manager are subscribed to the Global Profile Manager.
8. Set the distribution defaults for each profile.

6.2.5.1 Populating a Tivoli User Profile

In our first scenario, we will show a scenario of populating a Tivoli User Profile using the `w4getusr.pl` script and the `wpopusers` command. Note that the User Administration portion of Chapter 9, "AS/400 management scenarios" on page 437, populates the Tivoli User Profile using the `Populate...` option from the Profile menu; so, in this scenario, we will use a three step approach to this using the `w4getusr.pl` script to create lists of the

AS/400 User Profiles, manipulating these files to meet our requirements, and then running the `wpopusr` command with the `-f` option to populate the profiles using these files.

Before beginning, recall the following goals for populating our profiles:

- The Global User Profile Manager should contain only those AS/400 User Profiles that exist on *all* three AS/400s: ISHII1, ISHII2, and SHAFT.
- The Engineering User Profile Manager should contain the AS/400 User Profiles that exist on ISHII1 and ISHII2 but not on SHAFT.
- The Finance User Profile Manager should contain AS/400 User Profiles that exist on SHAFT but not on ISHII1 and ISHII2.
- Within each of the User Profile Managers, we want to group the AS/400 User Profiles into Tivoli User Profiles based on the following ranges:
 - A - F
 - G - L
 - M - R
 - S - Z

To accomplish this, we perform the following steps:

1. Run the `w4getusr.pl` scripts on ISHII1, ISHII2, and SHAFT to produce files containing a list of the AS/400 user profiles on each.

The `w4getusr.pl` script will return the output from the AS/400 Display User Profile (`DSPUSRPRF`) command. Its options are:

```
# ./w4getusr.pl
Usage: ./w4getusr.pl <option> host-name
-d <dir>      Destination directory where to write output files
-n <Number>   Maximum of number of entries to put in each file.
```

As part of the hostname field, you can include the AS/400 User Profile and password to perform the `DSPUSRPRF` function. For example, you could run:

```
# ./w4getusr.pl -d /tmp -n 25 "host1 qsecofr passwd"
```

The script will also look for an entry for the endpoint host in the `$home/.netrc` file. If an entry for this machine exists, it will attempt to run as the user profile and password specified in this file. Otherwise, it will prompt for the user profile and password.

Note that the `-n` option can be used, as in the above example, to produce an ordered list of files in the destination directory (`usr0`, `usr1`, `usr2`, and so on). These files can then be used with the `wpopusrs` command to populate a user profile. However, in our environment, we will not use this `-n` option since we want to divide the user profiles by our specific divisions.

Therefore, in our scenario, we run:

```
# ./w4getusr.pl -d /tmp/shaft shaft
We need a user account, please type or just hit enter to use [default]
login for shaft ([]): qsecofr
Password for account qsecofr:
>
      81      81      652 /tmp/shaft/usr0
```

Likewise, we run this also on ISHII1 and ISHII2. When we are finished, the following files exist:

- /tmp/shaft/usr0
 - /tmp/ishii1/usr0
 - /tmp/ishii2/usr0
2. Now, in the /tmp/shaft directory, we first divide the `usr0` file into the following files based on the OS/400 user profile name:
 - finance.a-f
 - finance.g-l
 - finance.m-r
 - finance.s-z
 3. Note that the `usr0` files from ISHII1 and ISHII2 should be identical in our environment. If differences are noted, they should be corrected. Assuming they are what was expected, we use the `usr0` files from ISHII1 to produce the following files:
 - engineering.a-f
 - engineering.g-l
 - engineering.m-r
 - engineering.s-z
 4. Now, we must compare the following files looking for AS/400 User Profiles, which exist in both:
 - finance.a-f and engineering.a-f
 - finance.g-l and engineering.g-l

- finance.m-r and engineering.m-r
- finance.s-z and engineering.s-z

Those user profiles that do exist in both are removed from the above files and placed in the following files as appropriate:

- global.a-f
- global.g-l
- global.m-r
- global.s-z

This is quite simple to do manually using cut and paste functions in our small test environment. But, clearly, in a large customer environment, a program should be written to automate this task as well as to provide the ability to run the `w4getusrs.pl` script on all the desired AS/400 endpoints. Validation scripts should also be used to maintain correctness within the profiles.

5. Now that our files are created, we use `wpopusrs` to populate the profiles with the appropriate files. For example, the `wpopusrs` for the finance user profiles are:

- `wpopusrs -f finance.a-f @Endpoint:shaft @UserProfile:"Finance User Profile [A-F]"`
- `wpopusrs -f finance.g-l @Endpoint:shaft @UserProfile:"Finance User Profile [G-L]"`
- `wpopusrs -f finance.m-r @Endpoint:shaft @UserProfile:"Finance User Profile [M-R]"`
- `wpopusrs -f finance.a-f @Endpoint:shaft @UserProfile:"Finance User Profile [S-Z]"`

The `wpopusrs` commands for finance and global are straightforward revision of these.

Once the `wpopusrs` command has completed, the Tivoli User Profiles will be populated with the appropriate AS/400 User Profiles. The Finance User Profile [A-F], for example, appears as shown in Figure 80 on page 188.

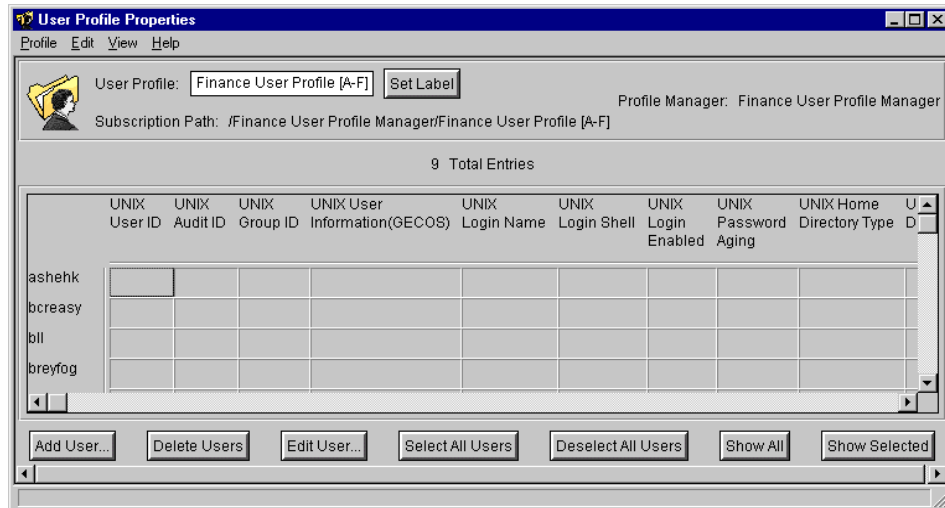


Figure 80. AS/400 User profile properties

Moving the scroll bar along the bottom to the right shows the user record values for all attributes.

6.2.5.2 Creating a new AS/400 User Profile

In this scenario, we will use Tivoli User Administration for AS/400 Endpoints to create a new AS/400 user profile for Beth Davis, a new employee in the Finance Department. Her user profile should have the following values:

User Profile	BDAVIS
Text	Beth Davis
User Class	User
Special Authority	Spool Control
Initial Program	FINANCE/FINPGM
Initial Menu	FINANCE/FINMENU

Beth also needs to be enrolled in the AS/400 System Directory.

Note

Since most employees in the Finance Department will have this same initial program and initial menu, it may be beneficial to create a User Profile Template with these values. Chapter 9, "AS/400 management scenarios" on page 437, shows an example of creating a User Profile template.

We begin by adding the user record for BDAVIS to the “Finance User Profile [A-F]” Tivoli user profile and setting the OS/400 Attributes as illustrated in the following screens. Figure 81 shows the OS/400 login attributes on the User Properties screen.

The screenshot shows the 'User Properties' dialog box with the 'OS/400 Login' panel selected. The 'Category' dropdown is set to 'OS/400'. The 'User Name' field contains 'BDAVIS'. The 'Common Login' field contains 'BDAVIS' and the 'Common Password' field contains '*****'. The 'Login' field contains 'BDAVIS'. The 'User ID number' field is empty. The 'Login enablement' dropdown is set to 'Enable user for processing'. The 'Action taken when profile is deleted' dropdown is set to 'Do not delete, if user owns objects'. The 'New owner login' field is empty. The 'Text description' field contains 'Beth Davis'. At the bottom, there are buttons for 'Add & Close', 'Add', 'Generate Defaults', 'Reset', 'Close', and 'Help'.

Figure 81. OS/400 login attributes

On the OS/400 Login panel, we specify:

Login BDAVIS
Text description Beth Davis

Figure 82 on page 190 shows the OS/400 privileges on the User Properties screen.

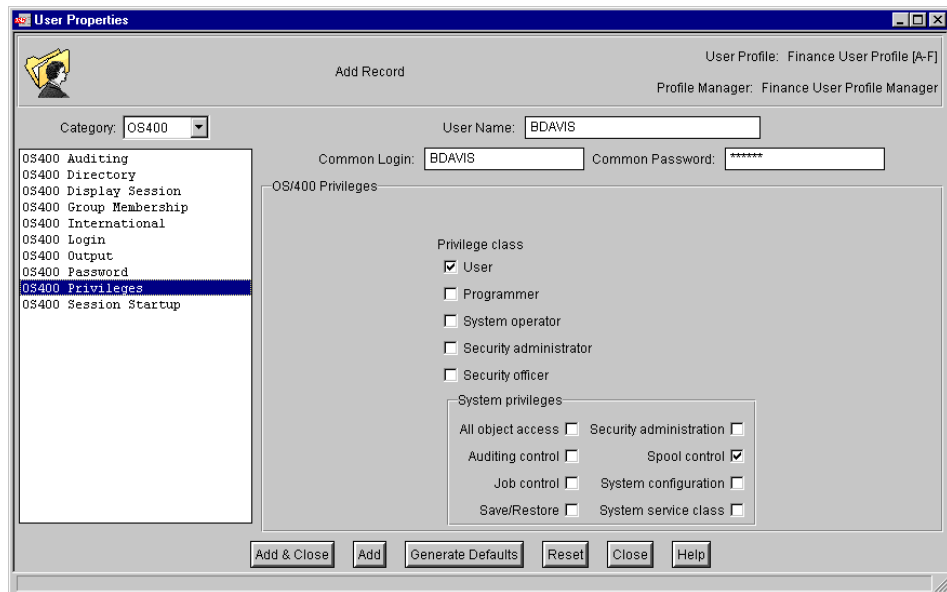


Figure 82. OS/400 privileges

On the OS/400 Privileges panel, we set the following values:

Privilege Class	User
System privileges	Spool Control

Figure 83 on page 191 shows OS/400 session startup on the User Properties screen.

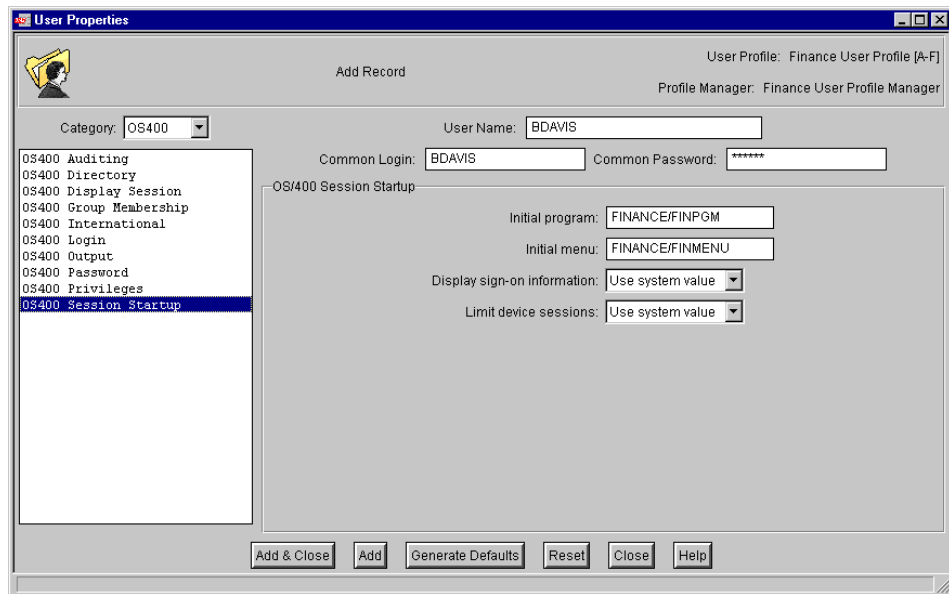


Figure 83. OS/400 session startup

On the OS/400 Session Startup panel, we set the following values:

Initial program FINANCE/FINPGM

Initial menu FINANCE/FINMENU

Once we set these and any other appropriate values, we select **Add & Close** to add the BDAVIS user record.

We then distribute the updated User Profile by dragging and dropping it onto our target subscriber as shown in Figure 84 on page 192.

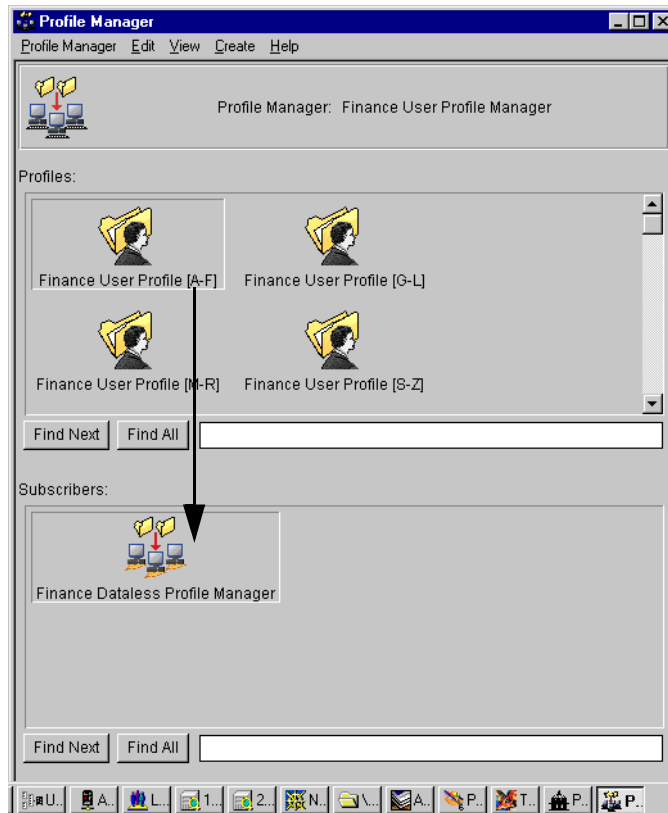


Figure 84. Distributing the user profile

As a result of this profile distribution, we can now issue the following Display User Profile (DSPUSRPRF) command to see the new BDAVIS user profile:

```
DSPUSRPRF USRPRF (BDAVIS)
```

Figure 85 on page 193 shows the displayed user profile information.

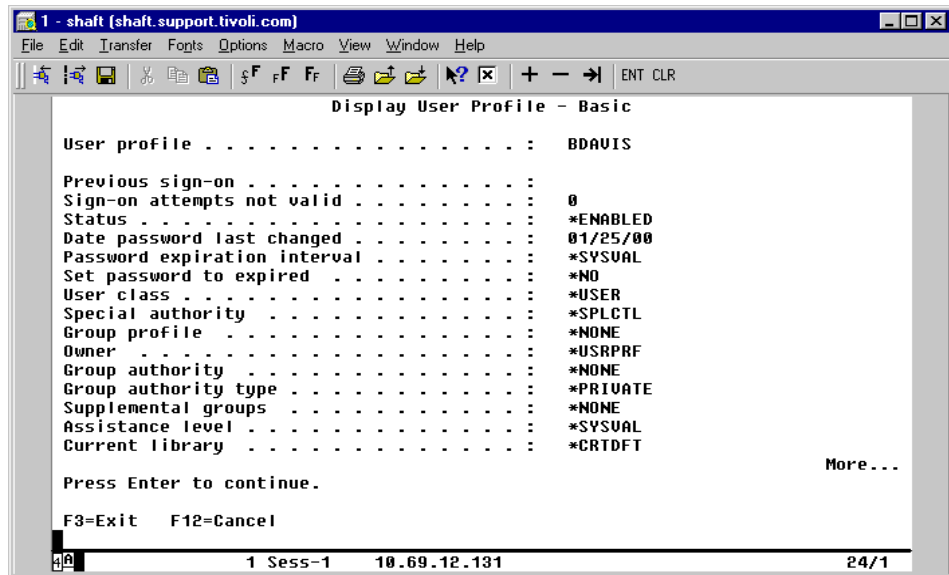


Figure 85. Display user profile (1 of 2)

Figure 86 shows the next page of the screen.

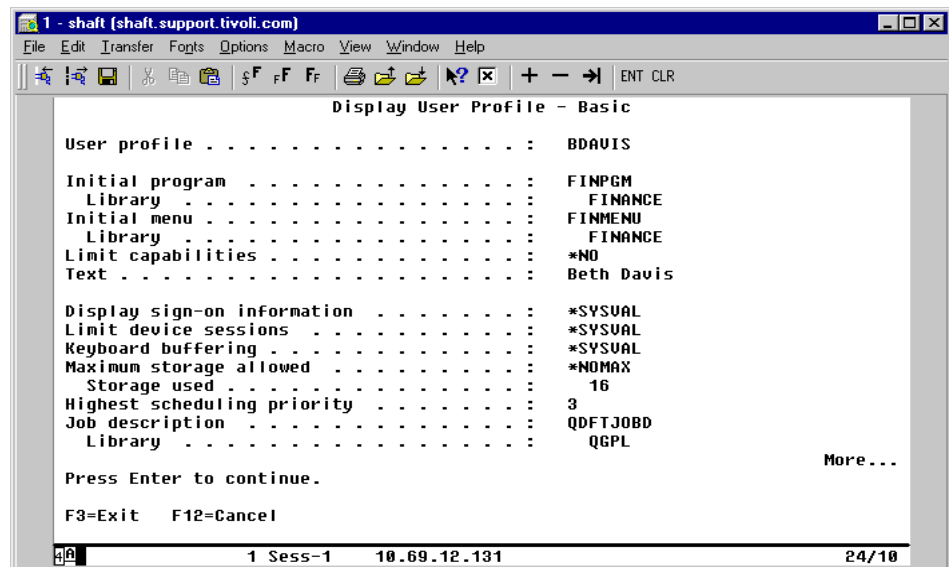


Figure 86. Display user profile (2 of 2)

Notice the following values as set based on our Tivoli user profile:

User Profile	BDAVIS
Text	Beth Davis
User Class	User
Special Authority	Spool Control
Initial Program	FINANCE/FINPGM
Initial Menu	FINANCE/FINMENU

Now that we have created the BDAVIS user profile, we want to add an AS/400 System Distribution Directory Entry for Beth. We will use the `Add_Directory_Entry` task which we created in our AS400 User Administration Tasks task library to do this. First, we open the task library as shown in Figure 87.

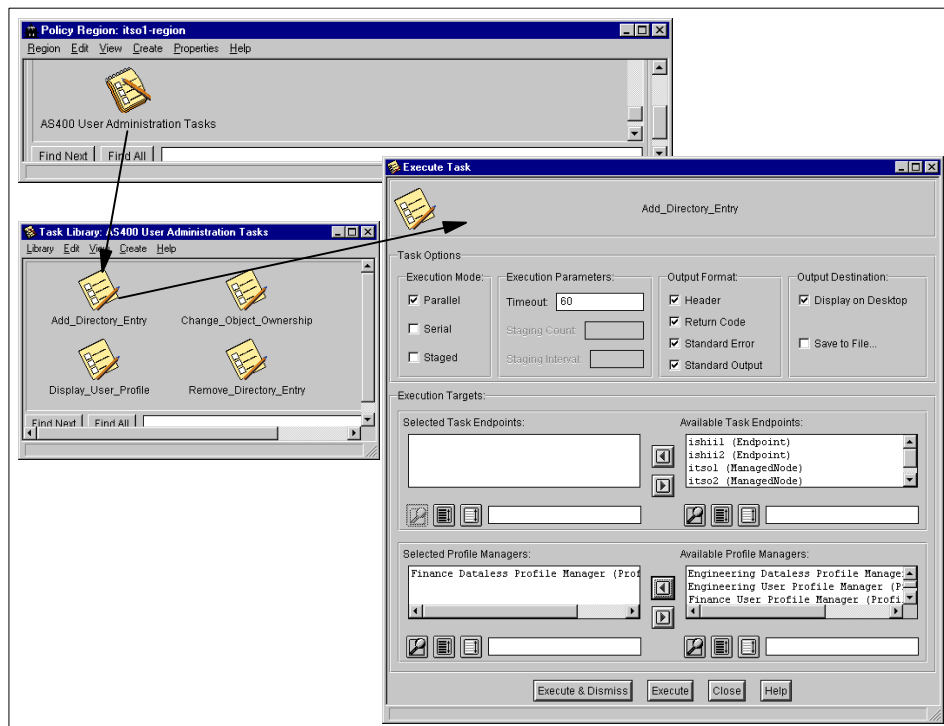


Figure 87. The `Add_Directory_Entry` task execution

Now, we select the `Add_Directory_Entry` task. Complete the Execute Task panel as shown in Figure 87. Once completed, select **Execute & Dismiss**. The panel shown in Figure 88 on page 195 will then be displayed.

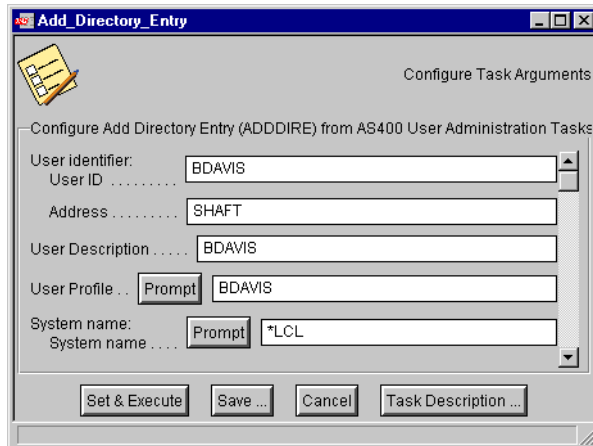


Figure 88. Configuring Add_Directory_Entry task arguments

Provide the desired System Directory Entry values on the Configure Task Arguments panel for the Add_Directory_Entry task as shown in Figure 88.

Press **Set & Execute** to run the task. Since we selected *Display on Desktop* for our output destination, the following output screens will be shown once the tasks complete. Figure 89 on page 196 shows page 1 of the Add_Directory_Entry task output screen.

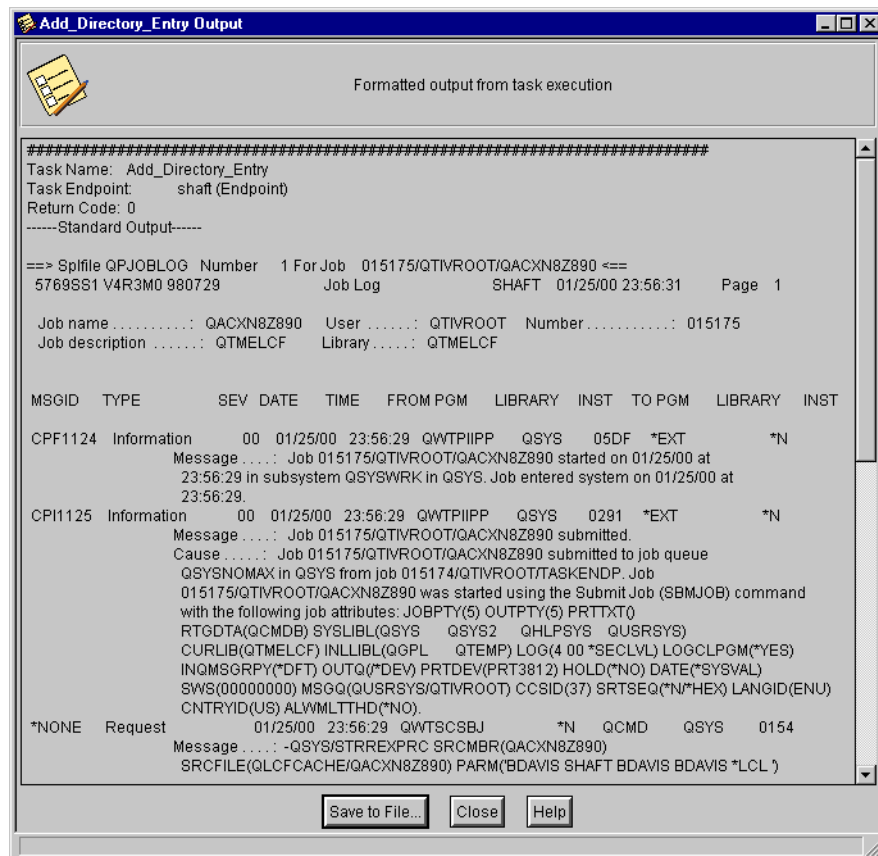


Figure 89. Add_Directory_Entry task output (1 of 2)

Figure 90 on page 197 shows page 2 of the Add_Directory_Entry task output screen.

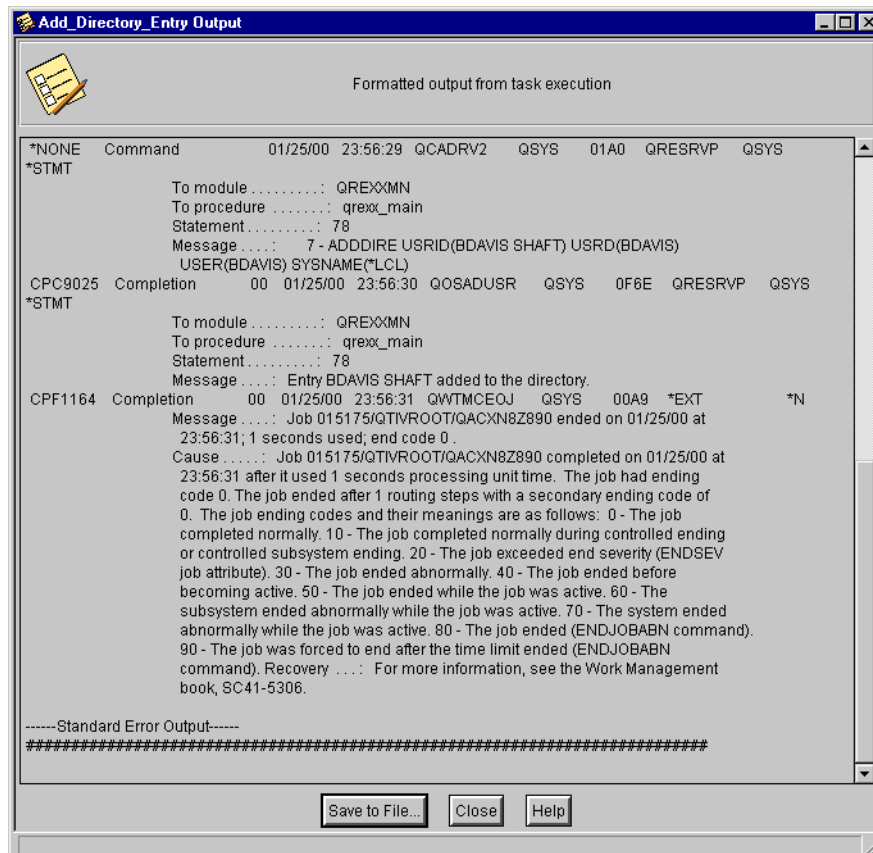


Figure 90. Add_Directory_Entry task output (2 of 2)

Notice that this actually shows the resulting joblog; so, we should check it for any errors. In this case, the CPC9025 message confirms that the entry was successfully added to the directory.

6.2.5.3 Removing an AS/400 directory entry

A common task of AS/400 system administrators is to maintain the AS/400 System Directory. In the preceding scenario, we showed the use of the sample Add_Directory_Entry task. Here, we show the use of the Remove_Directory_Entry task to remove the entry for BLL. In this case, we do not want to delete the BLL user profile. We simply want to remove the unnecessary directory entry.

We begin by selecting our Remove_Directory_Entry task within the AS400 User Administration Tasks library as shown in Figure 91 on page 198.

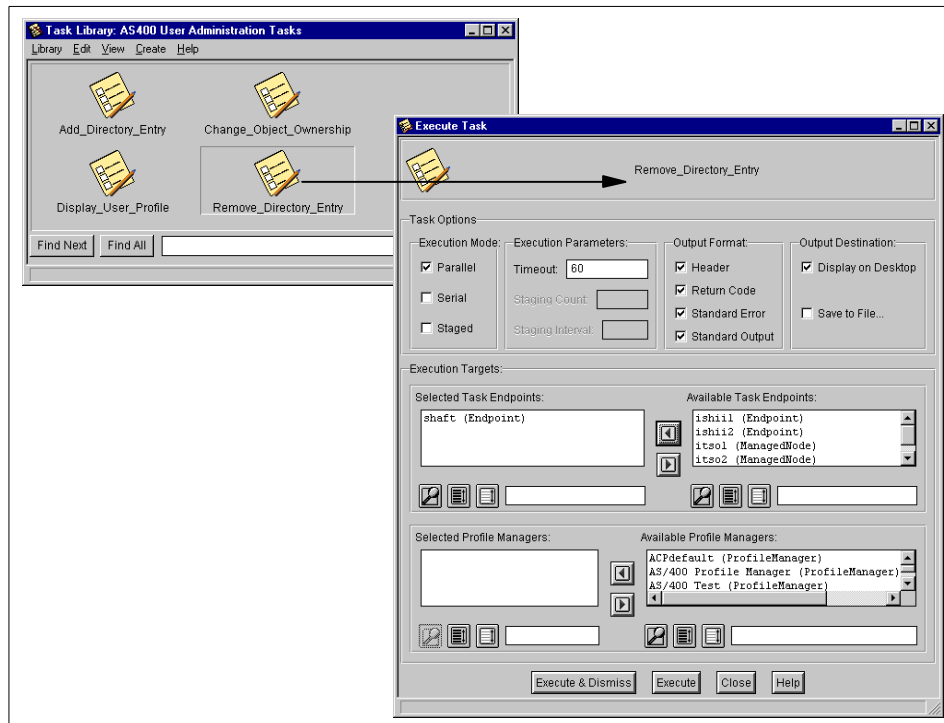


Figure 91. The Remove_Directory_Entry task execution

Complete the Execute Task panel as shown in Figure 91. Once completed, select **Execute & Dismiss**. Figure 92 shows the Configuring Remove_Directory_Entry task arguments screen.

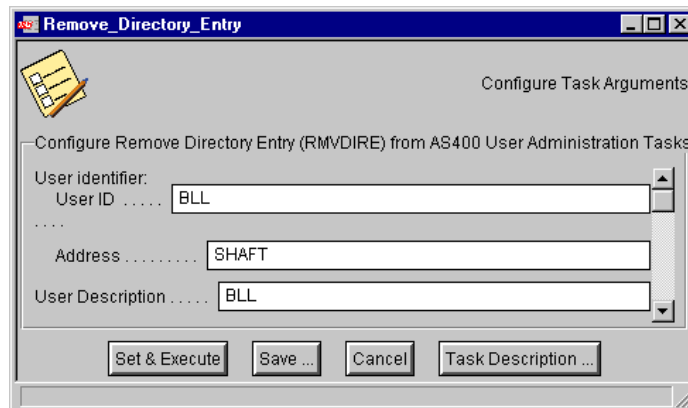


Figure 92. Configuring Remove_Directory_Entry task arguments

Provide the desired Remove Directory Entry values on the Configure Task Arguments panel for the Remove_Directory_Entry task as shown in the preceding screen.

Press **Set & Execute** to run the task. Since we selected Display on Desktop for our output destination, the following output screens will be shown once the tasks complete. Figure 93 shows the Remove_Directory_Entry Output screen.

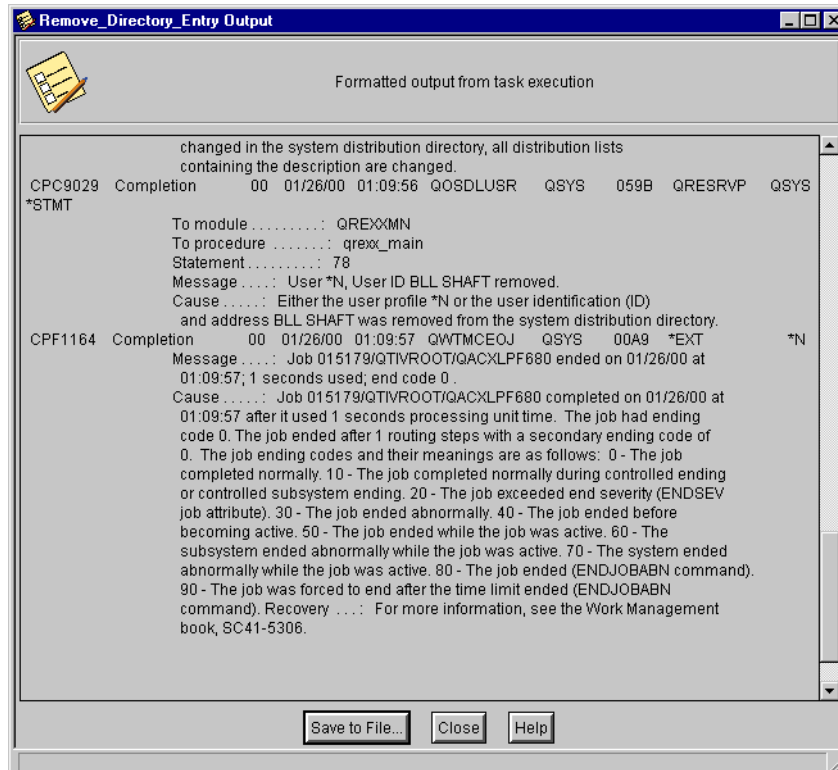


Figure 93. Remove_Directory_Entry task output

The CPC9029 message confirms that the entry was successfully added to the directory.

6.2.5.4 Deleting an AS/400 User Profile

Now, let us suppose that Beth Davis has left the company after several months.

We want to delete her AS/400 user profile. Since she has not been using the system for very long, we suspect that she may not own very many objects. Nonetheless, we want to take a look at the objects she owns before deciding what to specify for the “Action taken when profile is deleted” value. We use the Display_User_Profile Task to see a list of the OS/400 objects that she owns.

We begin by selecting our Display_User_Profile task within the AS400 User Administration Tasks library as shown in Figure 94.

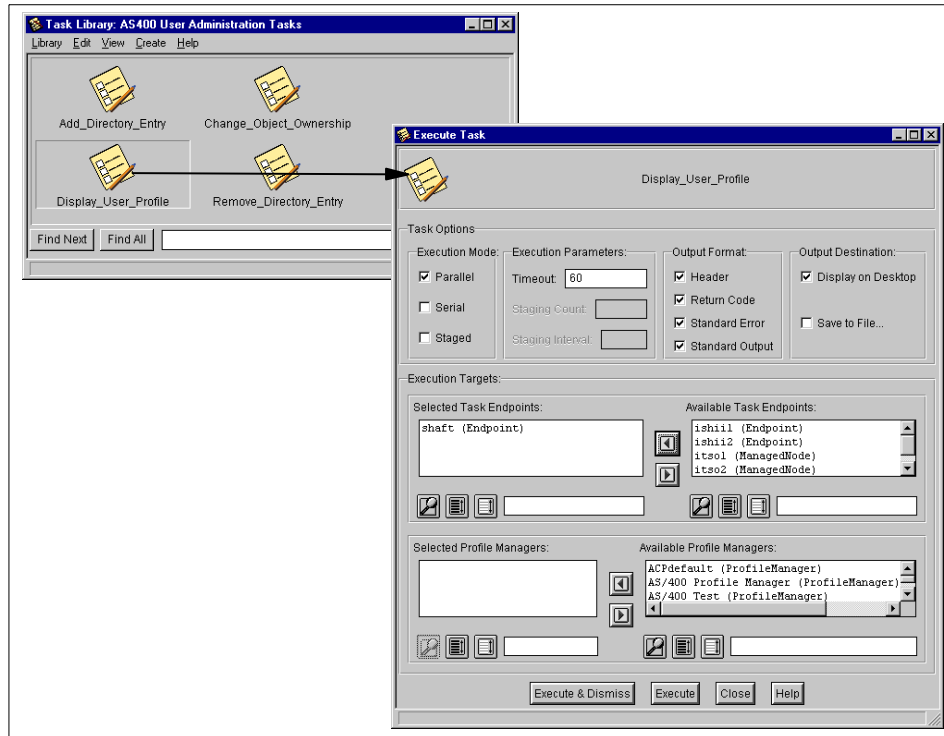


Figure 94. The Display_User_Profile task execution

Complete the Execute Task panel as shown in Figure 94. Once completed, select **Execute & Dismiss**. Figure 95 on page 201 shows the configuration of Display_User_Profile task arguments.

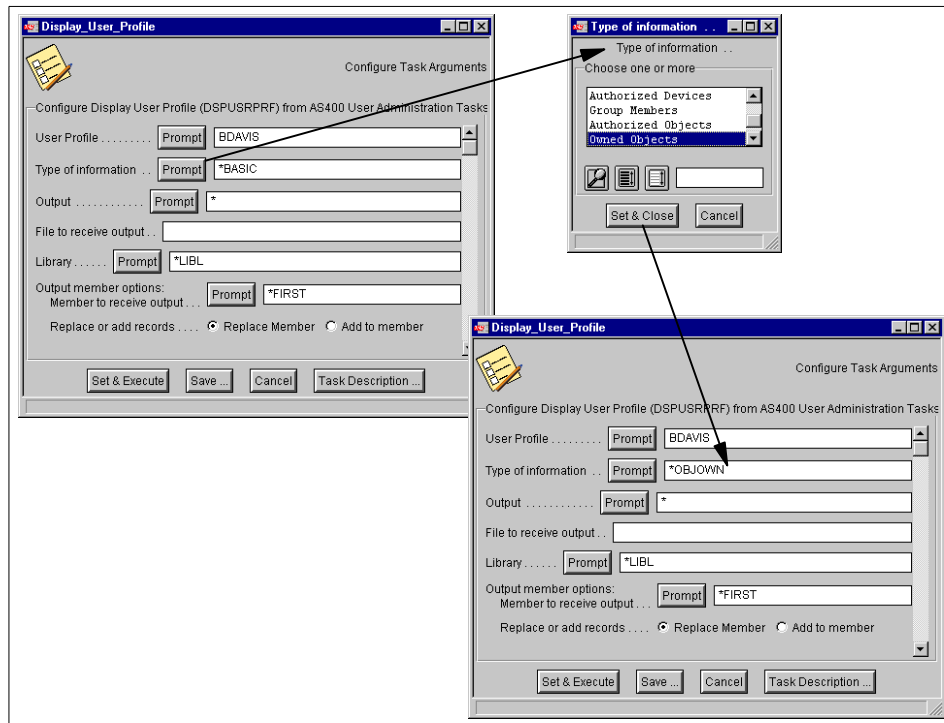


Figure 95. Configuring Display_User_Profile task arguments

Provide the desired Display User Profile command values on the Configure Task Arguments panel for the Display_User_Profile task as shown above. Notice that you can use the Prompt box on the Type of Information field to select Owned Objects from the list.

Press **Set & Execute** to run the task. Since we selected Display on Desktop for our output destination, the following output screens will be shown once the tasks complete. Figure 96 on page 202 shows the Display_User_Profile Output screen.

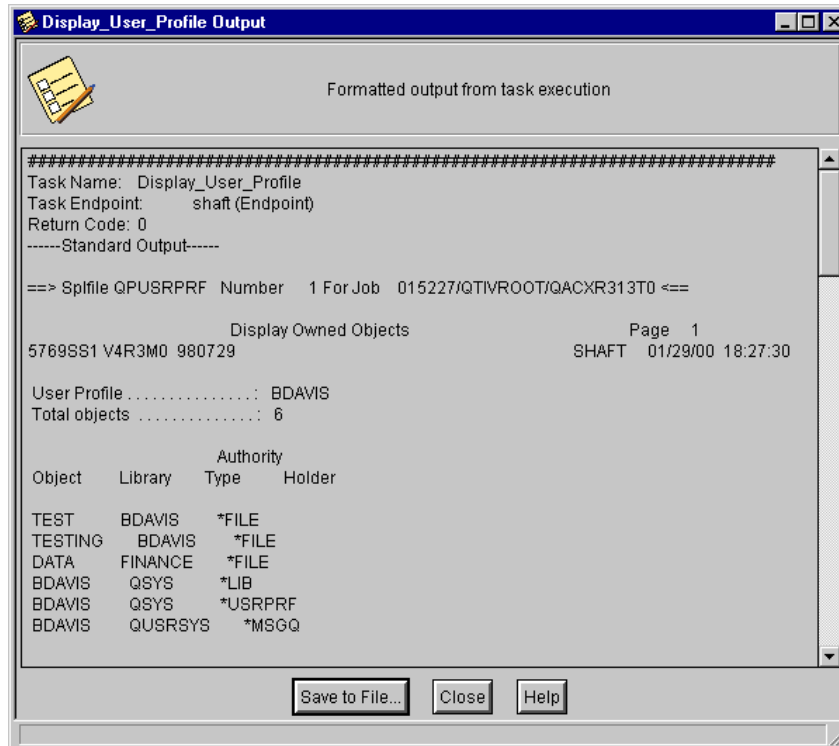


Figure 96. *Display_User_Profile* output

From Figure 96, we notice that BDAVIS owns several objects. The Delete User Profile command will automatically handle the deletion of the message queue. We are content to delete the BDAVIS library, as well as the TEST and TESTING files within it. However, we notice that BDAVIS also owns the DATA file in the FINANCE library. We decide to change the ownership of this object to the FINGRPA group profile. Then, we will delete the BDAVIS user profile and specify to delete the objects she owns.

We begin by selecting our Change_Object_Owner task within the AS400 User Administration Tasks library as shown in Figure 97 on page 203.

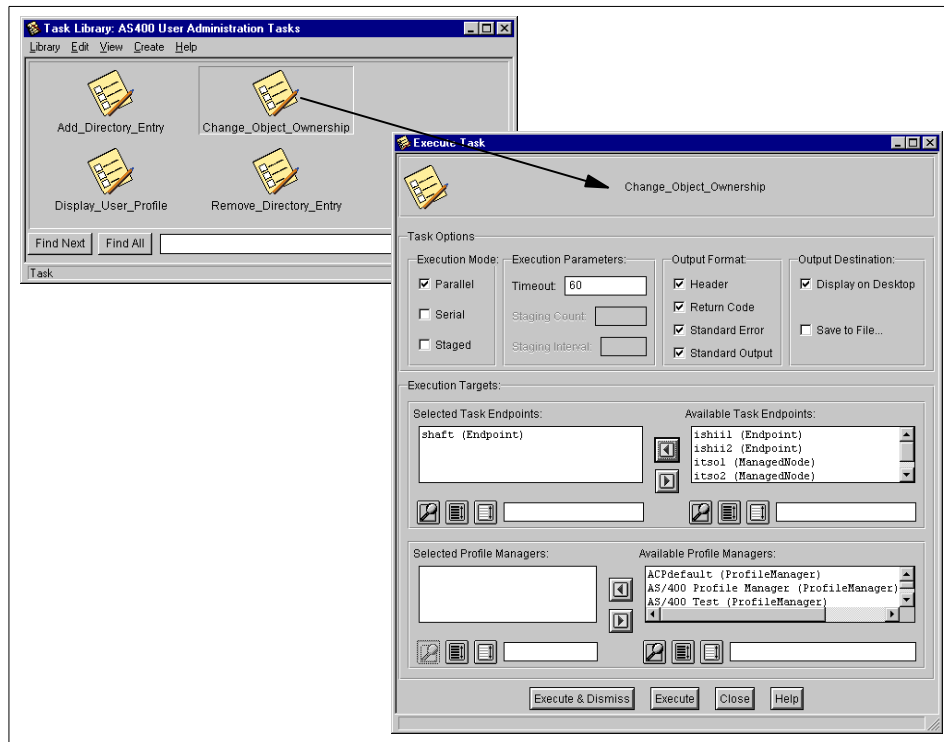


Figure 97. The *Change_Object_Ownership* task execution

Complete the Execute Task panel as shown in Figure 97. Once completed, select **Execute & Dismiss**. Figure 98 on page 204 depicts the configuration of the *Change_Object_Ownership* task arguments.

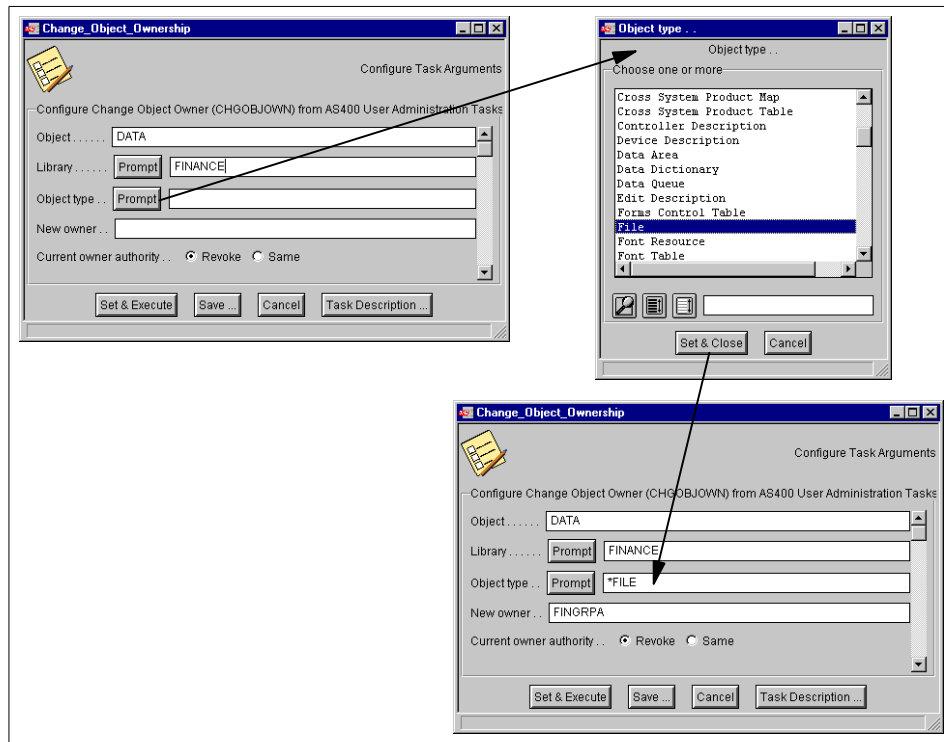


Figure 98. Configuring Change_Object_Ownership task arguments

Provide the desired Change Object Owner command values on the Configure Task Arguments panel for the Change_Object_Owner task as shown in Figure 98. Notice that you can use the Prompt box on the Object Type field to select **Owned Objects** from the list.

Press **Set & Execute** to run the task. Since we selected Display on Desktop for our output destination, the following output screens will be shown once the tasks complete. Figure 99 on page 205 shows the Change_Object_Ownership Output screen.

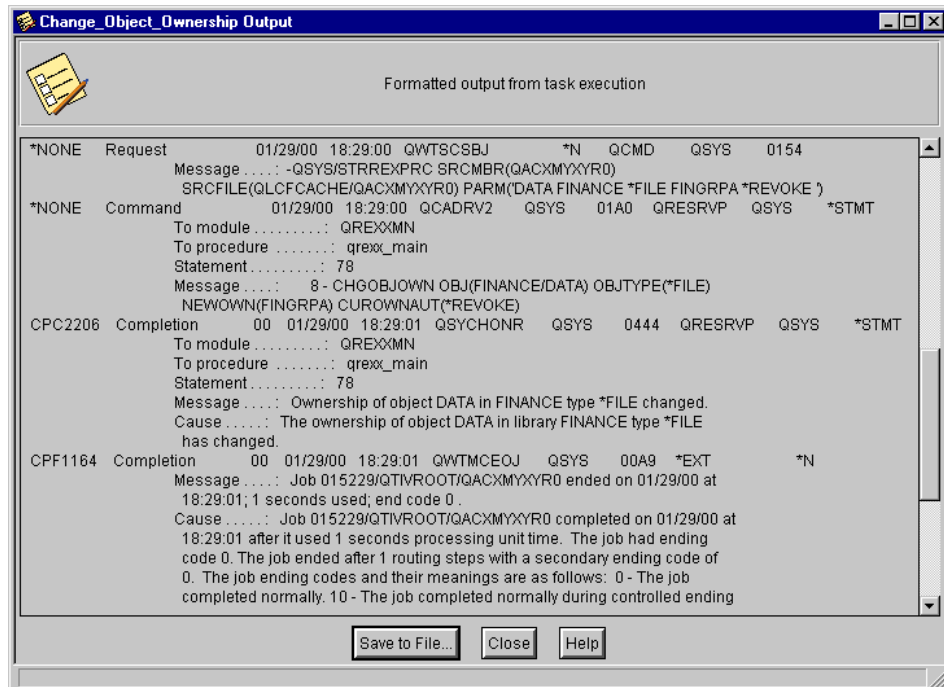


Figure 99. Change_Object_Ownership output

The CPC2206 message confirms that the ownership of the file was successfully changed.

Now that the DATA file is no longer owned by BDAVIS, we decide to set the Delete Action to *DLT and then delete the BDAVIS user profile. To do this, we open the Tivoli user profile and select to Edit the BDAVIS user record. We make the change shown in Figure 100 on page 206 on the OS400 Login panel.

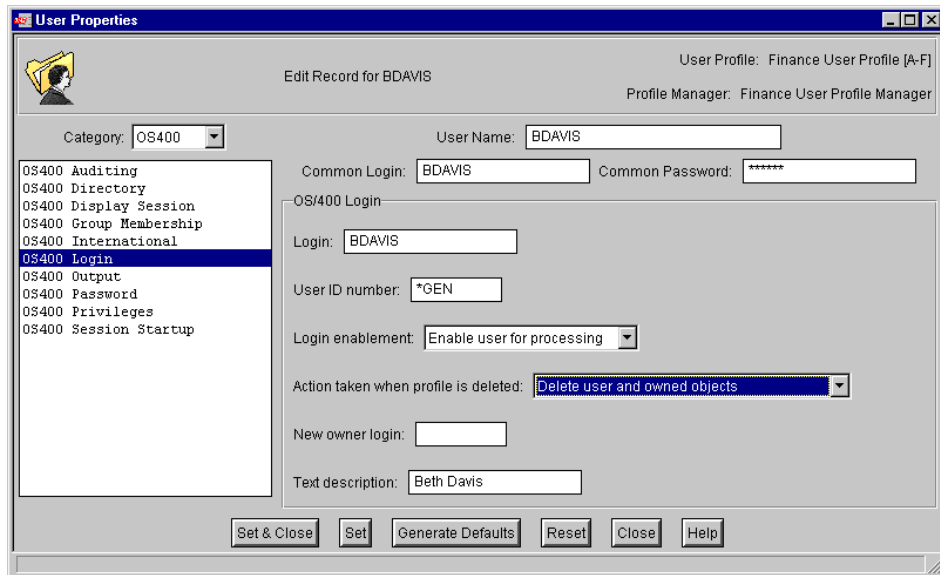


Figure 100. OS/400 login attributes

Press the **Set and Close** button. This puts us back in the user profile. With the BDAVIS entry selected, we press Delete Users to delete the user record. Figure 101 shows the User Profile Properties screen.

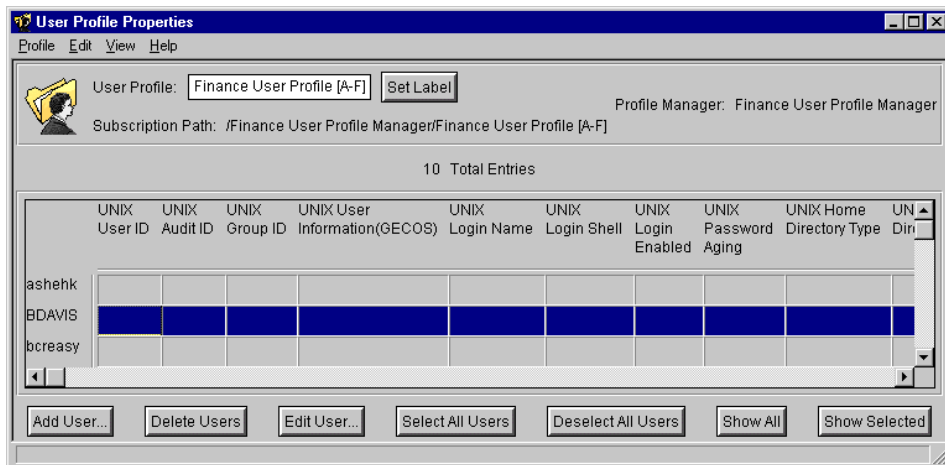


Figure 101. Deleting BDAVIS user record

The user profile is updated showing that the BDAVIS record has been deleted as shown in Figure 102 on page 207.

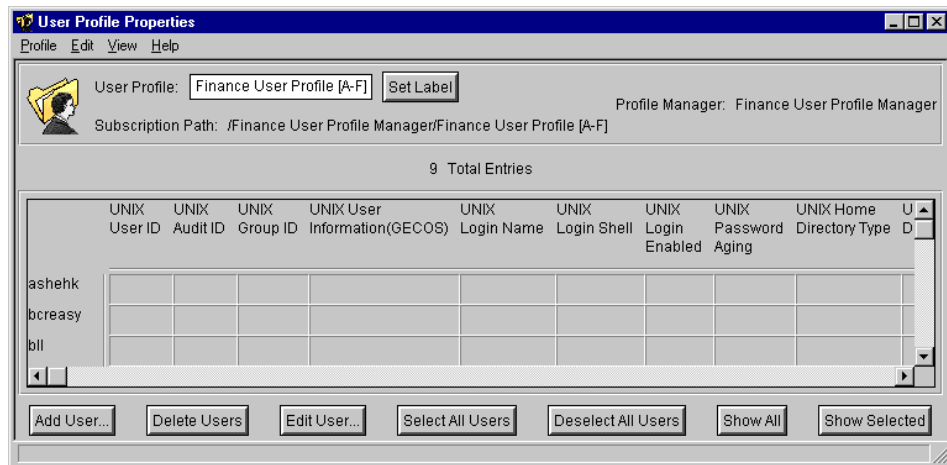


Figure 102. User profile with the DBAVIS record removed

At this time, no change has yet been made to the BDAVIS user profile on the AS/400 endpoint. To actually remove the BDAVIS user profile from the AS/400 endpoint, we must distribute the profile as shown in Figure 103 on page 208.

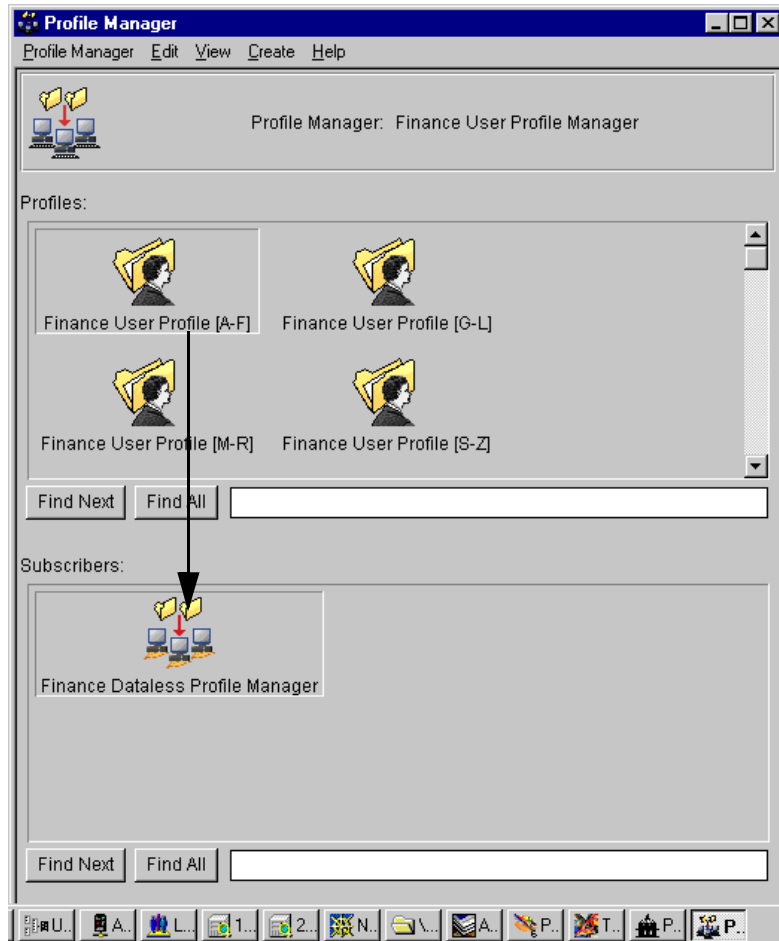


Figure 103. Distributing the user profile

Once the profile has been deleted, the Work with User Profiles (WRKUSRPRF) command:

```
WRKUSRPRF USRPRF (*ALL)
```

shows that the BDAVIS user profile has successfully been deleted as shown in Figure 104 on page 209.

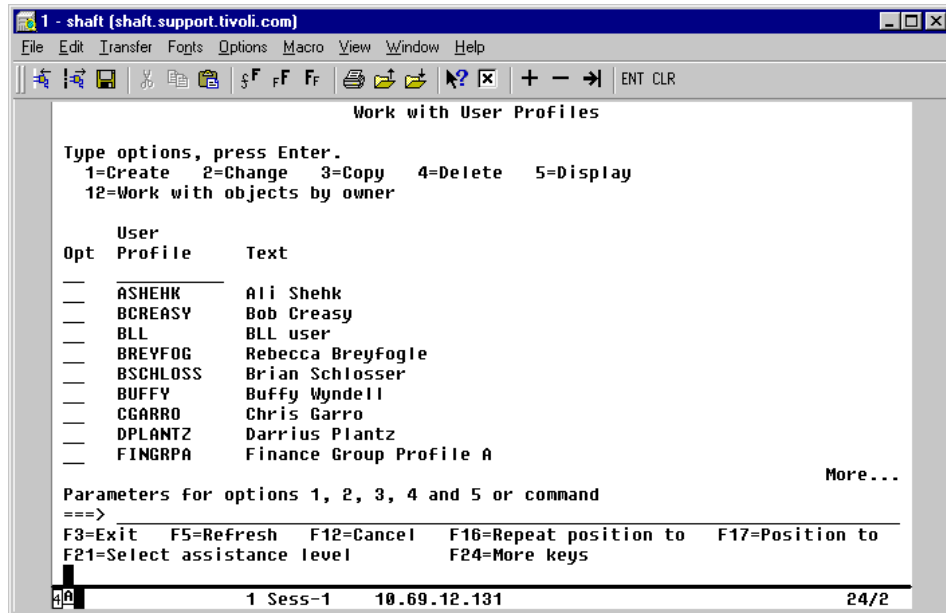


Figure 104. Work with user profiles

6.3 Tivoli Security Management for AS/400 Endpoints

Protecting information and business resources is critical to the success and productivity of any enterprise.

The daily work of security administrators for large organizations with a variety of computing platforms is a tough job that gets tougher all the time. For example, in a large network with interconnected Windows NT workstations, UNIX systems, and AS/400 systems, the sheer volume of accounts and trust relationships is known to require an inordinate amount of administrative time. Security administrators need to have a detailed security knowledge of all platforms in use, and they spend a lot of time trying to integrate security solutions that are never designed to fit together.

One way of solving these problems is to use Tivoli Security Management. This product provides centralized role-based security administration across distributed computing platforms, such as UNIX, Windows NT, AS/400, OS/390, NetWare, or OS/2.

With Tivoli Security Management, you can manage system security on Managed Nodes as well as TMA Endpoints including AS/400 Endpoints.

Please refer to the redbook, *Tivoli Security Management Design Guide*, SG24-5101, for more detailed information about Tivoli Security Management.

6.3.1 Installing Tivoli Security Management for AS/400

Tivoli Security Management for AS/400 Endpoints consists of optionally-installable patches that allow the Tivoli Security Management, Version 3.6.1, product to support AS/400 Endpoints. Therefore, before installing these patches, both the Tivoli Security Management, Version 3.6.1, base product and the AS/400 Endpoint must be installed. For complete details of installing Tivoli Security Management, Version 3.6.1, refer to the *Security Management User's Guide 3.6*, GC31-8475. Refer to Chapter 3, "AS/400 Endpoint installation and deployment" on page 37 of this book, for AS/400 Endpoint installation details including:

- Hardware requirements
- Software requirements
- Installation methods
- A complete AS/400 Endpoint Enablement Patch Listing

Additionally, be sure to completely review the *Tivoli AS/400 3.6.1 Release Notes*, GI10-8016, since this document provides the most current information for the product including:

- New features
- Installation notes
- Documentation changes
- Defects, workarounds and limitations

Descriptions of the enablement patches necessary for Tivoli Security Management for AS/400 Endpoints follow.

3.6.1-SEC-0001

This is the TME 10 Security Management for AS/400 Endpoints, Version 3.6.1 patch, and it contains the Security Management graphical user interface extensions to be installed on the TMR server to support AS/400 endpoints.

Note

It is very important that this patch be installed *only* on the TMR server. Installing this patch on a managed node will corrupt the installation.

3.6.1-SEC-0002

This is the TME 10 Security Management Gateway for AS/400 Endpoints, Version 3.6.1, and it contains the Security Management methods which the Security Management gateway will need to support AS/400 Endpoints. This patch needs to be installed on any Security Management gateway which will support AS/400 Endpoints.

Note

Just as when managing TMA Endpoints, it is a good rule of thumb to install the Security Management Gateway on all of the Endpoint Gateways, it is also recommended that you install this Tivoli Security Management Gateway for AS/400 Endpoints patch on all of your User Administration Gateways. By doing this, you are assured that the AS/400 Endpoints will remain supported should you reconfigure your gateways or move the AS/400 Endpoints.

6.3.2 Using Tivoli Security Management for AS/400

This section introduces an example of using Tivoli Security Management for AS/400.

6.3.2.1 Creating a security profile

First, we will create the Security profile to be used in the following sections:

1. Select the appropriate profile manager created earlier.
2. From the Profile Manager dialog box, select **Create -> Profile** from the menu bar as shown in Figure 105 on page 212.

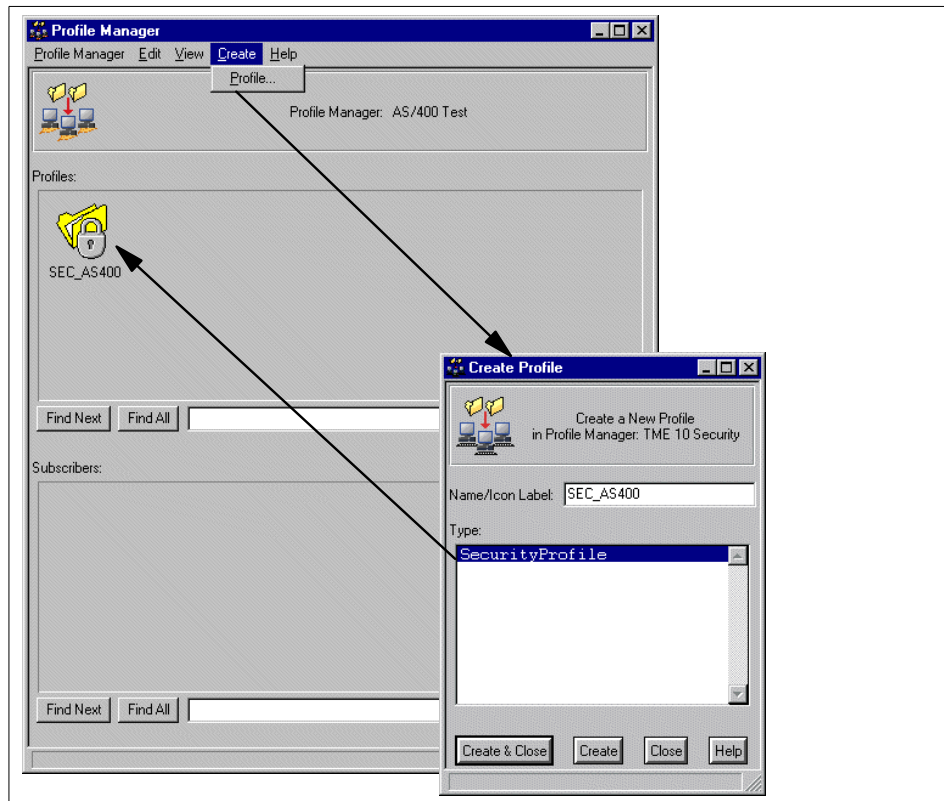


Figure 105. Creating Security profile

3. From the Create Profile dialog box, select **SecurityProfile** as the profile type. Then, give the profile a name, and press the **Create & Close** button.

6.3.2.2 Populate Security profile

The next step is to populate the profile from an AS/400 Endpoint. In this example, we use the AS/400, ishii1.

1. Select the appropriate Security profile created before and double click on the icon.

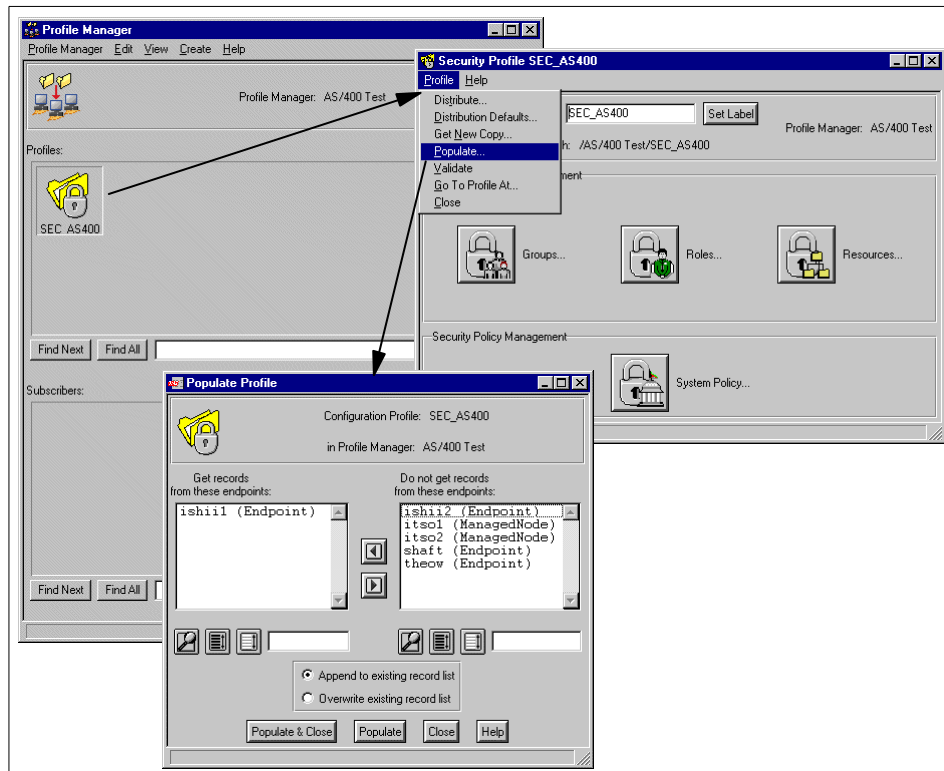


Figure 106. Populating the security profile

2. Select **Profile -> Populate** from the Security Profile menu bar.
3. From the Populate Profile dialog box, add subscribers from the list of available subscribers.
4. Press the **Populate & Close** button on the Populate Profile dialog box.

6.3.2.3 Populated security profile

The following two figures provide examples of the type of information held in the populated Security profile. This information can then be modified as required and distributed to other AS/400s in your TMR. The profile was populated with the Security Group as shown in Figure 107 on page 214.

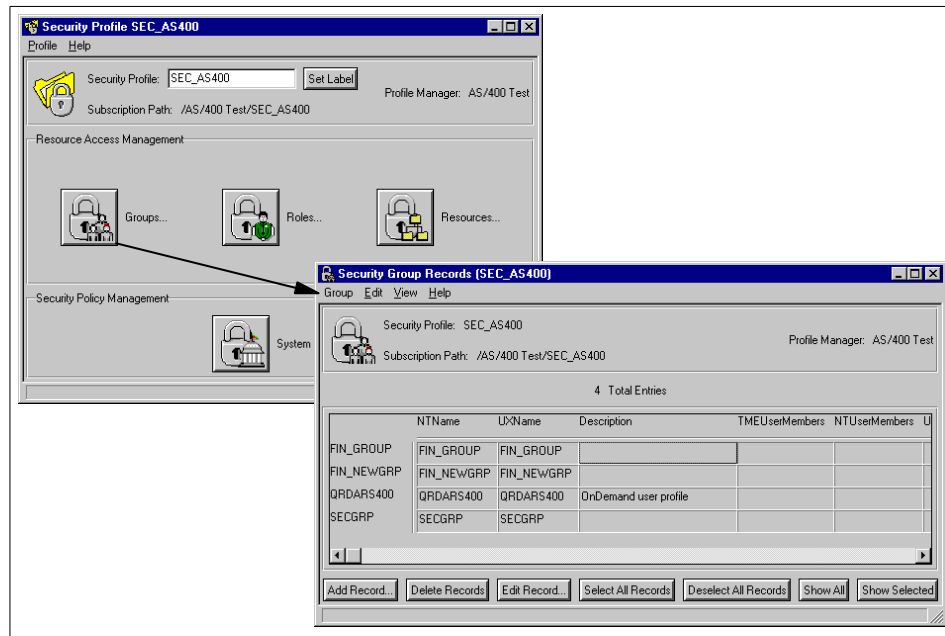


Figure 107. Security group information

Figure 108 on page 215 shows Security System Policy information.

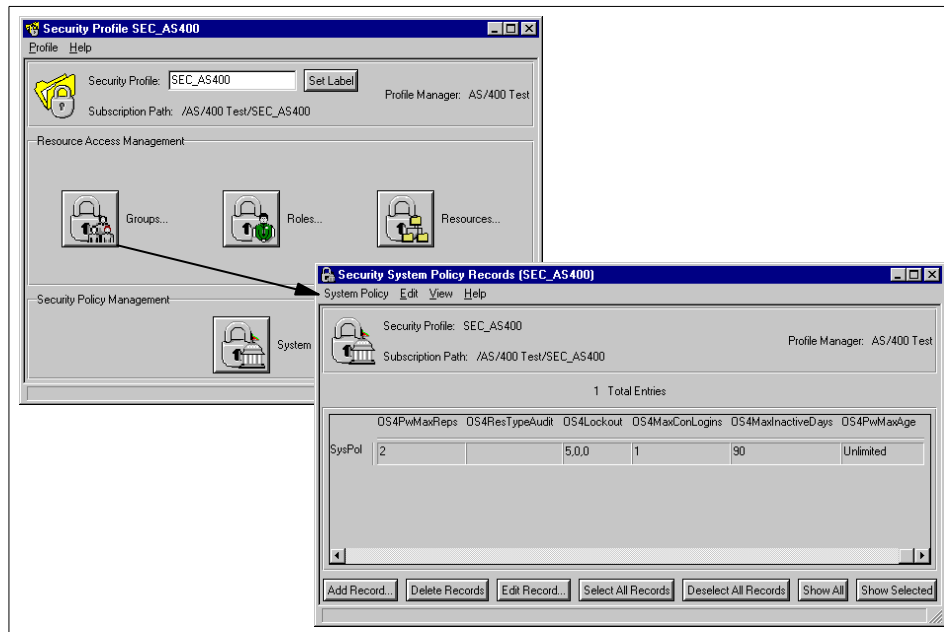


Figure 108. Security system policy information

We will introduce another example of Tivoli Security Management in a subsequent chapter; so, refer to Chapter 9, "AS/400 management scenarios" on page 437, for more information about Security Management.

Chapter 7. AS/400 availability management

Tivoli provides availability management for the AS/400 in much the same way as it does for all supported TMA Endpoints. Both Distributed Monitoring and the TEC AS/400 Event Adapters can be used to monitor critical events on the machine. TEC Message Adapters can be used to monitor message queues while the TEC Alert Adapter monitors OS/400 Alerts. The Distributed Monitoring engine runs either standardized monitors from a supplied monitoring collection or custom monitors created by the user to query data on the machine and report the result back to the monitoring engine.

7.1 Overview of Tivoli availability management applications

With Version 3.6.x, Tivoli has extended its management ability to the AS/400 platform. The Tivoli availability management solution covers AS/400 systems as well as other platforms, and it enables you to perform seamless operations in Tivoli availability management.

The Tivoli availability management solution consists of the following Tivoli applications:

- Tivoli Distributed Monitoring
- Tivoli Enterprise Console AS/400 Event Adapter

The following sections introduce the overview of Tivoli availability management applications.

7.1.1 Tivoli Distributed Monitoring

Tivoli Distributed Monitoring is an application that allows you to monitor the status of a wide range of geographically-dispersed platforms from different vendors running different operating systems including resources that are not part of your Tivoli environment.

A monitor is an entry that controls specific aspects of a resource, such as disk utilization, the status of a print queue, database process status, CPU load average, network collisions, and so on. Its definition contains threshold values and various response actions triggered upon reaching a threshold.

Tivoli Distributed Monitoring uses the concept of management by subscription as the other Tivoli core applications. Monitors are defined centrally in Distributed Monitoring (Sentry) profiles and then distributed and activated on the target systems.

Tivoli Distributed Monitoring provides your network computing environment with the following features.

- Centralized monitoring of remote resources.
- Predefined monitors for almost every resource (monitoring collections).
- Strong mechanism to generate events and alarms.
- Automated decisions and actions in response to alarms or events.
- Various responses (e-mail, running program, pop-up, and so on).
- Custom scripts for monitoring specific applications.
- Full integration with the Tivoli Enterprise Console (TEC) event server.
- Data collection for statistical analysis and capacity planning.

7.1.2 TEC AS/400 Event Adapter

Tivoli Enterprise Console (TEC) is a rule-based event management application that uses a central server to process incoming events. These incoming events are generated by adapters running on hosts throughout a TMR. These adapters are either TME or non-TME adapters. The distinction between these types is made based on how the event will reach the event server. A TME event adapter sends events through the object request broker (oserv) on the Managed Node or gateway to the oserv on the TEC event server. A non-TME event adapter will send events from the adapter directly to the *tec_reception* process on the TEC event server.

The details of the TEC event server daemons are described in detail in the redbook, *Tivoli Enterprise Internals and Problem Determination*, SG24-2034, and in the *Tivoli Enterprise Console User's Guide*. Additional information on the available TEC Adapter's including the TEC AS/400 Adapters can be found in the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507.

We will assume a knowledge of Tivoli Enterprise Adapter concepts including such things as:

- How events get sent to the Event Server from a Non-TME Adapter.
- Event Classes and Attributes.
- Adapter Configuration Files and Keywords.

A thorough reading of Chapter 1, "Understanding Adapters" in the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507, should provide you with the necessary information. This chapter is also very good in pointing out AS/400-specific differences where appropriate.

7.2 Tivoli Distributed Monitoring for AS/400

The AS/400 is one of the supported Endpoints for the Tivoli product. As such, the redbook *Introduction to Tivoli Enterprise*, SG24-5494, contains a complete description of Distributed Monitoring for Tivoli. These are highlights.

Profiles and profile managers

Distributed Monitoring uses the monitor by subscription concept where management regions are created. Within a region, profile managers are created and contain profiles and each manager has subscribers that will act as a distribution point for the profiles contained within. Special considerations are needed when designing the profile management hierarchy and are not covered here. The redbook *An Introduction to Tivoli Enterprise*, SG24-5494, goes into detail about how to design the distribution system.

For Distributed Monitoring, you create a data profile manager where the profiles are stored. A dataless profile manager is also created to be a subscriber to this data profile manager and will contain local copies of the profiles that may be modified as needed. All AS/400 endpoints are, in turn, subscribed to this dataless profile manager and should be grouped so that all subscribing machines will always be receivers of all profile copies of this manager. When endpoints should not receive a profile to work with, they must be in a separate dataless profile manager.

Monitoring collections

The actual code that performs the monitoring actions on an endpoint is contained in a monitoring collection stored on the TMR server. This monitoring collection is downloaded via an endpoint gateway to the endpoint being monitored when it is needed. Custom monitoring collections can be created by the user using the mcsI utility, but, normally, the user would simply create a custom monitor via the Universal collection for monitoring special needs.

How the engine works

The Distributed Monitoring engine is a long-running daemon that is installed on the endpoint when the very first dm profile is pushed to the endpoint. Along with the actual engine code, other programs are downloaded on this initial push. They are stored in an out-of-cache location on the endpoint because they cannot be swapped out when the cache is full.

The following things happen upon startup of the engine (DMEPENGINE) on the endpoint:

1. The engine makes an upcall, via the lcf process, to load the collection list (all available collections and their object IDs) from the endpoint gateway. If the collection is not stored in memory on the gateway, it first downloads it from the TMR server and then passes it to the engine. Only the initial connection from the lcf to the gateway takes place over the default port 9495, all other communications and upcalls take place over an out-of-band port that is not controllable by the user.
2. Once the collection list is in place, the engine reads the contents of the QIBM/UserData/Tivoli/LCF/.sntcfg file to load the profiles it already has stored. If the engine has started as a result of a profile push (the engine was either shut down or it is the initial push), it accepts the new profile data from the DOGENPT process started by the profile push.
3. The engine starts its polling loop. At the top of each minute, the engine examines all of the monitors it has and marks any scheduled for that minute as ready to run. It then launches a single monitor and waits for a response. It will wait for up to 60 seconds for the monitor to complete. If the monitor is still running or has not reported back after 60 seconds, the engine process kills off the monitor and reports a NO DATA error in the log file and any response. If the monitor returned data to the engine prior to this 60 second time period, normal responses take place.

Note

When determining how many monitors to run and how often to run them, you need to understand what happens in the engine. Since a single monitor can take up to 1 minute to run, if you distribute a profile with 10 monitors that run each minute, they may not actually run each minute. The engine goes through the ready to run list one at a time. If 10 monitors are scheduled and each takes 30 seconds to run, it will take 5 minutes to run the current set. The next available time for a monitor to run would be 5 minutes later. If the engine is overloaded in this manner, monitors will eventually run but it won't be at the time you expect them to. In the example here, before the engine would go look for a new top of minute, it would first look to see if any monitors were scheduled to run during the time it was busy and would immediately start running them. In our example of running each minute, it would see that even though it missed running the same monitor 4 times, it would only schedule it to run once more. Be careful about giving the engine more to do than there is time available. Depending upon the system load, a monitor will take some variable amount of time to run. Start small and add monitors one at a time to see when they run.

For more detailed information about Distributed Monitoring performance, please refer to the redbook, *Tivoli Enterprise Performance Tuning Guide*, SG24-5392.

What it reports, actions taken

When configuring a monitor, you set it to respond to data that is either a string of text or a number for each of five possible states:

- critical
- severe
- warning
- normal
- always

Note that these do not correspond exactly to TEC error levels. These states do not have a default meaning; you determine what values make sense for what you are monitoring. For each of these states, you have possible responses:

- Send a Tivoli notice (Sentry, Sentry-log, and Sentry-urgent).
- Pop-up a message on one or many Administrators' desktops.
- Change icon (use a previously-defined Indicator collection).

- Run a task on the endpoint (normal Framework task).
- e-mail the response data to someone (no checking is performed to see if the address is reachable or valid).
- Log the response data to a file either locally on the endpoint or remotely on a Managed Node.
- Run a program either locally on the endpoint or on a remote Managed Node.
- Send an event to TEC (a primary server or a backup server only if the primary is down).

Note

To allow TEC to correctly parse Distributed Monitoring events from the AS/400 Monitors, you must first import the Tivoli Distributed Monitoring OS/400 baroc file appropriate for the monitor being used into the TEC rule base. These baroc files are located in the \$BINDIR/./generic/SentryMonitors directory.

You can use any or all responses for each state defined.

The data sent is the response data from the individual monitor run and can be in one of several message styles selected when the monitor is created.

The *always* state says to always perform this action even if a trigger level is not reached. This is useful for testing whether a monitor is running properly and has a time stamp; so, you can see exactly when it ran.

When do the monitors run?

When a monitor is created, you give it a start date and time for the initial run. You also provide a time interval for the monitor to repeat. This time is the TMR server's local time. You must make sure the time zone is properly set on the endpoint using the system value, QUTCOFFSET. This can be checked and adjusted using the Work with System Values (WRKSYSVAL) command:

```
WRKSYSVAL SYSVAL(QUTCOFFSET)
```

Since the time is set using the TMR servers time and time zone, the monitor may not run when you expect if your endpoint is not in the same time zone. A monitor set to run at 5:00 p.m. TMR server time (assuming the TMR is set for central/US time) will run at 5:00 p.m. central time on your machine. If your local time zone is PST/US, this monitor would run at 3:00 p.m. your time. You may have to create multiple monitors in separate profiles if these times are not correct for your environment.

What user is running the monitors?

When the lcf daemon is installed, Three user ID's are created:

- QTIVROOT
- QTIVUSER
- QTIVOLI

Monitors supplied in a monitoring collection run as the QTIVROOT user. Run a program, log to file, and any custom monitors run as the user and group defined for each monitor (see the *Distributed Monitoring User's Guide 3.6*, GC31-8382) and default to QTIVUSER. You need to make sure that the user id the monitor has sufficient rights to perform the desired operations. See the *Tivoli Framework AS/400 3.6.2 User's Guide*, GC32-0279.

What is the difference between AS/400 and other platforms?

What is the difference between AS/400 and other endpoints running Distributed Monitoring? When you monitor AS/400 Endpoints using Distributed Monitoring, you should consider the following differences.

- The name of the dm engine is DMEPEENGINE instead of dm_ep_engine.
- The name of the distribution endpoint is DOGENDPT instead of dogEndpoint.
- Custom monitor.

There is no difference between using and configuring Distributed Monitoring for AS/400 Endpoints and other platforms. Refer to the rest of this chapter and Chapter 9, "AS/400 management scenarios" on page 437, for more detailed information about Distributed Monitoring.

7.2.1 Installing Tivoli Distributed Monitoring for AS/400

The lcf daemon must be installed per the *Distributed Monitoring User's Guide 3.6*, GC31-8382, before Distributed Monitoring profiles can be pushed to the endpoint. For the endpoint, no Distributed Monitoring code is installed directly but is pushed to the endpoint upon the first profile push. You must install Distributed Monitoring on the TMR server and all endpoint gateways (see the *Distributed Monitoring User's Guide 3.6*, GC31-8382, for more detailed information). The monitoring collections for the AS/400 must be installed on the TMR server only. See the *Distributed Monitoring User's Guide 3.6*, GC31-8382, and the Release Notes for the current list of collections that can be installed. In addition to installed Distributed Monitoring and the monitoring collections, the following additional enabling patches need to be installed on the endpoint gateways.

3.6.1-DMN-0001 Contains the engine and assorted files.

3.6.1-DMN-0002 For ARM transactions.

For information about installing all products, see the *TME 10 Framework 3.6 Planning & Installation Guide*, SC31-8432.

Tivoli Distributed Monitoring for AS/400 Endpoints consists of optionally installable patches that allow the Tivoli Distributed Monitoring, Version 3.6.1, product to support AS/400 Endpoints. Therefore, before installing these patches, both Tivoli Distributed Monitoring, Version 3.6.1 base product, and the AS/400 Endpoint must be installed. For complete details of installing Tivoli Distributed Monitoring, Version 3.6.1 refer to the *Distributed Monitoring User's Guide 3.6*, GC31-8382. Refer to Chapter 3, "AS/400 Endpoint installation and deployment" on page 37 of this redbook, for AS/400 Endpoint installation details including:

- Hardware requirements
- Software requirements
- Installation methods
- A complete AS/400 Endpoint Enablement Patch listing

Additionally, be sure to completely review the *TME 10 AS/400 3.6 Release Notes*, GI10-4775, because this document provides the most current information for the product including:

- New features
- Installation notes
- Documentation changes
- Defects, workarounds, and limitations

The enablement patches necessary for Tivoli Distributed Monitoring for AS/400 Endpoints are:

3.6.1-DMN-0001

Tivoli Distributed Monitoring for AS/400 Endpoints, Version 3.6.1

Contains the Distributed Monitoring engine and probes to be installed on the endpoint gateway to support AS/400 endpoints.

3.6.1-DMN-0002

Tivoli Distributed Monitoring ARM for AS/400 Endpoints, Version 3.6.1

Contains the Application Response Management (ARM) agent to be installed on the endpoint gateway to monitor availability and performance transactions for supported AS/400 Endpoints.

Refer to the *Tivoli Framework AS/400 3.6.2 User's Guide*, GC32-0279, for more detailed information about ARM for AS/400.

7.2.2 Understanding Tivoli Distributed Monitoring for AS/400

This section introduces some AS/400 Endpoint-specific information about Tivoli Distributed Monitoring.

7.2.2.1 Run Programs responses on an AS/400 Endpoint

A Run Program response that is run on the AS/400 Endpoint can be any AS/400 command that can be run from an OS/400 command line. In many cases, a desired response will be to run a program. Most programs can be started by using the Call Program (*CALL*) or Submit Job (*SBMJOB*) commands. If you have the Qshell Interpreter (5769SS1 Option 30) and Qshell Utilities (5799XEH) installed, you can also run scripts using the Start QSH (*QSH*) command. Note that you can use the Distribution Actions to distribute this script to the Endpoints from a source system.

To run a program via Distribution Actions or as a monitor response, you must use the exact AS/400 command syntax. For example, to run a QSHELL program called *pgm1.sh* in the */tmp* directory on an AS/400, you need to use the following syntax for the name of the program:

```
qsh cmd('/tmp/pgm1.sh')
```

Note

When using a Run Program response, you must assure that the Remote User ID value is set to an OS/400 user profile that is authorized to perform the command on the AS/400. The default value for this is *nobody*, which is mapped to the Tivoli-supplied user profile, QTIVUSER.

Refer to Section 7.2.4, “Troubleshooting Tivoli Distributed Monitoring for AS/400” on page 227, for a complete discussion of authority checking when using the Run Program response on an AS/400.

The Remote Group ID field is ignored on AS/400 Endpoints.

Custom monitors

Tivoli supplies 15 standard monitoring collections for AS/400. They are divided into two basic groups, Core AS/400 monitors and Network Protocol

collections. The *Tivoli Framework AS/400 3.6.2 User's Guide*, GC32-0279, details these collections and what values you need to provide when the collection is used. We also strongly recommend that you refer to the *TME 10 AS/400 3.6 Release Notes*, GI10-4775, for up-to-date information about AS/400 monitors.

In the case where a standard monitor does not monitor your special case, a custom monitor can be created using either a custom string or custom numeric monitor. The type you select is determined by what your custom program will return to the engine, a string quantity, such as PASSED or a number, such as 100. That is the only difference between the two monitor types.

Custom monitors that run shell scripts do not require any special syntax for the AS/400 (once the GA patch for e-fix 44E is available).

Custom monitors in the Universal collection are not supported on the AS/400.

7.2.3 Customizing Tivoli Distributed Monitoring for AS/400

This section introduces some examples of Distributed Monitoring customization.

7.2.3.1 Run Program Responses

As mentioned earlier, a Run Program response can be any command that can be run from an OS/400 command line. For instance, you might want to:

- Send a message to the AS/400 System Operator using the Send Message (SNDMSG) command.
- Start a print writer using the Start Print Writer (STRPRTWTR) command.
- Run a program using the Call Program (CALL) command or Submit Job (SBMJOB) commands.

Note

When running a program as a response, we suggest using the Submit Job command over the Call Program command. A program started using the CALL command will actually execute within the DMEPENGINE job. The drawbacks of this are two-fold:

1. The executed job will be run within the DMEPENGINE job environment including such things as the library list, logging level, and printer device. Additionally, this program would produce no joblog. Rather, any program messages for it would actually be part of the DMEPENGINE job.
2. The DMEPENGINE may be adversely affected by the program's activities.

Additionally, the Submit Job (`SBMJOB`) command allows you to run your program (or simply a command) as a separate job. You can specify the appropriate job attributes including such things as Job name, Priority, Library List, Job Queue, and Job Description.

In the scenario in Section 7.3.5.1, “Editing the configuration file” on page 341, we use the `CALL` and `SBMJOB` commands to illustrate running a CL program as a Run Program response.

7.2.3.2 Custom monitors

A custom monitor also provides very powerful monitoring capability to AS/400 availability management. Refer to Appendix A, “Custom scripts for AS/400 Distributed Monitoring” on page 503, for more detailed information about AS/400 custom monitors. We also strongly recommend that you refer to Section 7.2.5, “AS/400 Distributed Monitoring scenarios” on page 234.

7.2.4 Troubleshooting Tivoli Distributed Monitoring for AS/400

Along with the normal debugging techniques used for the Endpoint, Distributed Monitoring can also write more specific details to its own log file called DM36.log. Additionally, on the AS/400, the joblogs for the Distributed Monitoring Processes may also provide detailed messages in the event of an error. We will now discuss the use of these tools in more detail.

Note

Refer to the following redbooks for more detailed information about the normal debugging techniques for the Endpoint.

- *All About Tivoli Management Agents*, SG24-5134
- *Tivoli Enterprise Internals and Problem Determination*, SG24-2034
- *Tivoli Enterprise Performance Tuning Guide*, SG24-5392

These redbooks provide very detailed technical information about the Endpoint and should be useful when debugging or troubleshooting.

7.2.4.1 Setting the Log Level for the DM engine

By default, the DMEPENGINE logs to the DM36.log file in which /QIBM/UserData/Tivoli/LCF directory with a log level of 1. When trying to determine the cause of a problem, technical support may ask you to increase the log level from 0 to some other level between 1 and 4.

If you have the Qshell Interpreter and the Qshell Utilities installed, you can run the following commands from the AS/400 command line:

- QSH
This will put you into the QSH Command Entry screen. On the QSH command line, issue the following:
- `echo n > /tmp/dm36.ll`
where *n* is an integer between 1 and 4. The technical support engineer will let you know what value of *n* to use.
Finally, run the following command to verify that the change was made correctly:
- `cat /tmp/dm36.ll`
This should return the value *n*.

If you do have the Source Entry Utility available with the Application Development Toolset for AS/400 product, you can set the log level by issuing the following commands from an AS/400 command line:

1. `CRTSRCPF FILE(QTEMP/DM36.LL)`
2. `ADDPFM FILE(QTEMP/DM36.LL) MBR(LEVEL)`
3. `STRSEU SRCFILE(QTEMP/DM36.LL) SRCMBR(LEVEL)`

This will bring the empty file into the Source Entry Utility editor. You must insert a new line, which can be done simply by pressing **Tab** to move to

the line reading *Beginning of data*, pressing **I**, and then pressing **Enter**. On the new line that is inserted, type the appropriate log level value. The file should be similar to the one in Figure 109 in which a 3 was inserted.

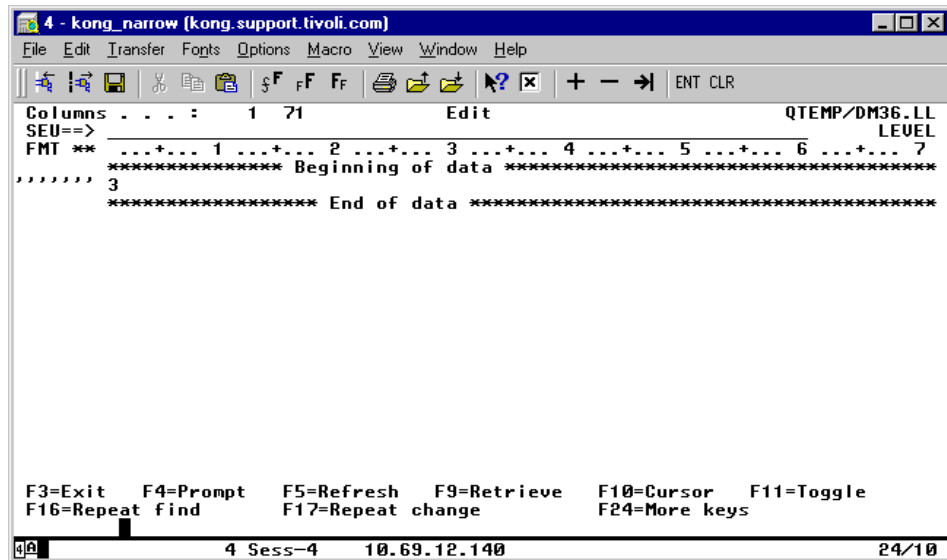


Figure 109. Edit using Source Entry utility

4. Press **F3** (Exit) and be sure to select **Y** on the Change/create member parameter on the Exit screen.

Once the file is properly created, run the following command:

5. `CPYTOSTMF FROMMBR(' /QSYS.LIB/QTEMP.LIB/DM36.LL.FILE/LEVEL.MBR')`
`TOSTMF(' /tmp/dm36.11') STMFOPT(*REPLACE)`

Note

This operation will perform the equivalent UNIX commands of:

```
echo 3 > /tmp/dm36.11
```

If you have qsh installed, you can issue this UNIX command. Possible levels are 1, 2, 3, and 4. If you set a value above 4, it will be set to 0.

Finally, if you cannot use either the QSH or SEU, you can set the log level using the Data File Utility by issuing the following commands:

1. `CRTPF FILE(QTEMP/DM36.LL) RCDLEN(1) MBR(LEVEL)`
2. `STRDFU OPTION(5) FILE(QTEMP/DM36.LL) MBR(LEVEL)`

This will run the Data File Utility default program and bring up the screen in Figure 110.

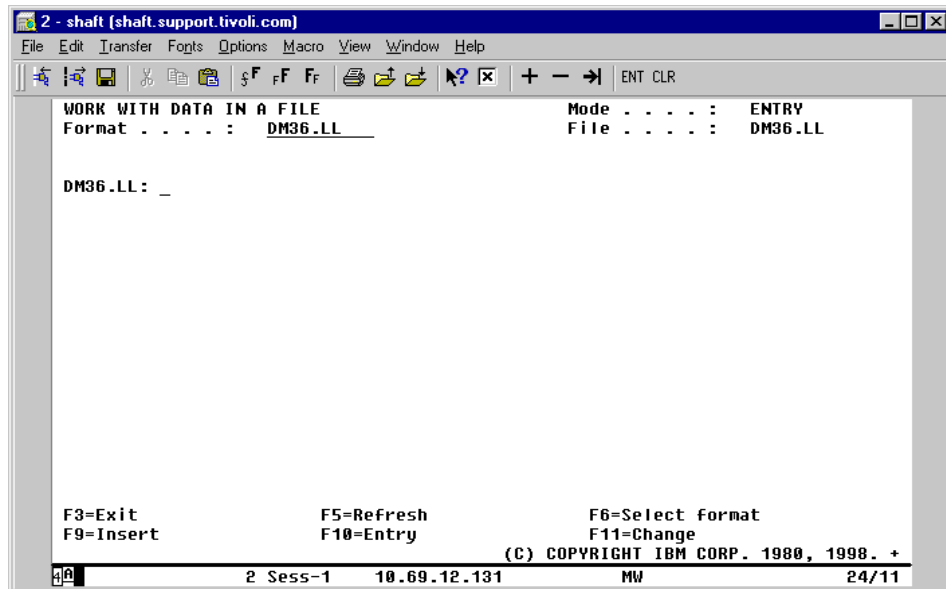


Figure 110. Work with data in a file

On this panel, perform the following:

3. On the blank line displayed, enter the appropriate log level and press **Enter**.
4. Press **F3** (Exit).

The End Data Entry screen should appear as shown in Figure 111 on page 231.

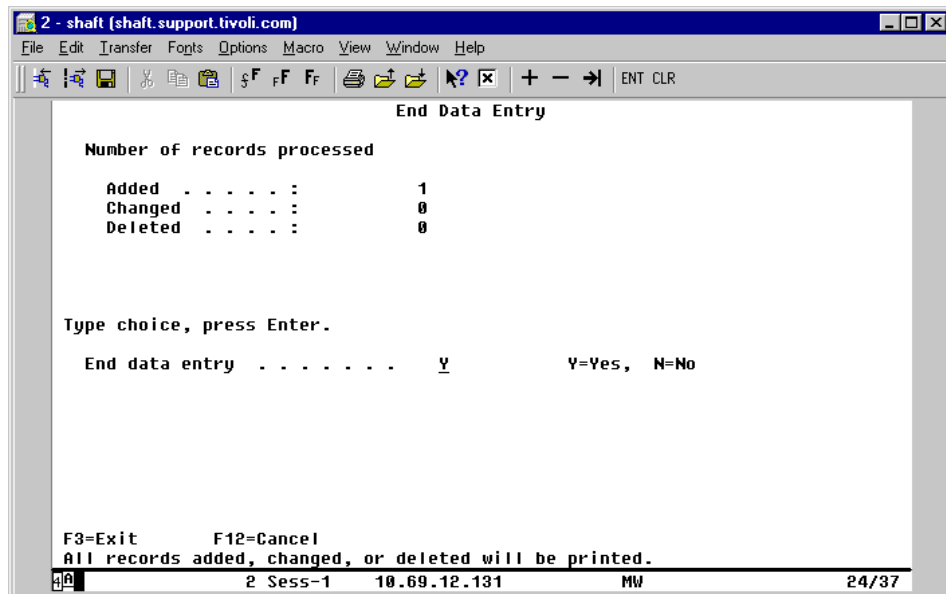


Figure 111. End Data Entry

5. Press **Enter**.

Once the file is properly created, run the command in Step 6.

6. `CPYTOSTMF FROMMBR ('/QSYS.LIB/QTEMP.LIB/DM36.LL.FILE/LEVEL.MBR')`
`TOSTMF ('/tmp/dm36.11) STMFOPT (*REPLACE)`

Once the log level is set, the DMEENGINE must be stopped and restarted for the new log level to become active.

7.2.4.2 Debugging Run Program Responses on the AS/400

The most common problems encountered when using Run Program responses are incorrectly-entered commands (syntax errors) and authority issues. The simplest way to test that the command is valid as typed and that the Remote User ID specified has the necessary authority to do this is to simply sign on the AS/400 as that user profile and attempt to run the command exactly as it has been entered on the Run Program field.

However, if the command being run is the Submit Job (`SBMJOB`) command, which specifies a `USER` value, the process is slightly more complicated. Let us examine in more detail how the Remote User ID and the `USER` parameter of the `SBMJOB` command are used. Consider the following generic example:

Let us suppose we have a Distributed Monitoring profile with the following specifications:

Remote User ID USERA

Run Program SBMJOB CMD(command-string) USER(USERB)

Within the DMEPENGINE job running as QTIVROOT, the following occurs:

1. The DMEPENGINE job swaps to User ID USERA (the Remote User ID)
2. The DMEPENGINE job, now running as USERA issues:

SBMJOB CMD(command-string) USER(USERB)

USERA (the Remote User ID), therefore, must be authorized to run the Submit Job command exactly as specified.

3. An entirely separate job will now be started as a result of the Submit Job command. This job will run with the attributes specified on the command. Therefore, it will be running as USERB and will issue the specified *command-string*.

Therefore, USERB must be authorized to run the *command-string*.

Note

In this case, the Remote Group ID is ignored on AS/400 Endpoints.

7.2.4.3 Examining Distributed monitoring joblogs

The following are the Distributed Monitoring job names on the AS/400.

DMEPENGINE	The Distributed Monitoring Engine
LCF-CLI	Processes Distributed Monitoring commands, such as the <i>wlseng</i> command
DOGENPT	Handles Distributed Monitoring profile distribution
MONITOR	Runs the Distributed Monitoring probes

In general, joblogs for these processes can be found in numerous ways including the following:

- If the job is active, you can access its joblog by using the Work with Active Jobs command, *WRKACTJOB*, to locate the job running on the system. From the Work with Active Jobs panel, you can select Option 5 (Work with) beside the job to bring up the Work with Jobs menu. From this menu, select **Option 10** to display the joblog. Once in the joblog, press the **F10** (Display) detailed messages key.

- If the job has ended, you can use the Work with Spool Files command, WRKSPLF, as shown here to see if the job produced a spooled joblog:

```
WRKSPLF SELECT (QTIVROOT *ALL *ALL jobname)
```

where *jobname* corresponds to the appropriate job name listed above. For example:

```
WRKSPLF SELECT (QTIVROOT *ALL *ALL MONITOR)
```

This will bring all of QTIVROOT's spoolfiles, which specify MONITOR as the User Data, into the Work with All Spooled Files screen.

Note

The following Function keys on the Work with All Spooled Files screen are very useful, particularly if there are a large number of spooled files.

F18 = Bottom will take you to the last screen of spooled files. Since the spooled files are generally listed from first to last, the last spoolfile is usually the one from the last running of the job.

F11 = View 1/2/3/4 will display other spoolfile attributes. We find View 2 to be particularly helpful because it provides the spoolfile creation date and time, and this can often be helpful to confirm that the desired joblog is being examined.

In the scenarios provided in Section 7.2.5, "AS/400 Distributed Monitoring scenarios" on page 234, we will illustrate locating and examining Distributed Monitoring joblogs. We will also show how information in the DMEPENGINE joblog can be used to assist in locating joblogs for jobs started by using the Submit Job command as a Run Program response.

7.2.4.4 Problems with monitors not running

Additionally, because the MONITOR jobs also use the QTMELCF job description, you can follow the same process discussed in Section 6.2.4, "Troubleshooting Tivoli User Administration for AS/400" on page 176, to verify that the MONITOR jobs are getting to the QSYSNOMAX job queue and being submitted to the QSYSWRK subsystem.

Assuming that this is occurring, you will want to review the MONITOR joblog for errors as discussed in the preceding section.

Refer to the redbook, *Tivoli Enterprise Internals and Problem Determination*, SG24-2034, for detailed information about using the `wlseng -lz` command to review the Distributed Monitoring configuration output.

7.2.5 AS/400 Distributed Monitoring scenarios

This section introduces some AS/400 management scenarios using Distributed Monitoring.

7.2.5.1 Stopping and starting Distributed Monitoring engine

Upon the first Distributed Monitoring profile distribution, the `boot_engine` method gets added to the endpoint. From then on, whenever the Endpoint job (QLCFD) starts, it will start the DMEPENGINE job.

It may be necessary to remotely stop or start the DMEPENGINE on the Endpoint from other site. Generally, starting and stopping the DMEPENGINE job will be performed from any Managed Node using the `wstopeng -z ep_label` and `wlseng -z ep_label` commands. However, some time, you may wish to stop or start the DMEPENGINE from the AS/400 Endpoint itself. If you need to do this from the AS/400 machine, you should perform the following steps:

1. Before running these commands, you must change your current directory to be the `/QIBM/UserData/Tivoli/lcf` directory by issuing:

```
CHGCURDIR DIR('/QIBM/UserData/Tivoli/lcf')
```

Note

This command can also be issued using the following alternative file system command names:

- `CD`
- `CHDIR`

Refer to the *Integrated File System Introduction*, SC41-5711, for more detailed information on IFS commands.

2. Then, type the following command:

```
call qtmelcfdm/wstopeng <ENDPOINTNAME>
```

or

```
call qtmelcfdm/wlseng <ENDPOINTNAME>
```

Note

To list the possible Distributed Monitoring commands on the AS/400 machine, execute the following command:

```
dsplib qtmelcfdm
```

7.2.5.2 Creating a Distributed Monitoring monitor

This section shows some screen images of Distributed Monitoring and introduces an example of creating a Distributed Monitoring monitor.

1. Start the Tivoli Desktop and select the profile manager on the Tivoli Desktop by double clicking on the icon shown in Figure 112.

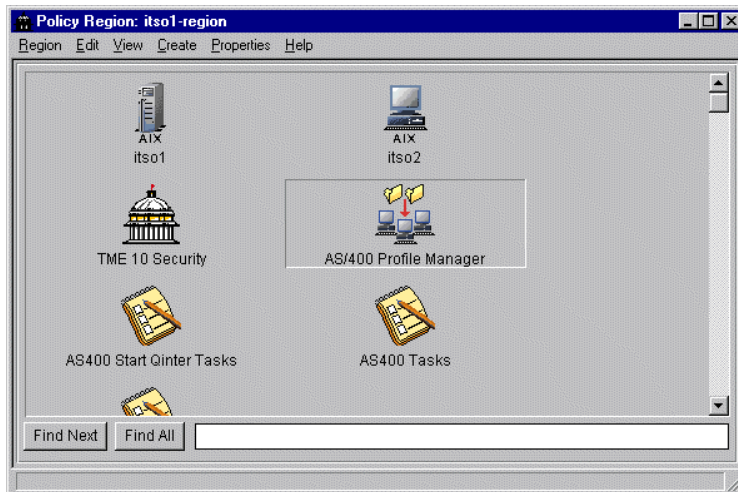


Figure 112. Selecting the profile manager on Tivoli Desktop

2. From the Create menu, select **Profile**. Then, pick a name for the profile (in this case, sentry_test) and make it a SentryProfile type. When you are finished, select **Create & Close**.

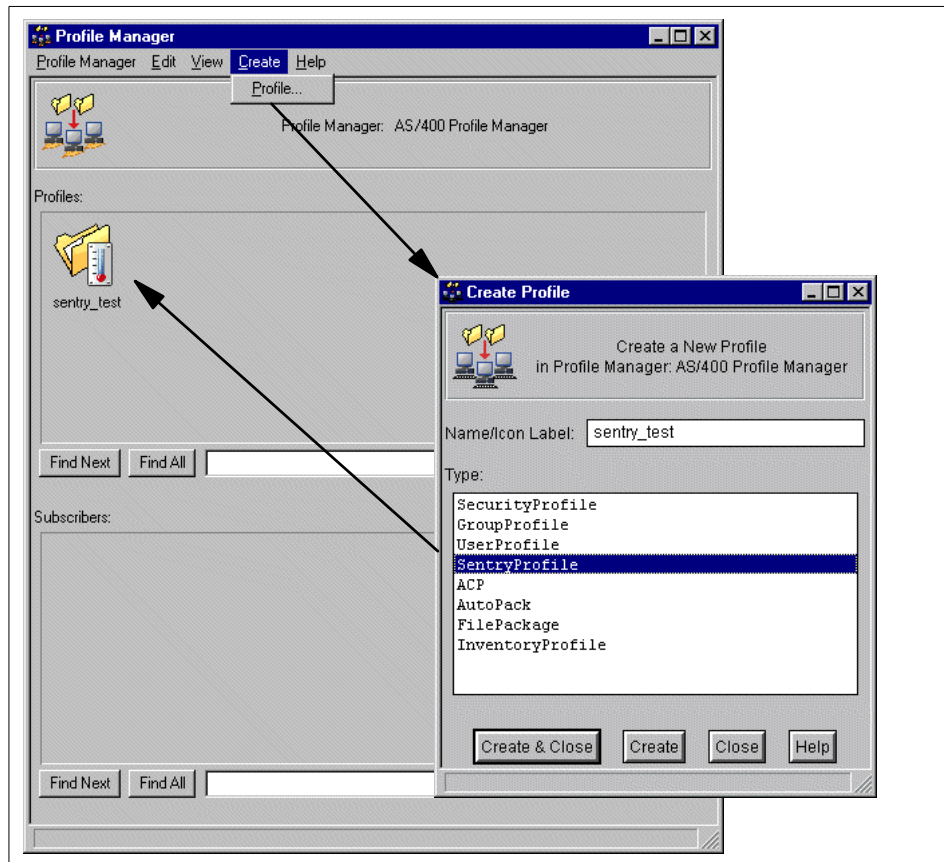


Figure 113. Creating a Distributed Monitoring profile

The profile sentry_test will appear on the profile manager screen with a icon showing an open file folder and an indicator (thermometer) icon to show you it is a Sentry profile.

3. To create a profile, double click on the icon, and you will see an empty profile. To add a monitor, select **Add Monitor** from the bottom left as indicated in Figure 114 on page 237.

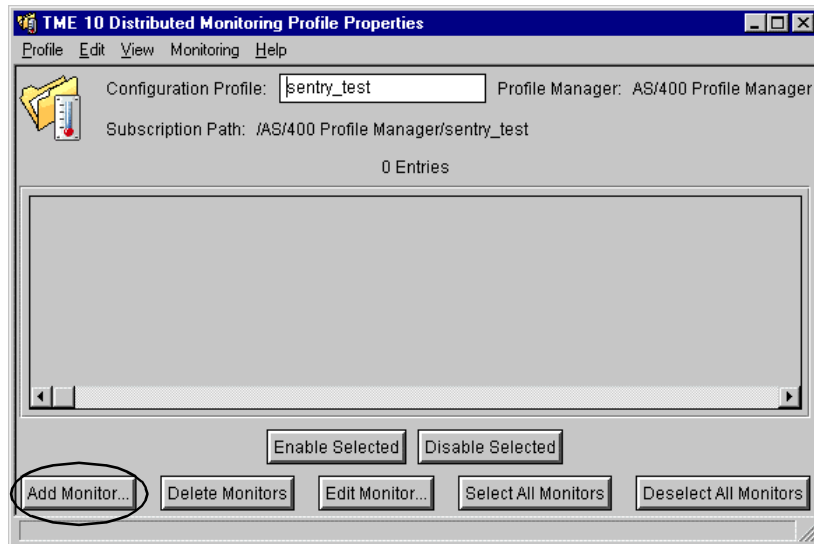


Figure 114. The Distributed Monitoring profile

4. You will be presented with a list of possible monitoring collections. Select a collection from the left panel, and a list of monitors from that collection will be displayed on the right panel. Select the monitor you wish to use, in this case, the *Pool size* monitor from the OS/400 Pool collection as shown in Figure 115 on page 238. You will be asked, at the bottom of the screen, for any arguments needed for the monitor. Refer to the *Distributed Monitoring User's Guide 3.6*, GC31-8382, for details on a particular monitor.

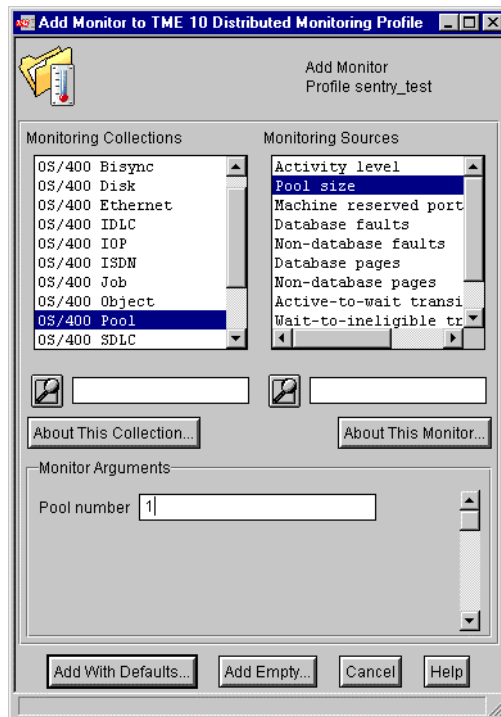


Figure 115. Selecting a monitoring collection and monitoring source

Select **Add Empty** to continue.

5. This is where you select the response level desired for the monitor. In this case, when the response level triggers the critical level, the response is to run a program on the monitored host. The run program entry is shown in Figure 116 on page 239.

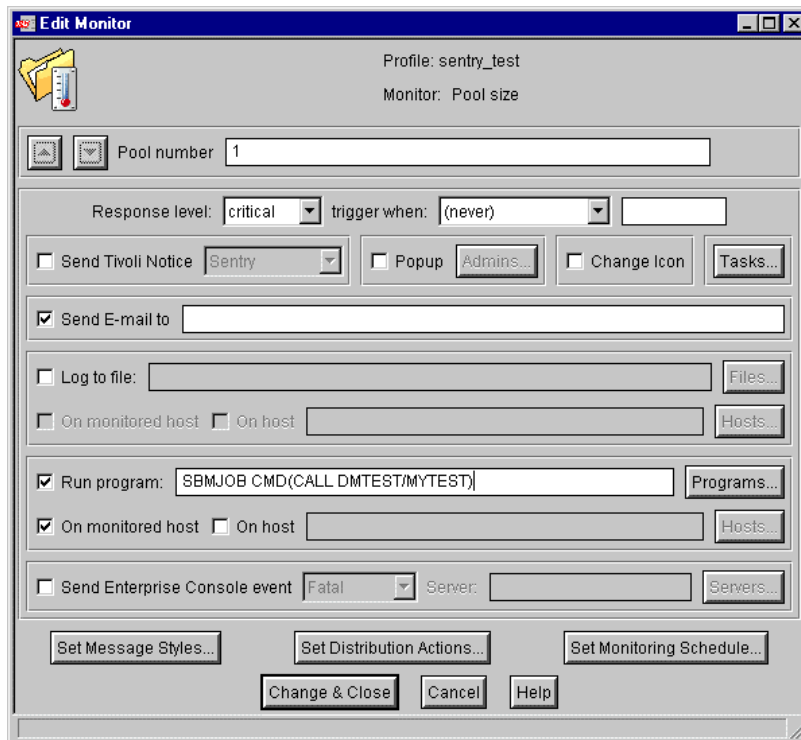


Figure 116. Configuring the Distributed Monitoring monitor

6. The next step is to select the **Set Monitoring Schedule** to set up the time that the monitor will first fire. After that, it will fire at regular intervals. The Set Monitoring Schedule is shown in Figure 117 on page 240.

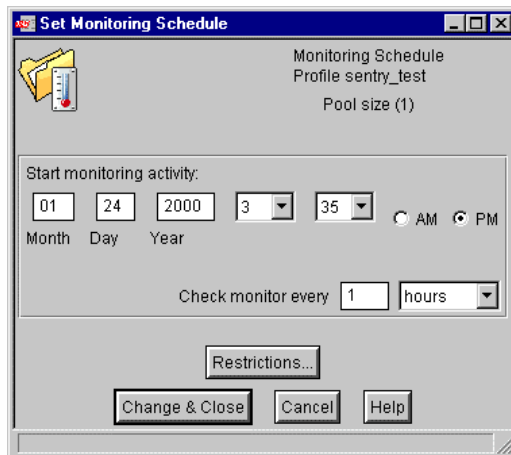


Figure 117. Setting monitoring schedule

7. Select **Change & Close** when done. Then select **Change & Close** from the Edit Monitor panel.

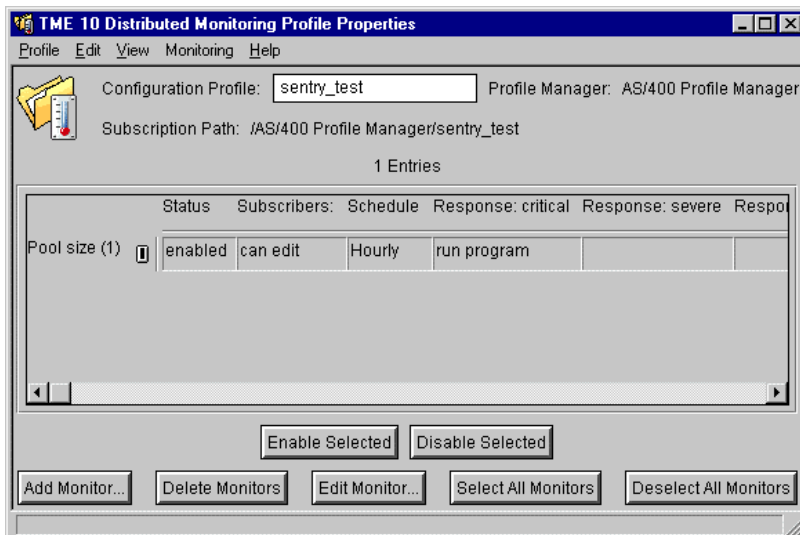


Figure 118. Saving configured Distributed Monitoring profile

Then select **Profile -> Close**. This will bring up the Confirm screen shown in Figure 119 on page 241. Select **Save & Exit** to return to the Profile Manager screen.

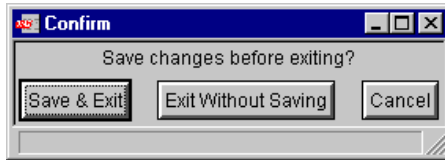


Figure 119. The confirmation screen for saving profile

8. Before you can distribute a profile to an endpoint, you must first make the endpoint a subscriber for this profile manager. From the Profile Manager screen, select **Profile Manager -> Subscribers**.

This brings you to the select subscribers dialog. From the right hand panel, select the subscribers (TMA Endpoints) to which monitors from this profile manager will be distributed by selecting each one and moving it to the left hand panel. When all TMA Endpoints have been selected, select the **Set Subscribers & Close** button shown in Figure 120.

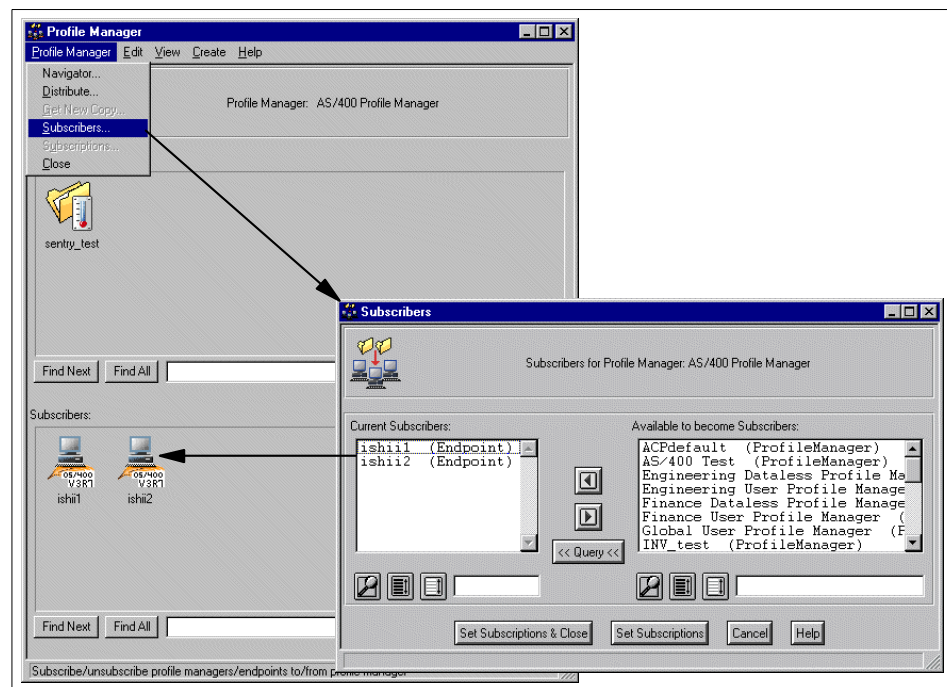


Figure 120. Setting subscribers for the monitors

This returns you to the profile manager screen and shows the subscribed endpoints. To do the actual distribution, you can drag and drop the

sentry_test icon to the TMA endpoint icon. This will move the profile copy from the profile manager to the DMEPENGINE on the AS/400 endpoint.

Using a Run Program response

If you use a Run Program response for an AS/400 Endpoint, you must first create the program for the monitor to run. You can use the following as a guide.

1. You may want to create a library in which to work. You can create one in QSYS (the default) by running CRTLIB (dmtest). This creates a library called dmtest.
2. Next, you will need to create a source physical file. You can issue the WRKPF command or, from the menu, select **4. Files, libraries, folders => 1. Files => 1. Work with files**. When the next screen appears, enter the following values for the File and Library:

File *ALL

Library DMTEST

Then, select **1. Create** (fill in QCLSRC DMTEST in the highlighted area). Locate Source Physical File in the list that follows and enter 1. The next step is to create/edit a CL program.

Note

In this process, QCLSRC can be named anything; an example is being followed for this description.

3. Then, select option **1. Programmer Menu**.
4. From here, select option **8. Edit a source file member**.
5. At this point, it will bring up the Source Entry Utility or SEU. If a program is already there from a previous session and you wish to create a new one, press **F3**.

Then, enter the information specific to your program:

Change/create member Y (Enter Yes)

Member MYTEST (Name of your program)

File QCLSRC (Use QCLSRC for this entry)

Library dmtest (Name of library you want to use)

Return to editing Y (Change to Yes)

Return to STRSEU by editing the last file, which should be your new file.

Note

In this process, if **Return to editing** is not checked, it will say *member created* and exit.

6. Testing and writing AS/400 CL programs is beyond the scope of this document, but, once you have created the CL program and compiled and tested it, you can configure a monitor response to run it as follows.

Check the Run program box and enter the following:

```
SBMJOB CMD (CALL DMTEST/MYTEST)
```

or

```
CALL DMTEST/MYTEST
```

Using CALL will run the program, but Distributed Monitoring will wait for the CL program to finish. Submitting the job using SBMJOB will cause the program to be disconnected from Distributed Monitoring. That is the AS/400 equivalent of running in the background.

In addition, you need to set the user and group ID for the profile containing this monitor if you need to run the program as other than QTIVUSER. The most common problem with getting either custom monitors or run programs to run correctly is that the user running the program does not have sufficient OS permissions to perform the actions needed.

We introduce more detailed information about running a CL program as a Run Program response in the next scenario.

7.2.5.3 Running a CL Program as a Run Program response

As mentioned previously, a Run Program can be any AS/400 command that you might enter at a command line, including the Call or Submit Job commands. In this scenario, we will actually show the running of a CL program using both of these in order to point out the fundamental differences between them. We also use these two examples to attempt to point out the common problems and debugging techniques mentioned in Section 7.2.4, “Troubleshooting Tivoli Distributed Monitoring for AS/400” on page 227.

For this example, we suppose that we are in on an AS/400 that is using the TEC AS/400 Message Adapter to monitor QSYSOPR message queue. At some point, we start this message adapter, give it the job name, LPVMSGADP, and run it as user LORRAINE who has *ALLOBJ special authority. We will set up an OS/400 job monitor that will run every 15 minutes to check to see if this job is active. If it is not active, we will run a CL program as a response that will restart the message adapter. It also sends a message

to the QSYSOPR message queue indicating that the Distributed Monitoring has restarted the adapter. In this manner, not only will the AS/400 system operator be notified that this was done, but the Tivoli Administrator will also be notified when the message adapter sends this message to TEC. They can then both take any appropriate action.

First, we create our CL response program by performing the following steps:

1. Use the Create Source Physical File command, `CRTSRCPF`, to create a source file in which to place the source for our CL program. As shown in Figure 121, we call our source file `DMCLSRC` and place it in a library we created called `DMTEST`.

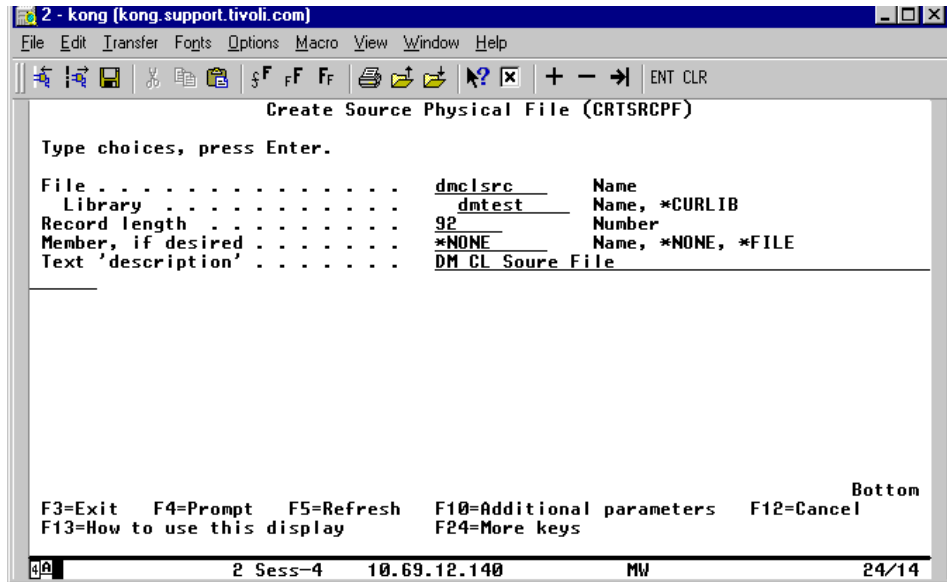


Figure 121. Create source physical file

2. We now issue the Work with Members using PDM command, `WRKMBRPDM`, as follows:

```
WRKMBRPDM FILE (DMTEST/DMCLSRC)
```

to access our `DMCLSRC` file using the Programming Development Manager (PDM) shown in Figure 122 on page 245.

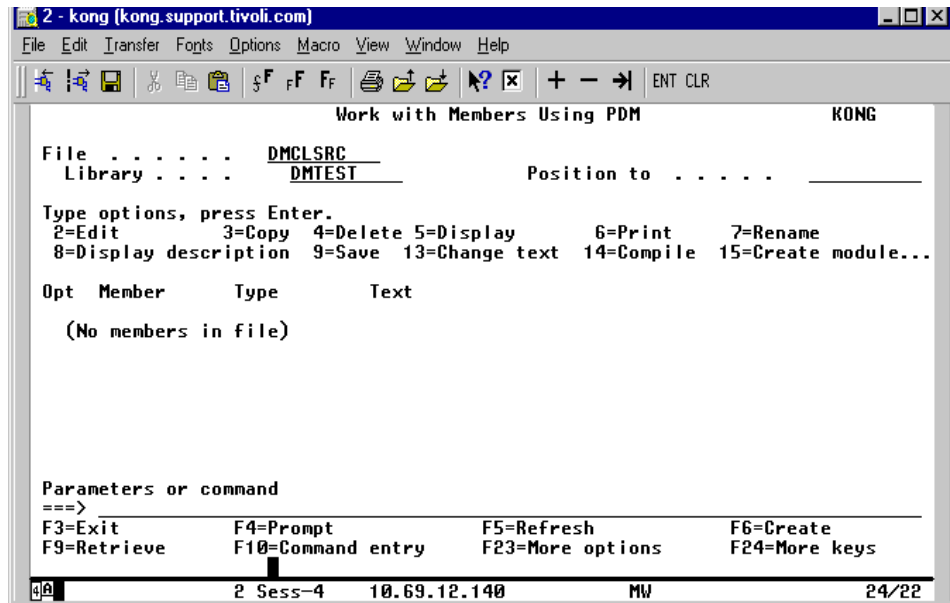


Figure 122. Work with members using PDM

3. On this screen, Press **F6** (Create) to create a member in this file. This will bring up the Start Source Entry Utility command, *STRSEUW*, and, here, we name the source file member, provide the source file type, and type a description as shown in Figure 123 on page 246.

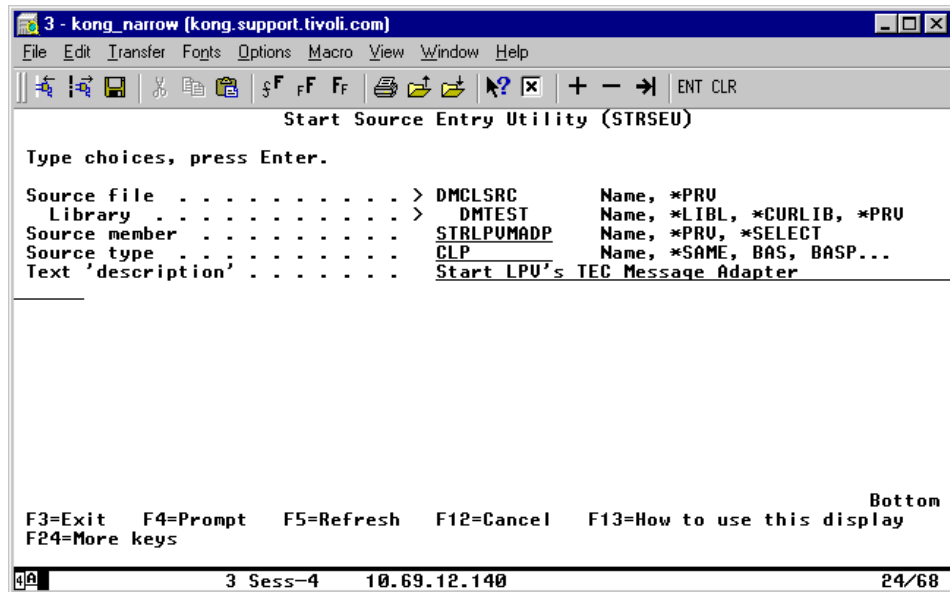


Figure 123. Start source entry utility

4. Pressing Enter puts us into the Source Entry Utility Edit panel where we enter the CL source program shown in Figure 124 on page 247.

The screenshot shows a Tivoli Enterprise Console window titled "3 - kong_narrow (kong.support.tivoli.com)". The window has a menu bar (File, Edit, Transfer, Fonts, Options, Macro, View, Window, Help) and a toolbar with various icons. The main area displays the code for the STRLPVMADP program. The code is organized into columns, with column numbers 1 through 7 indicated at the top. The code includes comments and program logic for handling LPUMSGADP events. At the bottom, there are function key definitions: F13=Change session defaults, F14=Find/Change options, F15=Browse/Copy options, and F24=More keys. The status bar at the bottom shows "3 Sess-4", "10.69.12.140", and "24/15".

```

Columns . . . : 1 71 Edit DMTEST/DMCLSRC
SEU=> STRLPVMADP
FMT ** ..... 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6 ..... 7
***** Beginning of data *****
0001.00 /* This program will be submitted by DM in the event that */
0002.00 /* it detects that the LPUMSGADP (LPU's TEC Message Adapter) */
0003.00 /* has ended. It will also send a message to QSYSOPR message */
0004.00 /* queue indicating that DM detected that LPUMSGADP had to be */
0005.00 /* restarted. */
0006.00 PGM
0007.00          STRTECADP  EUTADP(LPUMSGADP) +
0008.00                      CFGFILE('/qsys.lib/lpvlib.lib/qaymacfg.file+
0009.00                      /qaymacfg.mbr')
0010.00          SNDMSG    MSG('DM restarted LPUMSGADP job.') +
0011.00                      TOUSR(QSYSOPR)
***** End of data *****

F13=Change session defaults  F14=Find/Change options
F15=Browse/Copy options      F24=More keys

```

3 Sess-4 10.69.12.140 24/15

Figure 124. Sample STRLPVMADP program

- When finished, press **F3** (Exit). On the Exit panel, be sure to select **Y** at the Change/Create member option.

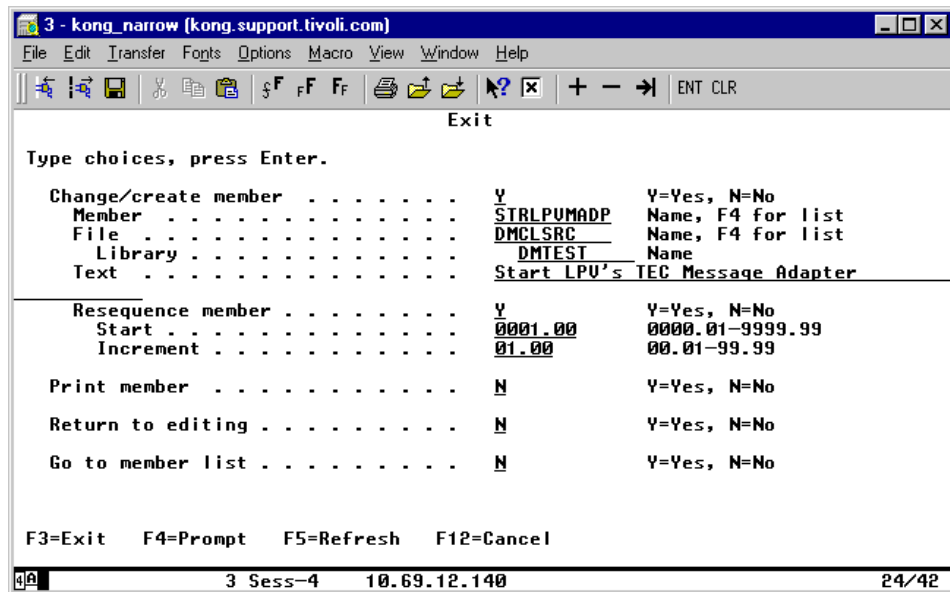


Figure 125. SEU exit panel

Pressing Enter puts us back on the Work with Members screen. Notice that the source member, STRLPVMADP, we just created is now listed.

6. Compile the program by selecting Option 14 (Compile) as shown in Figure 126 on page 249.

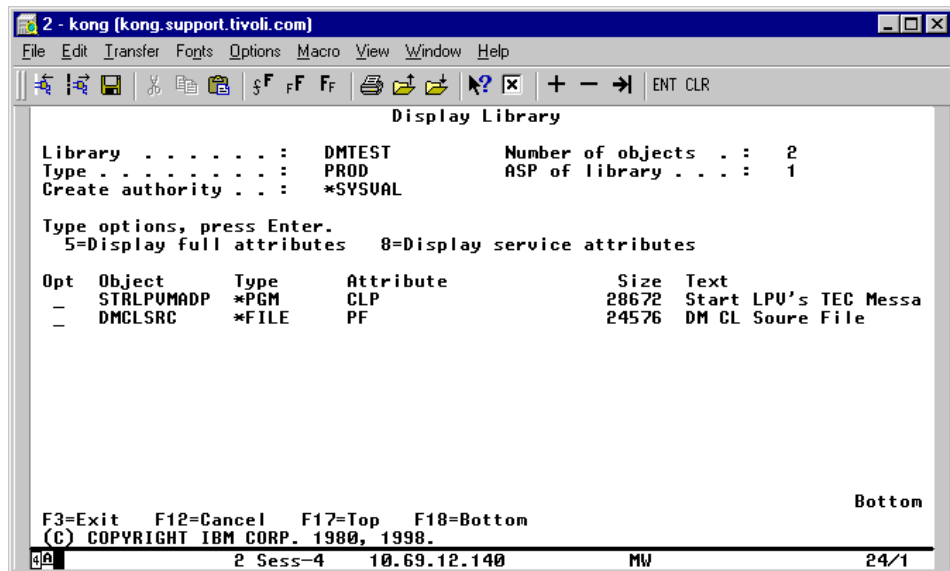


Figure 127. Display library

Note

It has not been our intention here to provide much detail in writing CL programs or in using Programming Development Manager. In fact, we strongly suggest you refer to the *AS/400 Programming Guides* and PDM documentation for more complete examples and explanation of CL programming techniques and practices.

PDM and SEU are part of the AS/400 Application Development Tools license program.

Now that we have successfully created our program, we wish to test it briefly before attempting to use it as a Run Program response. In order to do this, we first use the Work with Active Jobs command, `WRKACTJOB`, to see if the `LPVMSGADP` job is running.

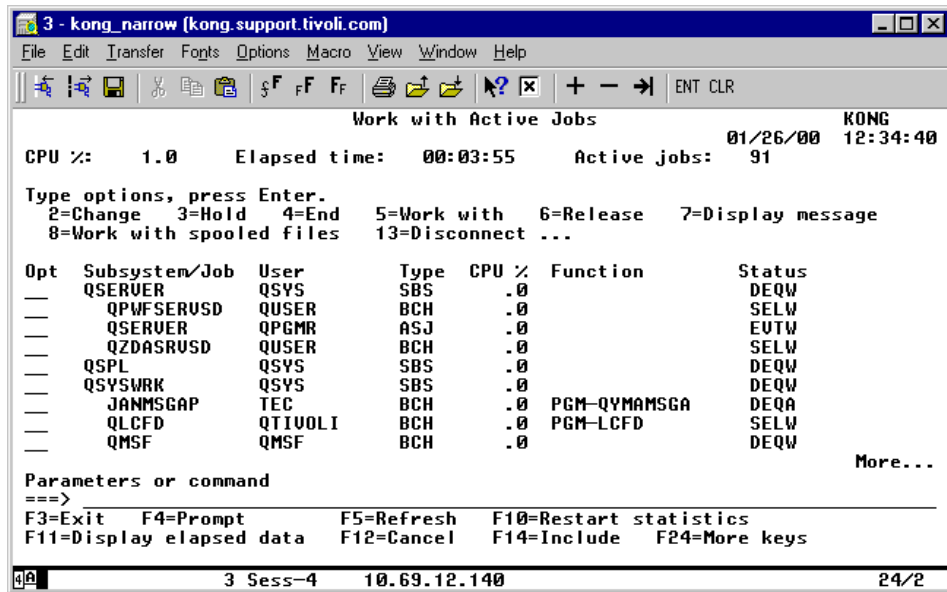


Figure 129. Work with Active Jobs: LPVMSGADP not running

As shown here, the LPVMSGADP job is no longer running; so, we call our program using the Call Program command as follows:

```
CALL PGM(DMTEST/STRLPVMADP)
```

Again, since we issued this command at the command line on the Work with Active Jobs screen, we can now simply press F5 (Refresh) to verify that the LPVMSGADP job becomes active. See Figure 130 on page 253.

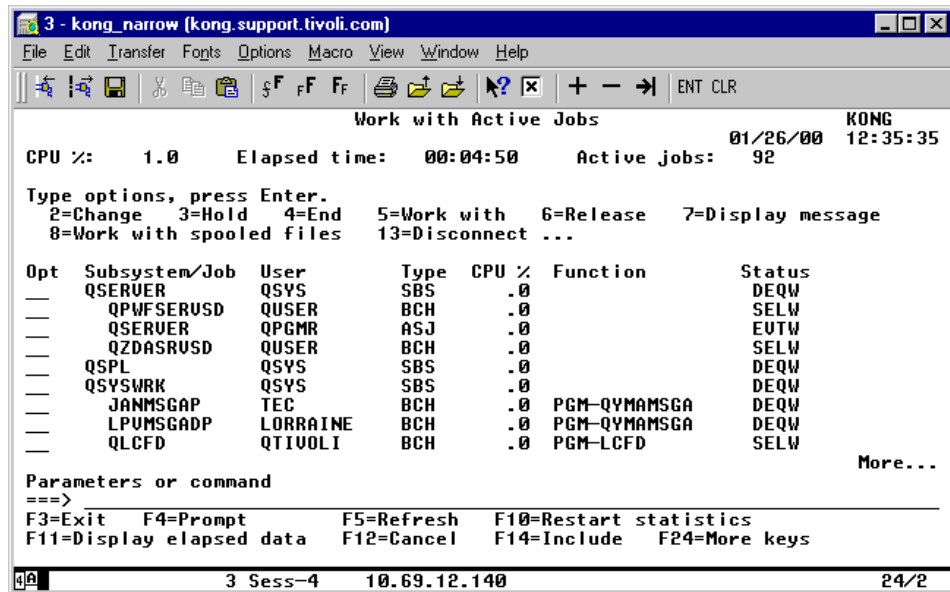


Figure 130. Work with Active Jobs: LPVMSGADP job restarted

Finally, we use the Display Messages command, `DSPMSG`, to verify that the Send Message command, `SNDMSG`, within our program sent the message to the QSYSOPR message queue as appropriate:

```
DSPMSG MSGQ(QSYSOPR)
```

This brings up the QSYSOPR message queue where we verify that the message was sent.

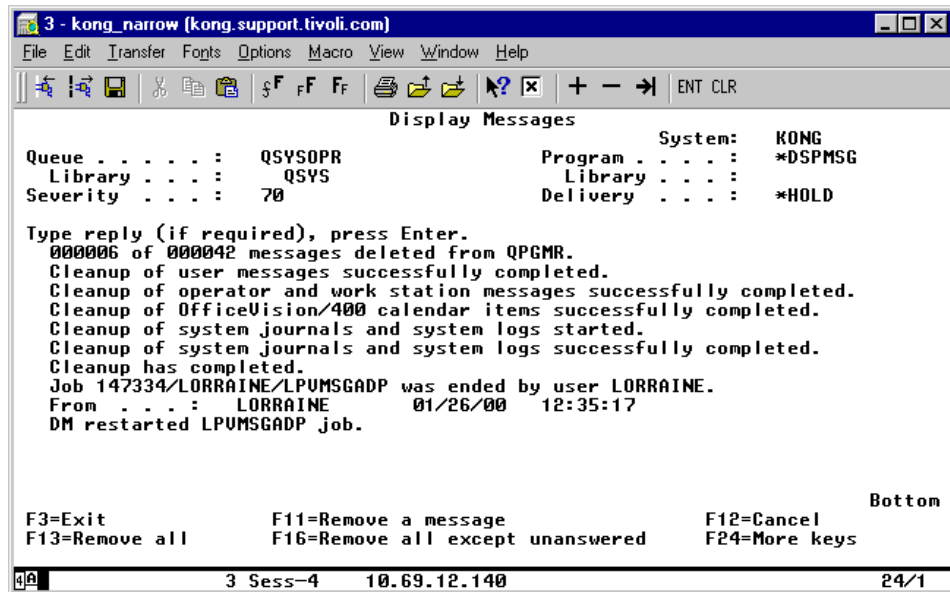


Figure 131. Display QSYSOPR messages

Now that we have tested the program and know that it works well from the AS/400 command line, we will create our Distributed Monitoring monitor.

Note

In this scenario, we will intentionally create an error. We do this to illustrate some of the troubleshooting techniques discussed in Section 7.2.4, “Troubleshooting Tivoli Distributed Monitoring for AS/400” on page 227.

To create our job monitor, we make the selections shown in our Distributed Monitoring profile shown in Figure 132 on page 255.

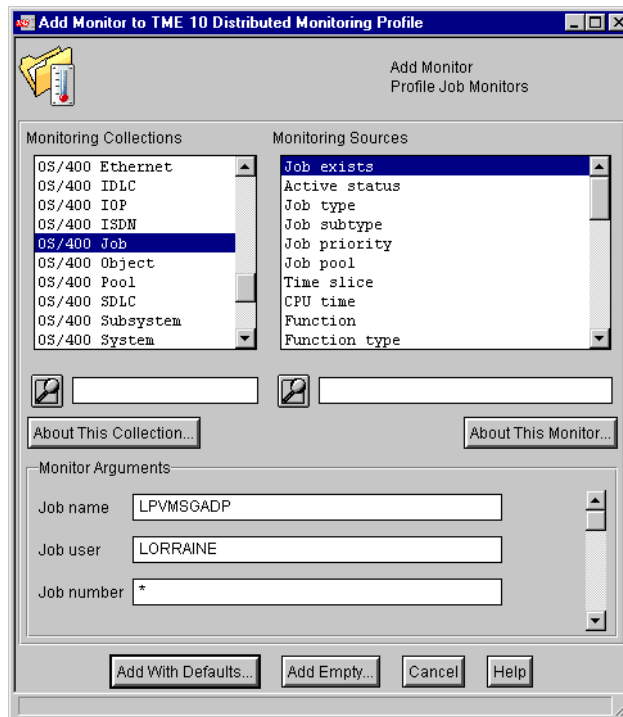


Figure 132. OS/400 Job exists monitor

Notice that we have entered a wildcard symbol, * (an asterisk), for the Job number value because this job number is assigned by OS/400 at the time the job begins, and, in this case, we will not necessarily know it. If necessary, you may also specify a wildcard in the Job user value, but we always want our LPVMSGADP job to run as the specified user.

After completing this panel, we press **Add Empty**.

This brings up the next panel on which we can define our desired response levels and the actions taken for each.

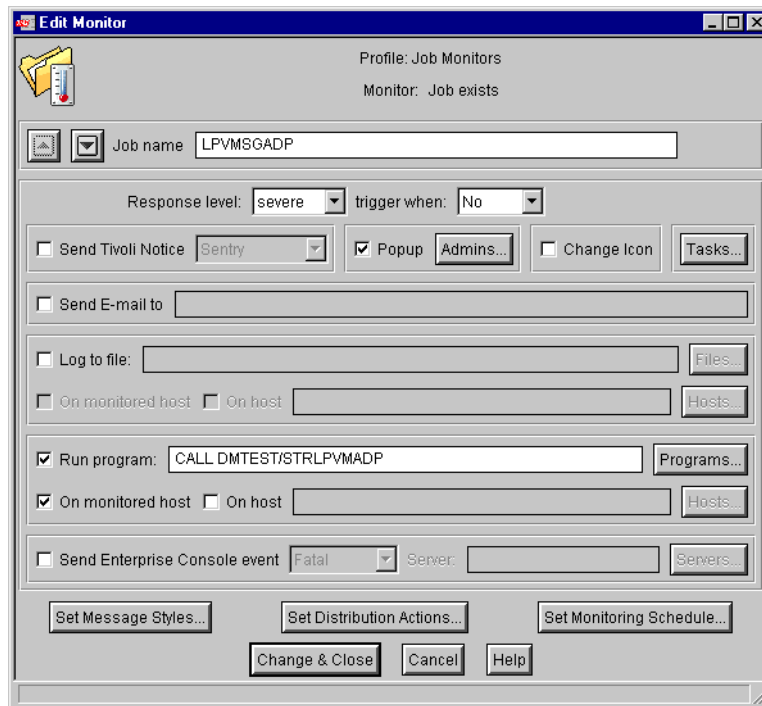


Figure 133. Job monitor responses

As shown, we have defined a severe response that, in the event our job monitor returns a No (indicating that the LPVMSGADP job is not active), will:

1. Create a Popup on our Tivoli Desktop.
2. Select our Administrator using the **Admins** button.
3. Run the following command on the AS/400 Endpoint:

```
CALL DMTEST/STRLPVMADP
```

We then set the monitoring schedule to run the monitor as desired, select **Change & Close**, and distribute the profile to our AS/400 Endpoint.

We now return to the AS/400 to end the LPVMSGADP job, which is still running, as shown in the Work with Active Jobs screen in Figure 134 on page 257.

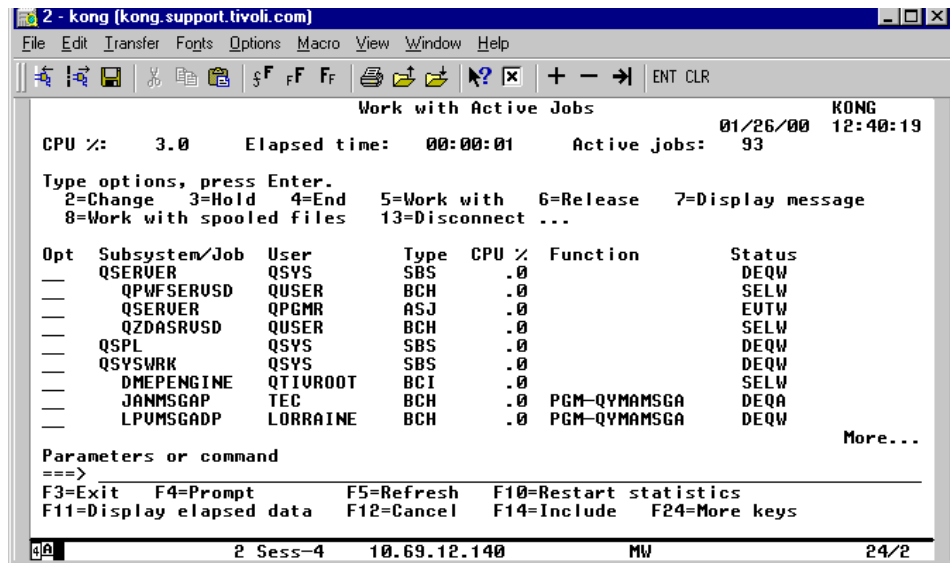


Figure 134. Work with Active Jobs: LPVMSGADP job running

To end this TEC AS/400 Message Adapter job properly, we issue the End TEC Event Adapter command:

```
ENDTECADP EVTADP (LPVMSGADP) OPTION(*IMMED)
```

The next time our job monitor runs, we receive the following pop-up on our Tivoli Desktop.

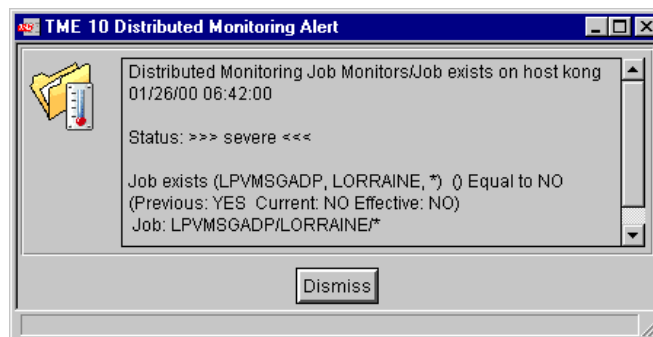


Figure 135. Job monitor popup response

Despite successfully receiving our popup message, we still do not see the LPVMSGADP job running. We press F5 (Refresh) several times on the Work with Active Jobs screen, but the LPVMSGADP job never starts. It appears as

though our Run Response program did not run. Actually, as shown in the following DMEPENGINE joblog in Figure 136, our Run Response program really did run.

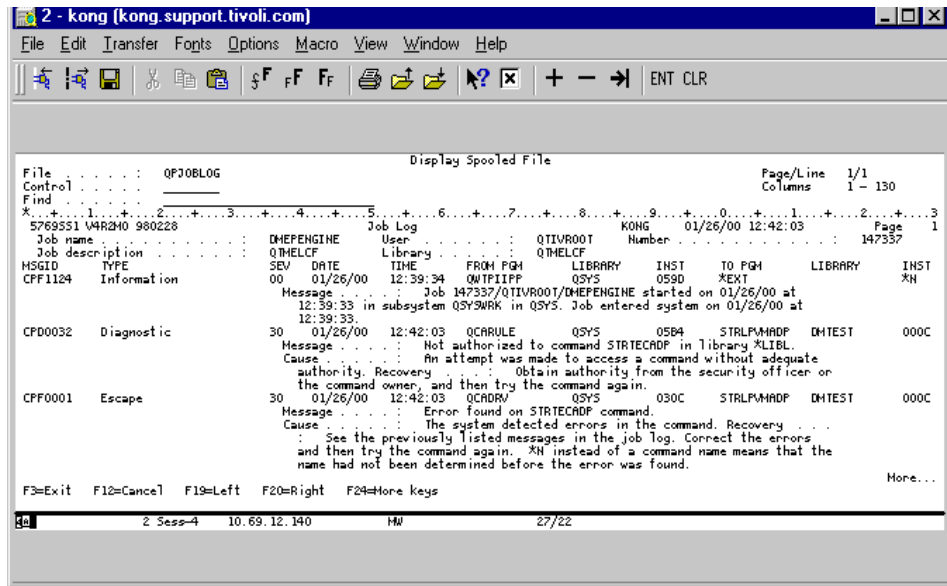


Figure 136. DMEPENGINE joblog

However, it failed to start the LPVMSGADP job due to an authority issue. Notice the CPD0032 error:

"Not authorized to command STRTECADP in library *LIBL."

Recall that, by default, the Remote User ID specifies *nobody*. As a result, when the Run program response was run, the DMEPENGINE job swapped to QTIVUSER under the covers and then attempted to issue this command. However, QTIVUSER has no special authority and has not been specifically authorized to the necessary objects to issue this command.

We neglected to set the Remote User ID field to an appropriate OS/400 user profile. We do this now as shown in Figure 137 on page 259.

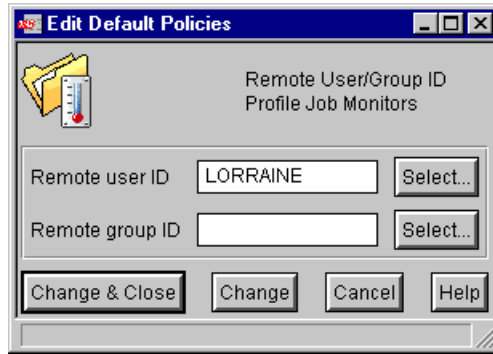


Figure 137. Edit default policies: Set Remote User ID

We set this to LORRAINE because this is the user profile that the job usually runs and so we can be assured that it has the necessary authority to run this program. Now, before issuing the `CALL` command, the DMEPENGINE will swap to this user profile. This is also very important because, when we created the Monitor, we gave it the job's user name of LORRAINE; so, we must assure that this is also the restarted job's user name so that our job monitor will recognize this as matching our monitor values and not continue to run our response program at each specified interval.

Recall that, earlier, in Section 7.2.3, "Customizing Tivoli Distributed Monitoring for AS/400" on page 226, we recommended using the Submit Job command, `SBMJOB`, over the Call Program command, `CALL`. In Figure 138 on page 260, we illustrate running this same monitor now using the `SBMJOB` command.

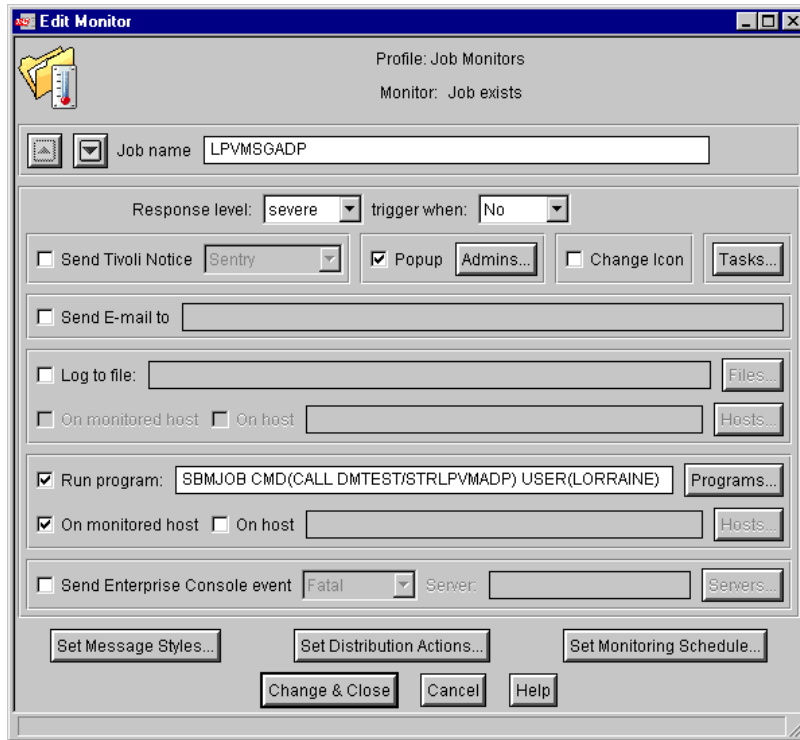


Figure 138. Editing the job exists monitor

First, notice that the Submit Job command allows us to specify which user the submitted job will run as. This is specified in the `USER(LORRAINE)` parameter. This will now be the user that runs the `CALL DMTEST/STRLPVMADP` command, regardless of the Remote User ID specification.

Note

The Remote User ID must still be authorized to issue the Submit Job command as entered here, that is, they will become the user profiles that issue the Submit Job command. It is the new job that results from the Submit Job command that will run as the user profile specified in the `USER` parameter. This new job is the job that will actually be running the `CALL` command. If necessary, refer to Section 7.2.4, “Troubleshooting Tivoli Distributed Monitoring for AS/400” on page 227, again for complete details of authority checking with the Run Program response.

After updating the monitor as shown, we press **Change & Close** and distribute the profile to the AS/400 Endpoint. Since our LPVMSGADP job was never

restarted, at the first running of our monitor, we again receive the popup shown in Figure 139.

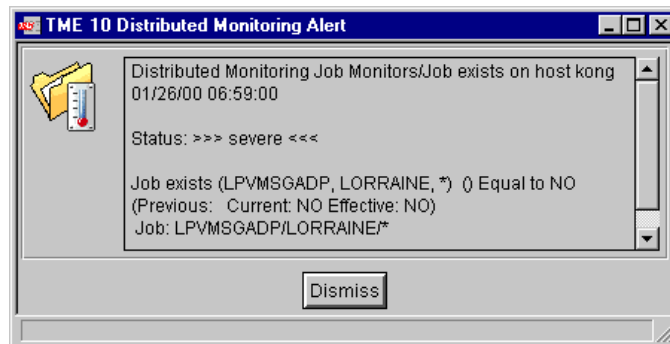


Figure 139. PopUp job monitor response

On the AS/400, we again press F5 (Refresh) on the Work with Active Jobs screen shown in Figure 140.

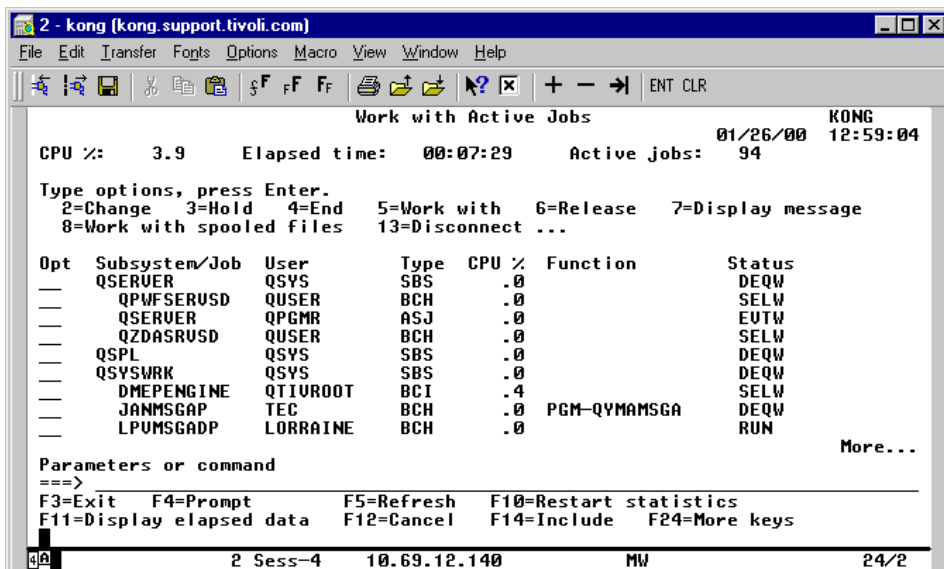


Figure 140. Work with active jobs: LPVMSGADP job restarting

Clearly, we see that the LPVMSGADP job is restarting as we desired. Let us stop for a moment and consider what we would do in the event this did not work. Just as we did previously, we would begin by examining the DMEENGINE joblog.

If the DMEPENGINE is an active job, you can access its joblog from this same Work with Active Jobs screen by performing the following steps:

- Select **5** (Work with) beside it.
- On the resulting Work with Job panel, select option 10, “Display job log, if active or on job queue”.

Once in the joblog, we suggest immediately pressing F10 (Display Detailed Messages) and F17 (Top). Doing this will ensure that you see all possible messages starting when the job began.

Note

For completeness, we mention here that the messages actually getting placed in a joblog are also controlled by the job’s message log values. There are several ways in which these values can be set and adjusted. For instance, they can be specified as a Submit Job parameter. For complete details of job processing, refer to the *AS/400 Programming Work Management Guide*, SC21-8078.

Following this procedure on the DMEPENGINE job on the Work with Active Jobs screen yields the following joblog messages shown in Figure 141.

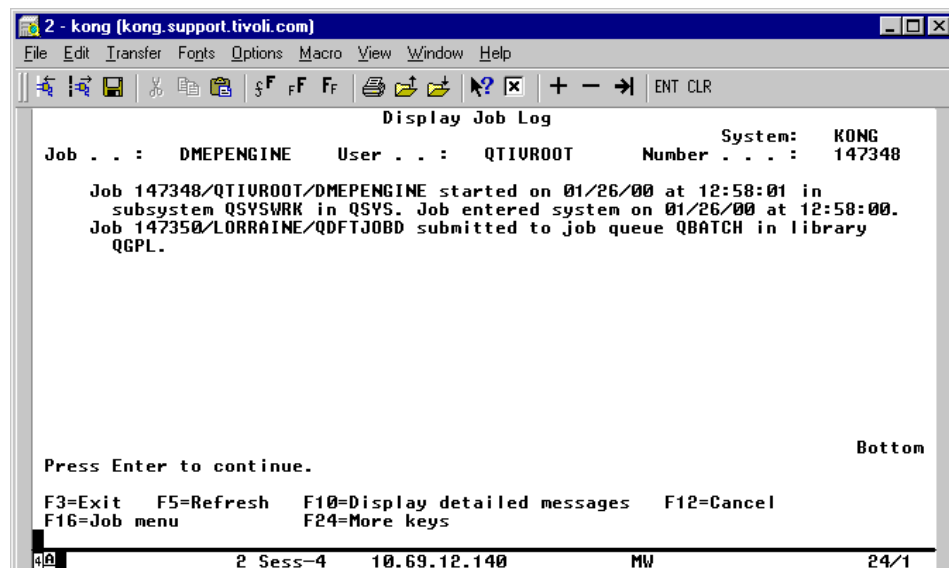


Figure 141. DMEPENGINE joblog showing SBMJOB response

Notice that when the DMEPENGINE job issued the Submit Job command, a message that provides the fully-qualified job name (Job, User, and Number) is provided in its joblog. The qualified job name for the Run Program response just performed is:

Job QDFTJOB
User LORRAINE
Number 147350

Had our Run Program not have performed as expected, we could use this information with the Work with Jobs command, WRKJOB:

WRKJOB JOB(147350/LORRAINE/QDFTJOB)

This would bring up the following Work with Job panel shown in Figure 142.

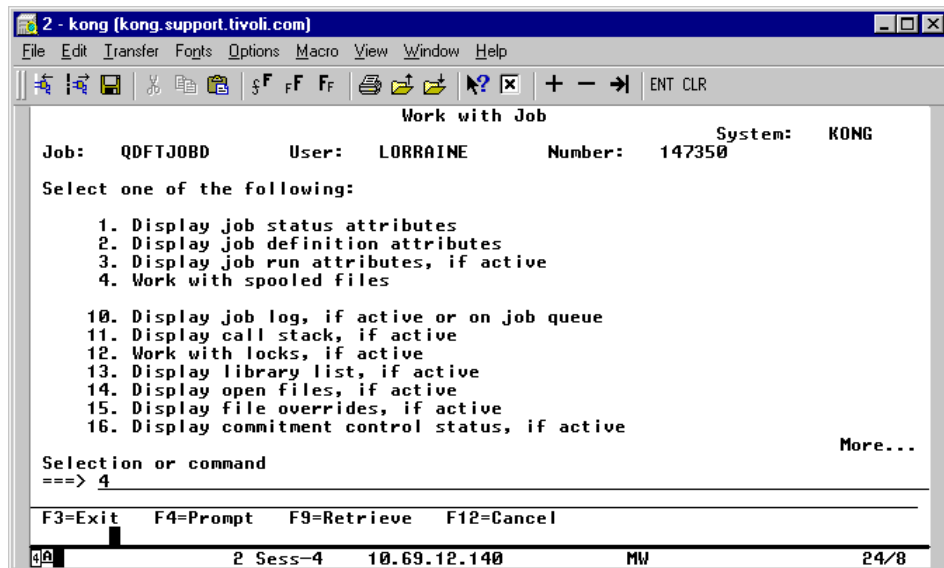


Figure 142. The work with job panel

Notice that the qualified job name we specified appears across the top of the screen. From this panel, we would select option 10 to see the joblog if the job is still active or option 4 to work with spooled files to see if the job produced a printed joblog upon completion. Since, in this case, our job completed normally, we will not examine the joblog.

7.2.5.4 Using environment variables in a Run program response

For the following scenario, let us suppose we have created a separate subsystem in which a particular application's programs are to run. We created the subsystem specifying that it will only allow seven jobs active at a time. The normal expected number of jobs to be active at one time is about four. More jobs than this may create resource contention; so, we want to have the AS/400 system operator notified when the number of jobs exceeds four. We will do this by sending a message to the system operator's message queue. We would like the message to include the number of jobs that were detected.

We create our Subsystem Monitor as shown in Figure 143.

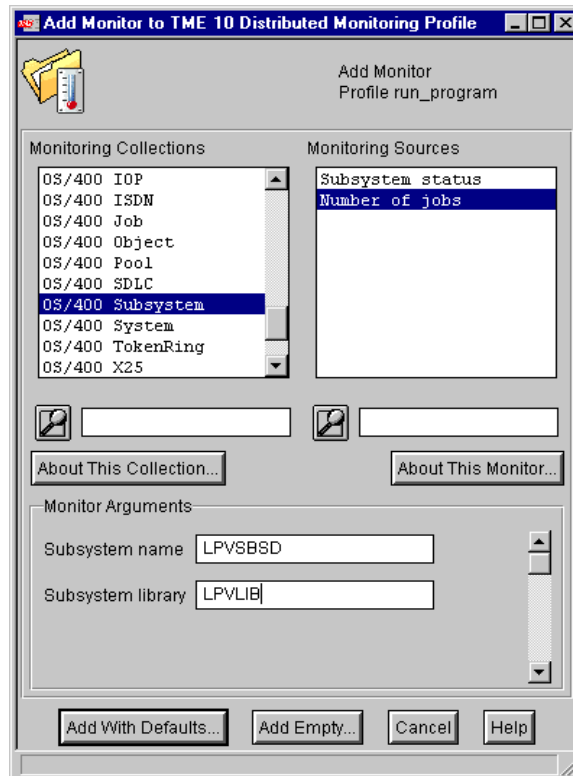


Figure 143. Creating the OS/400 subsystem monitor

For our example, the subsystem is named LPVSBSD and is located in the LPVLIB library. We set these as the Monitor Arguments.

We select **Add Empty** button, and then set a critical response as defined here. See Figure 144 on page 265.

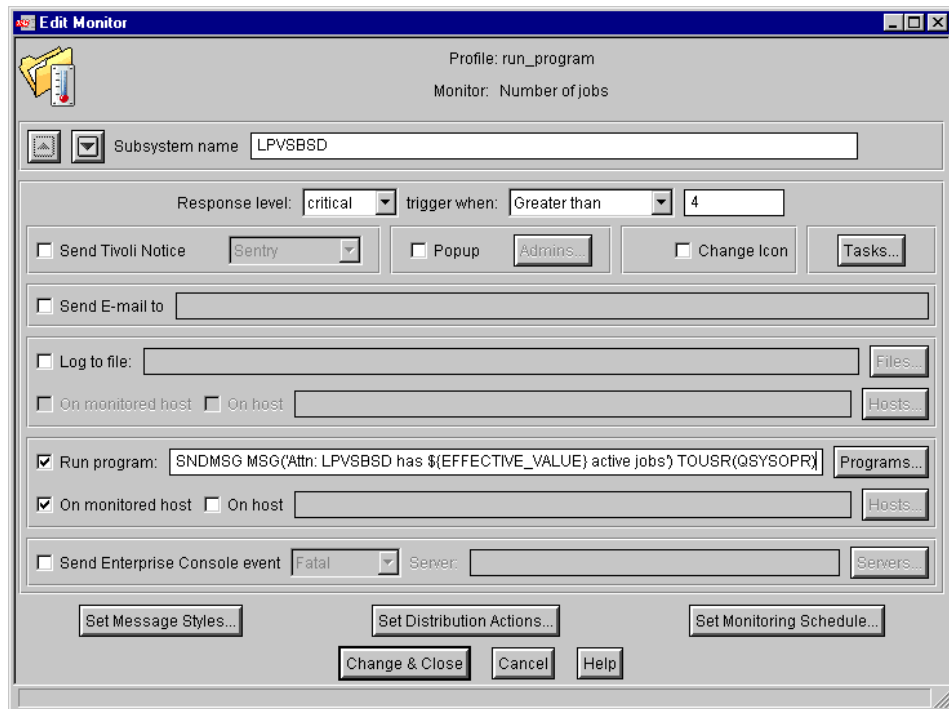


Figure 144. Editing the number of jobs monitor

Depending on how the Run Program responds, we will run the Send Message command, `SNDMSG`, with the following MSG parameter:

```
'Attn: LPVSBSD has ${EFFECTIVE_VALUE} active jobs'
```

Notice that the environment variable is the same as it normally would be for other Endpoints.

Again, we set our desired monitor schedule; select **Change & Close** and distribute the profile to the AS/400 Endpoint.

As illustrated in Figure 145 on page 266, on the AS/400, we have started the LPVSBSD subsystem and have started six jobs in it.

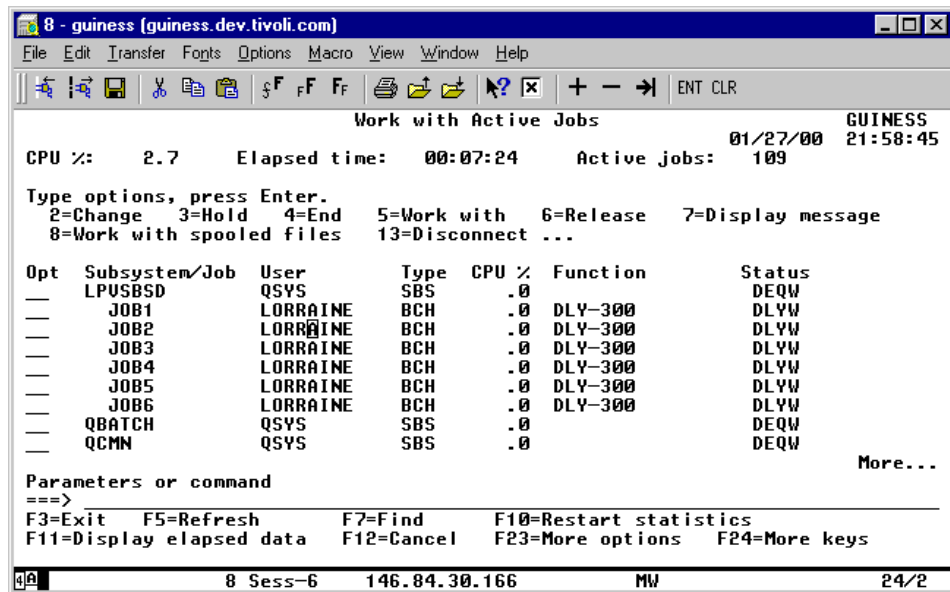


Figure 145. Work with Active Jobs

Following this, the next time our monitor runs, we will use the Display Message command, DSPMSG:

DSPMSG MSGQ(QSYSOPR)

This will bring up QSYSOPR's message queue shown in Figure 146 on page 267.

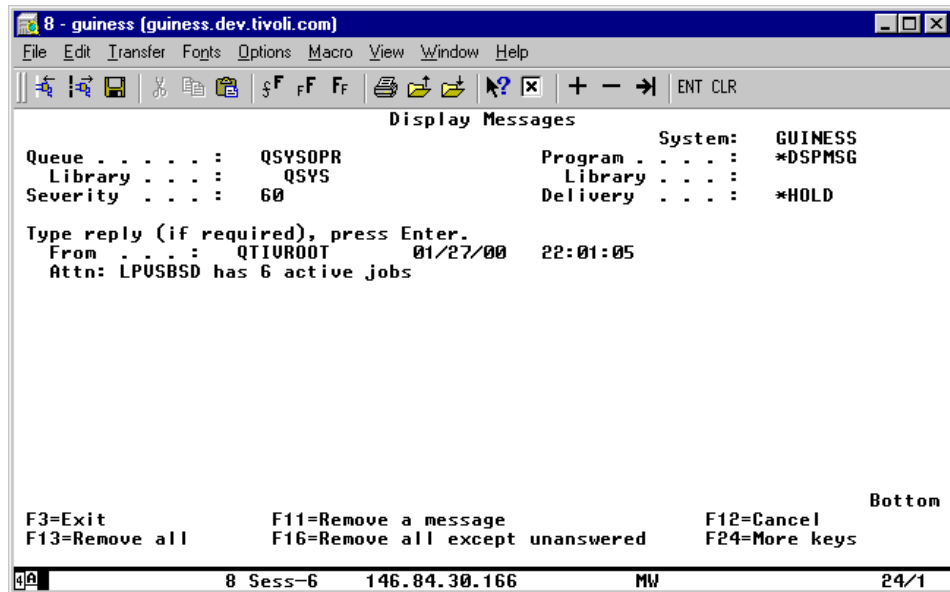


Figure 146. Display Messages

We observe that our Run Program response has successfully sent the message to the QSYSOPR message queue and resolved the environment variable appropriately.

7.2.5.5 Creating custom monitor for AS/400 Endpoint

This section introduces an example of creating a custom monitor for AS/400 endpoints.

1. To do a custom monitor instead of selecting a monitor from an AS/400 monitoring collection, you would select, from the Universal Monitoring collection, either a String script or a Numeric script (depending upon the value returned by the custom program) and give the monitor arguments in the same manner that you use to specify a run Program above. See Figure 147 on page 268.

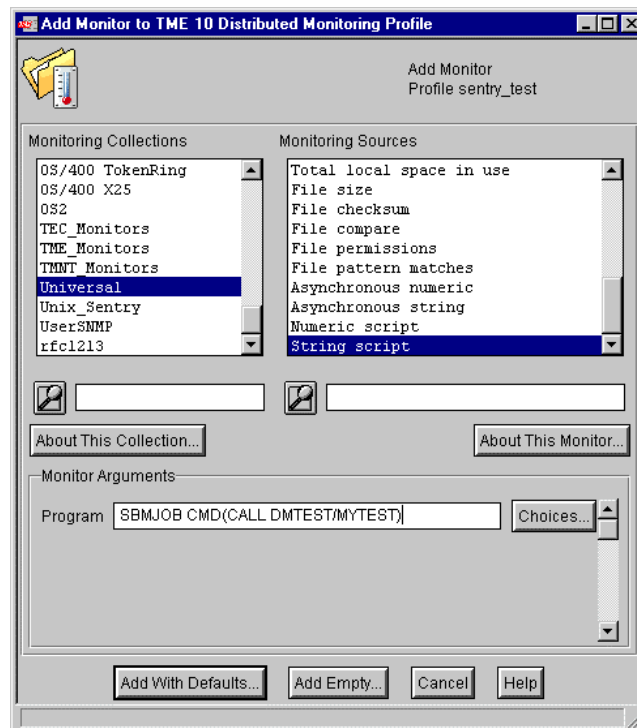


Figure 147. Selecting Universal monitoring collection

You would then select responses as normal.

2. In this case, the response is to send an event to the TEC server with the string returned from the program Equal to PASS as shown in Figure 148 on page 269.

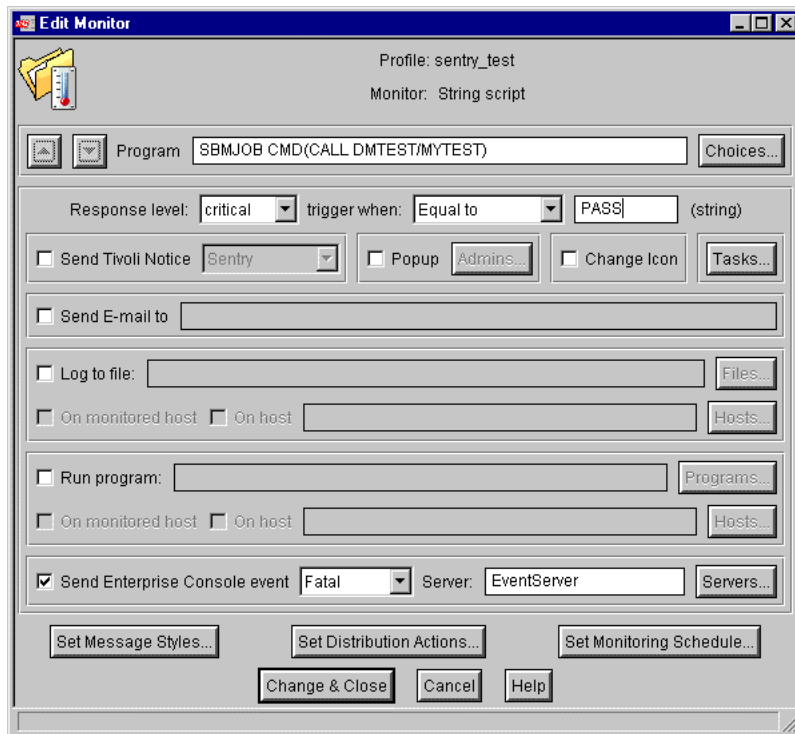


Figure 148. Configuring custom monitor

3. The last step for both a run program response or a custom monitor is to set the user and group ID to the proper user/group for your system. This user/group must exist on the AS/400 machine.

From the Profile Properties screen (this is set per profile, not per profile manager), select **Edit => Set User & Group ID** as shown in Figure 149 on page 270.

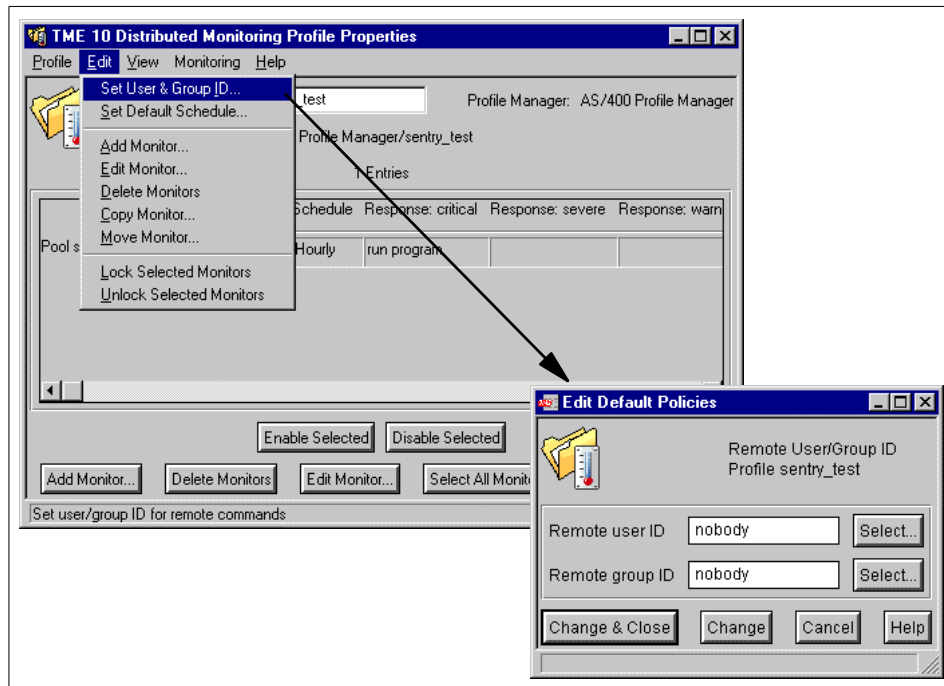


Figure 149. Setting user and group ID

On this panel, change the default user/group from nobody to the proper setting, and then select **Change & Close**. Then, you can distribute the monitor to the AS/400 endpoint subscribers.

Note

We have developed some examples of Distributed Monitoring custom script for AS/400. Refer to Appendix A, “Custom scripts for AS/400 Distributed Monitoring” on page 503, for more detailed information about Distributed Monitoring custom monitor for AS/400.

7.3 TEC AS/400 Event Adapters

TEC provides two event adapters that run on the AS/400:

AS/400 Message Adapter

This is detailed in Chapter 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507.

AS/400 Alert Adapter

Detailed in Chapter 3 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507.

As their names imply, these AS/400 adapters can forward OS/400 messages and alerts from an AS/400 to a Tivoli Enterprise Console event server. A few of the benefits this provides include:

- Consolidation of AS/400 messages and/or alerts from multiple AS/400s in the enterprise.
- Customized filtering of messages and/or alerts so that unimportant events are not forwarded allowing the TEC event server and its operators to identify and handle critical events quickly.
- Automation of actions on events using rules and tasks.

Note

Both of these AS/400 adapters are non-TME Adapters. Therefore, they do not communicate with the TMR server via an Endpoint Gateway; so, they do not require that the AS/400 Endpoint product be installed on the AS/400. Figure 150 on page 272 shows the difference between TME events and non-TME events.

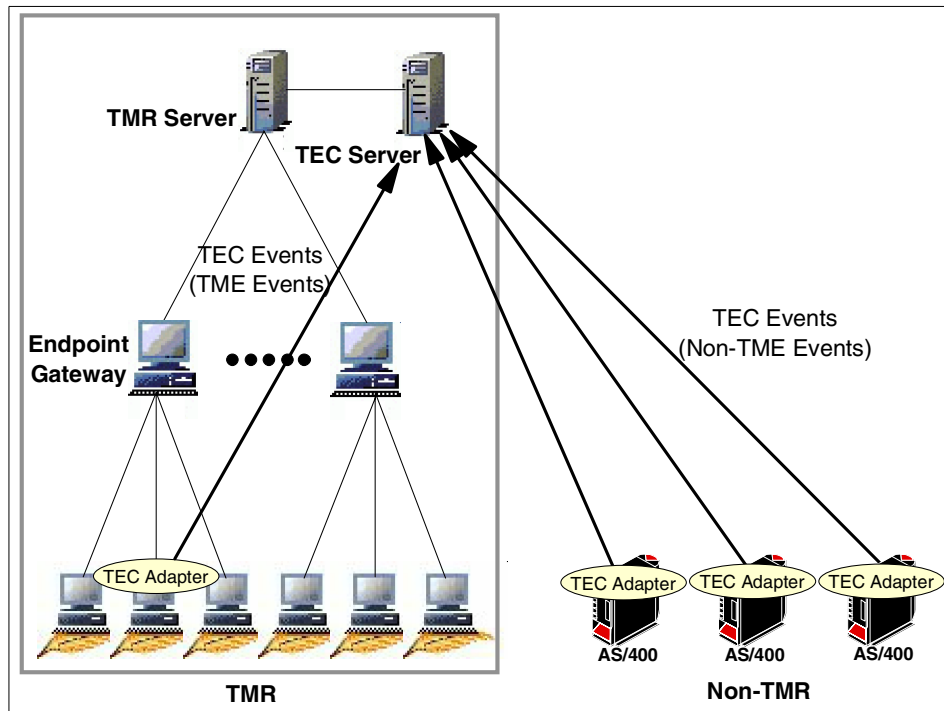


Figure 150. TME events and non-TME events

7.3.1 Installing the TEC AS/400 Event Adapters

Before installing the TEC AS/400 Event Adapters, you must be running a supported version of OS/400. You must also have the OS/400 PTFs listed in the “Software Requirements” section of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, installed or superseded. Refer to the “Documentation Changes” section of the product documentation, *Tivoli Enterprise Console 3.6.1 Release Notes*, GI10-8020, for an updated version of this chart.

Additionally, we recommend installing the Application Development ToolSet for AS/400. While this is not a prerequisite in a strict sense, this product provides the Source Entry Utility (SEU) component, which we recommend for editing the TEC Adapter configuration and CDS files on the AS/400.

Note

OS/400 provides another file editor, called the AS/400 Data File Utility (DFU), that can also be used. This can be used in place of SEU if necessary. If you have several AS/400s but SEU is available on only one of them, you can edit the Adapter files on the machine with SEU and then move the file to the appropriate AS/400. You can use various utilities to move these files between the AS/400 including FTP and OS/400 Save/Restore operations. In a large AS/400 environment, you may want to consider using Tivoli Software Distribution or the Object Packaging Utility to distribute these files as needed.

You can also use FTP to move the Adapter files to another machine type where you can edit them and FTP them back.

Refer to Chapter 5, “AS/400 Endpoint considerations and tips” on page 99, for several example of using FTP on the AS/400.

Installation of the TEC AS/400 Adapters is covered in detail in the “Installing AS/400 Adapters” section, in Chapter 2 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507. Unlike most TEC Adapters, these are not installed using an adapter configuration profile (ACP). Rather, the installation process involves using either of the following methods:

- Using FTP to move the appropriate savefiles from the TEC Event Server to the AS/400 and then running the Restore License Program command, `RSTLICPGM`, on each of these files to install EIF, the Alert Adapter, and/or the Message Adapter as desired
- Running the appropriate Restore License Program commands with the Tivoli Enterprise Console CD-ROM loaded onto the AS/400 CD-ROM device

The “Installing AS/400 Adapters” section, in Chapter 2 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507, contains complete detailed steps of these methods. Also, the “Uninstalling AS/400 Adapters” section in that same chapter provides complete instructions for removing these License Programs.

7.3.2 Understanding the TEC AS/400 Event Adapters

In order to understanding the TEC AS/400 Event Adapters, knowledge of OS/400 Message Support and OS/400 Alert Support is necessary. While we briefly attempt to discuss the details necessary for an understanding of these adapters, this will in no way be a complete treatment of either topic. Refer to

the books, *AS/400 System Operation V3.6.0*, SC41-4203, and the *OS/400 Alerts Support V3.2.0*, SC41-3413, for further details.

7.3.2.1 OS/400 Message Support

Messages are used as a means of communication between applications, processes, and users. In most cases, AS/400 messages are predefined in OS/400 Message File objects. Messages contain attributes including:

Message ID

The Message ID is a seven digit identifier in the form of *pppnnnn* where:

- *ppp* is a code beginning with one character followed by two alphanumeric digits
- *nnnn* is a four-digit hex numeric value

Severity

The Severity attribute is a value ranging from 00 to 99 indicating the severity of the condition in which this message would be issued. The higher the value, the more severe the condition.

Message Text

Message Text is the character string that is the text of the message.

Field Data

Field Data is the substitution variable fields whose value will be supplied at the time the message is sent.

The Operating System, as well as most AS/400 applications, supply message files containing the messages they issue. The TEC Adapters themselves, in fact, supply the following three message files:

1. QYMAMSGF in library QTMETEA
Contains the EIF-specific messages
2. QYAAMSGF in library QTMEYAA
Contains the AS/400 Alert Adapter specific messages
3. QYMAMMSGF in library QTMEYMA
Contains the AS/400 Message Adapter specific messages

Some of the TEC messages will be examined in Section 7.3.4, “Troubleshooting TEC AS/400 Event Adapters” on page 313.

The screen shown in Figure 151 lists the messages contained in the OS/400 QCPFMSG message file as shown by using the Work with Message Descriptions command, WRKMSGD:

WRKMSGD MSGID(*FIRST) MSGF(*LIBL/QCPFMSG)

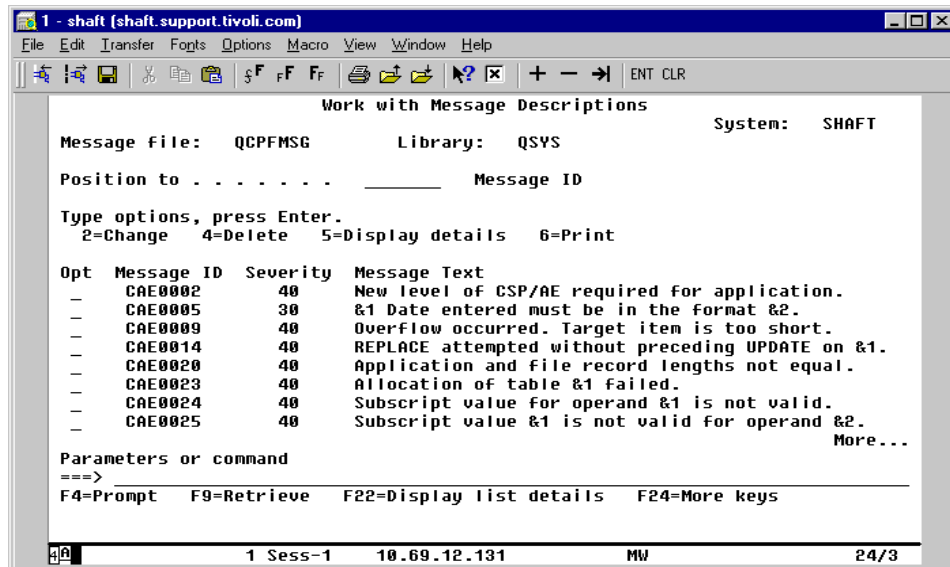


Figure 151. Work with message descriptions

Notice, in Figure 151, that an option is provided to change the message. This can be used, for example, to change the severity or even the text of a message if desired. Also, note that new messages can be added to the message file using the F6 (Add) key. However, the IBM-supplied message files and messages are very rarely changed. It is very common, however, for new message files to be created to contain new messages for user-written applications. We will create a message file and a new message in Section 7.3.5.5, "Using TEC to display User Admin UMBO_SKEL1 joblog" on page 371.

Now, let us examine the details of message CPF0907. Note that, rather than paging down to get to this message description, we simply put this message ID into the *Position to...* option at the top of the panel and press Enter. This immediately scrolls the list to this message. We then select option 5=Display details beside this message. These details include the message text, field data definition, and message attributes. Figure 152 on page 276 shows the message text.

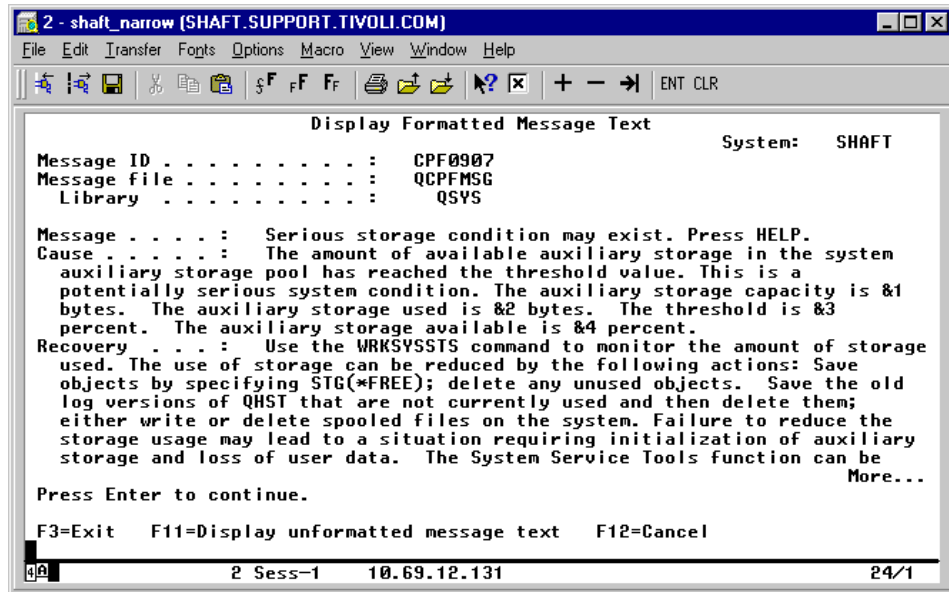


Figure 152. Display message description (message text)

Notice the use of the fields (&1, &2, &3, and &4) in the message text. Fields are variables that identify the location in the message text where data will be placed when it is supplied at the time the message is sent. These field definitions are shown on the Display Field Data panel in Figure 153 on page 277.

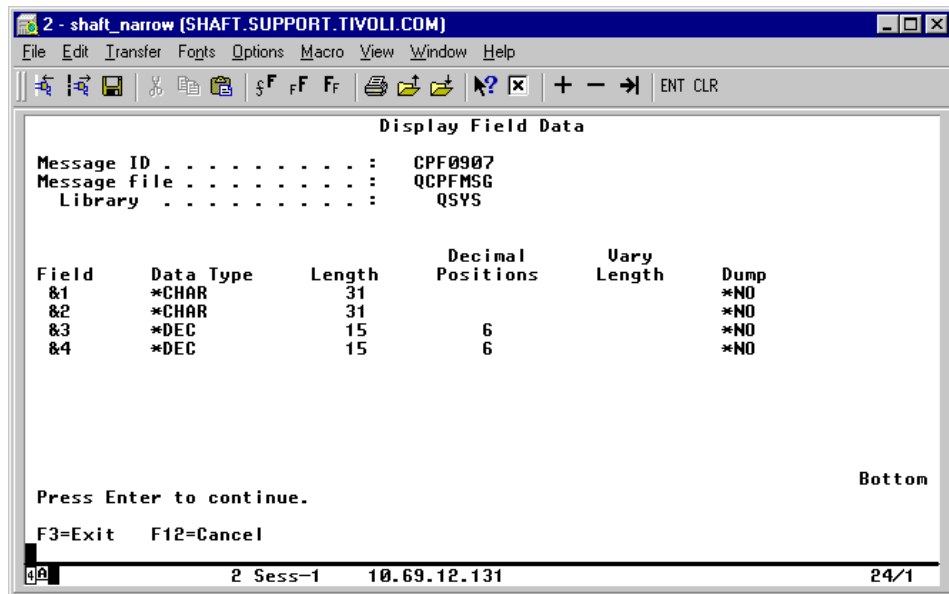


Figure 153. Display message description (field data)

The message detail panel, shown in Figure 154 on page 278, shows several of the message attributes.

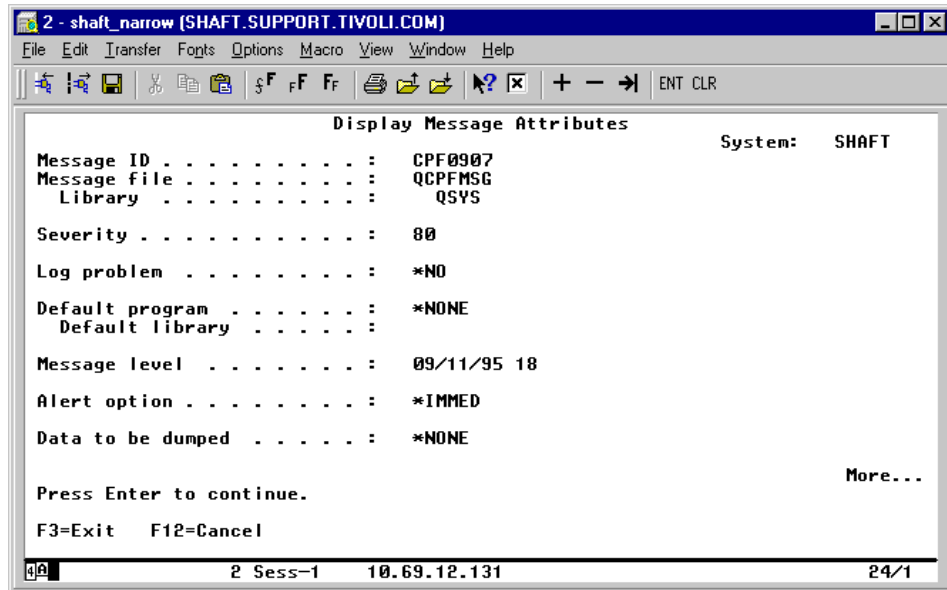


Figure 154. Display message description (message attributes)

For the purposes of our discussion, we will point out only two of the attributes: The Severity and Alert options. The Severity attribute will be used in our discussion of the AS/400 Message Adapter. The Alert option, on the other hand, will become significant in our discussion of OS/400 Alert support.

Not all messages on the AS/400 have a predefined structure like those we have seen. Impromptu or immediate messages, as they are called, can be sent using the OS/400 Send Message command, `SENDMSG`. The Send Message command can be used by an interactive user to send a message as shown in Figure 155 on page 279.

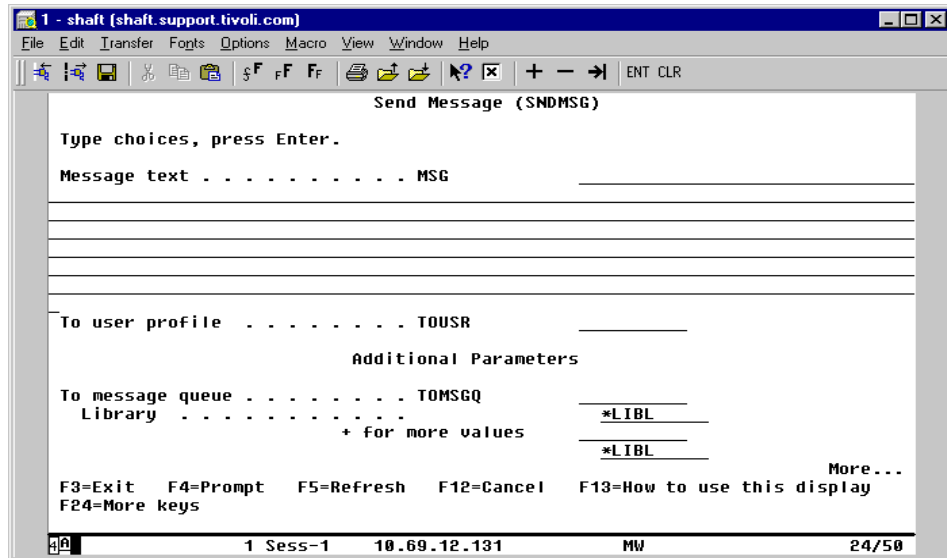


Figure 155. Send message (1 of 2)

Notice from the screen in Figure 155 that impromptu messages have no message ID associated with them, nor do they use Field data. Rather, they simply send a text string. From a TEC perspective, the message ID and field data values become TEC slot values that TEC can then use for such things as filtering and rule firing. Since impromptu messages do not have these attributes, whenever possible, we suggest the use of other OS/400 message commands, such as the Send User Message command, `SNDUSRMSG`. We will be using the Send User Message command described in Section 7.3.5.4, “Responding to an AS/400 Inquiry Message from TEC” on page 353.

Nonetheless, the Send Message command is often used by end users to send messages to the System Operator if they need assistance or have a question to which they would like a response. Notice the second panel of the Send Message command shown below the Message Type parameter in Figure 156 on page 280.

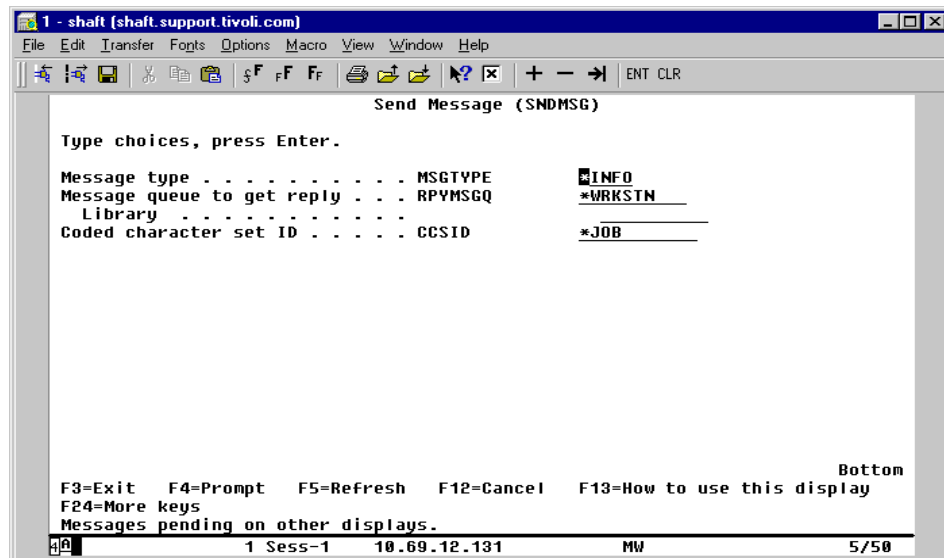


Figure 156. Send message (2 of 2)

When a message is sent, the sending user or program specifies where the message is sent and whether the message is an Informational message or an Inquiry message. We will see these two types of messages illustrated when we look at the System Operator Message queue. Additionally, in Section 7.3.5.3, “Editing the CDS file” on page 347, we will illustrate a response to an Inquiry message from a TEC console. For complete details of the Send Message and Send User Message commands, refer to the *OS/400 CL Reference V3R2*, SC41-3722.

On the AS/400, messages are sent to message queue objects: Each user has an assigned message queue, and each workstation has a message queue. Strictly speaking, programs also have message queues while they are running. However, these message queues are temporary internal objects that are not accessible using an AS/400 message adapter. Additionally, message queues can be created as desired using the Create Message Queue command, `CRTMSGQA`. Applications will often create and write messages to their own message queues. A complete list of available message queues on the AS/400 can be identified by using the Work with Message Queues command, `WRKMSGQ`, as follows:

```
WRKMSGQ MSGQ (*ALL/*ALL)
```

One of the most widely used message queues on the AS/400 is the system operator’s message queue called the QSYSOPR message queue. This

message queue is in the QSYS library. By using the Display Message command, DSPMSG, you can see the messages on a message queue. For example, the following command:

```
DSPMSG MSGQ (QSYS/QSYSOPR)
```

will display the messages on the QSYSOPR message queue. An example is shown in Figure 157.

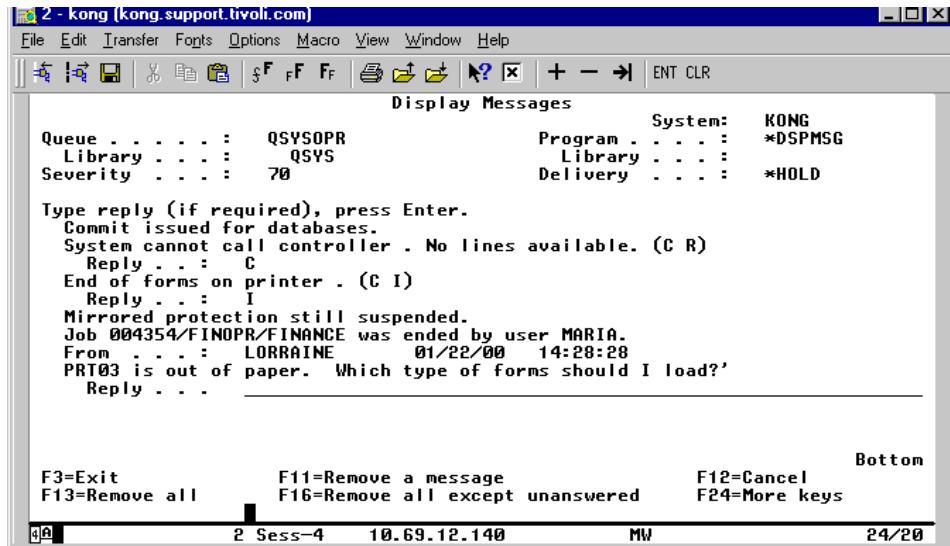


Figure 157. Display message (QSYSOPR)

By placing your cursor on one of the messages in the message queue and pressing F1 (Help), you can see the Additional Message Information. As shown in Figure 158 on page 282, for the *Mirrored protection still suspended.* message, the Additional Message Information will show the Message ID, Severity, Message Type, and other details.

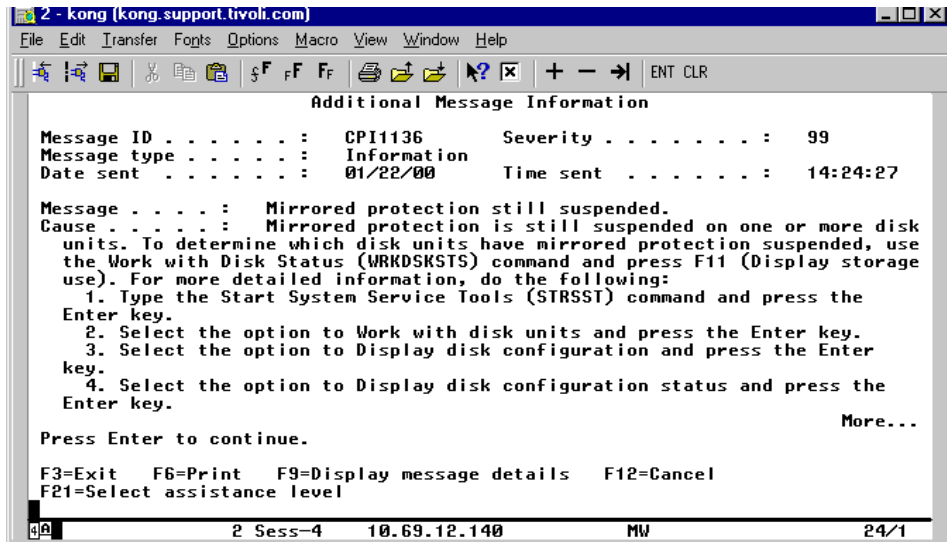


Figure 158. Additional message information

Additionally, pressing the F9 (Display message details) key will provide still more details including the sending job, user, and program.

7.3.2.2 AS/400 Message Adapter

The TEC AS/400 Message Adapters forward events created from messages on the monitored OS/400 message queue to a TEC event server. An AS/400 Message Adapter functions in the following manner.

1. At startup, it reads the specified adapter configuration file using the keywords specified within it to determine such things as:

ServerLocation

This specifies the name of the host on which the event server is installed. Since the AS/400 adapters are non-TME adapters, the hostname must be specified as either *host_name* or *IP_address*. @EventServer is not valid. Up to eight values, separated by commas, can be listed. The first location is the primary event server; others are secondary servers to be used in the order specified when the primary event server is down.

There is no default value specified. In order for the Adapter to function, this keyword value must be supplied.

ServerPort	<p>This specifies the port number on which the event server listens for events. A value of 0 means that the port number will be retrieved using the portmapper on the event server. If your event server is on Windows NT, which has no portmapper daemon, this value should be specified to match the <i>tec_recv_agent_port</i> value in the .tec_config file in \$BINDIR/TME/TEC directory.</p> <p>The default value is 0.</p>
MsgQueue	<p>This specifies the OS/400 message queue to poll. The message queue must exist at the time the adapter is specified. Only one message queue can be specified. If you wish to monitor multiple message queues, you should start a message adapter for each. The message queue should be specified as:</p> <p><i>library/message_queue</i></p> <p>The default value is QSYS/QSYSOPR.</p>
ProcessExistingMsgs	<p>This specifies whether, when it is initially starting, the AS/400 message adapter should process all existing messages on the message queue or just new ones.</p> <p>The default is NO.</p>
PollInterval	<p>This specifies the number of seconds the AS/400 message adapter will wait before checking the message queue for the arrival of new messages.</p> <p>The default value is 20.</p>
AdapterCdsFile	<p>This specifies the IFS path name of the Class Definition Statement (CDS) file. This file can be a native AS/400 physical file, but the path must still be specified using the IFS format:</p> <p>QSYS.LIB/<i>library</i>.LIB/<i>filename</i>.FILE/<i>member</i>.MBR</p> <p>The default value is:</p> <p>QSYS.LIB/QUSRSYS.LIB/QAYMACDS.FILE/MSGCDS.MBR.</p>

FilterMode	<p>This specifies whether events that match the Filter and FilterCache statements are sent (FilterMode=IN) or discarded (FilterMode=OUT).</p> <p>The default value is OUT.</p>
Filter	<p>This specifies how events are filtered. Filter statements are used with the FilterMode statement to determine which events are sent to the event server and which events are discarded. The format of Filter statements is:</p> <p><i>Filter:Class=class_name;attribute=value;...;attribute=value</i></p>
BufferEvents	<p>This specifies whether or not to buffer events if they cannot be sent to the event server. If this value is set to anything other than YES, events are not buffered.</p>
BufEvtName	<p>This specifies the name of the AS/400 message adapter's buffer file. This file will reside in the /QIBM/UserData/Tivoli/TEC/MSGQ/Cache directory.</p> <p>By default, this value will be set to the Adapter's name as specified on the Start TEC Event Adapter command, <code>STRTECADP</code></p>
FilterCache	<p>This specifies how buffered events are filtered. FilterCache statements are used with the FilterMode statement to determine which events are sent and cached (should the event server not be available) and which events are discarded. The format of FilterCache statements is:</p> <p><i>FilterCache:Class=class_name;attribute=value;...;attribute=value</i></p>
TestMode	<p>This specifies whether test mode is turned on or off. When TestMode=YES, the ServerLocation keyword specifies the file to which events are logged.</p> <p>The default value is NO.</p>

2. If ProcessExistingMsgs is set to YES, the message adapter will extract all the messages currently on the specified message queue (MsgQueue).

3. At the specified PollInterval, it checks the message queue for any new messages to extract.
4. Whenever a message is extracted from the specified message queue, the information from the message is formatted into a TEC event using the specified Class Definition Statement file (AdapterCdsFile).

The CDS file defines how TEC events are constructed from AS/400 messages using the following SELECT, FETCH, and MAP statements to define all supported event classes:

SELECT This specifies the criteria an incoming event must satisfy to match a class.

FETCH This retrieves data from the incoming event that is necessary to fill the attribute values.

MAP This specifies how to fill attribute values for an event instance from data retrieved by FETCH statements.

The provided default message adapter CDS file can be broken down into the following sections.

MAP_DEFAULT statement

Some attributes have a constant value for all events generated by the AS/400 message adapter. These attributes are defined in the MAP_DEFAULT statement so that each event does not have to specify them. The values are specified in *name=value* pairs where *name* specifies the attribute, or slot name, and *value* specifies the value placed in the attribute.

The following screen contains the MAP_DEFAULT statement in the supplied message adapter CDS file.

```

MAP_DEFAULT
  alert_option      = $ALERT_OPTION;
  arg1              = $ARG1;
  arg2              = $ARG2;
  arg3              = $ARG3;
  arg4              = $ARG4;
  arg5              = $ARG5;
  arg6              = $ARG6;
  arg7              = $ARG7;
  arg8              = $ARG8;
  date              = $DATE;
  hostname          = $HOSTNAME;
  msg               = $MSG;
  msg_file_library  = $MSG_FILE_LIBRARY;
  msg_file_name     = $MSG_FILE_NAME;
  msg_help          = $MSG_HELP;
  msg_id            = $MSG_ID;
  msg_key           = $MSG_KEY;
  msg_library_used  = $MSG_LIBRARY_USED;
  msg_severity      = $MSG_SEVERITY;
  msg_type          = $MSG_TYPE;
  origin            = $ORIGIN;
  send_date         = $SEND_DATE;
  send_job          = $SEND_JOB;
  send_job_number   = $SEND_JOB_NUMBER;
  send_program_name = $SEND_PROGRAM_NAME;
  send_time         = $SEND_TIME;
  send_user_profile = $SEND_USER_PROFILE;
  severity          = $SEVERITY;
  source            = $SOURCE;
  sub_source        = $SUB_SOURCE;
END

```

Notice that many of the attributes are initialized to variable values based on the message description or the message details on the message queue. \$MSG_ID and \$MSG_SEVERITY, for example, refer to the Message ID and Severity as defined in the message, while \$MSG_TYPE and \$SEND_TIME, for example, refer to the Message type and Time Sent as displayed when viewing the message on the message queue. Refer to the previous discussion using the Display Message Description and the Display Message Queue commands to see these values. Additionally, refer to Chapter 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507, for a complete list of the defined CDS keywords.

Notice, that arg1 through arg8 are also defined slot values. These values correspond to the field data values, &1 through &8, as defined in the message. There may be cases where these are not all used in the message description. In these cases, the unused attributes will be empty. Also, there may be cases where messages use field data values above &8. In these cases, in order for these field data values to be sent as TEC event attributes,

they must be defined within the specific Class definitions since they are not available as default values. An example of this is shown in the Supplied Class Definitions discussion to follow.

The last value that we wish to mention is \$SEVERITY because this is a value created by the TEC adapter by mapping the message severity of the AS/400 message to a TEC severity (HARMLESS, FATAL, and so on) using the message severity ranges specified in Chapter 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507.

Note

Currently, there is a known problem (APAR IX89491) where this mapping did not include severity 99 properly to FATAL. Actually, if you carefully examine the severity mapping documented in Chapter 4, "AS/400 Endpoint configuration" on page 61 of this redbook, you will notice that it also excludes severity 99. This is reported in APAR IX89493.

In Section 7.3.5, "TEC AS/400 Event Adapter scenarios" on page 341 of this redbook, we will illustrate a workaround for this problem.

Class *DISCARD* definitions

Following the MAP_DEFINITION statement, specific classes are defined. By specifying a Class Name of *DISCARD*, any incoming events matching the SELECT statement are discarded. The following is an excerpt from the supplied message adapter CDS file, which is used to discard OS/400 cleanup messages.

```

# If a match is made with a CLASS named *DISCARD*, the event is thrown away.
# It is more efficient to have these classes at the top of the evaluation than
# to go all the way through each CLASS without a match.

# CPI1E23 - Cleanup has started.
# CPI1E81 - xxxxxx of xxxxxx messages deleted from nnnnnnnn.
# CPI1E82 - Cleanup of user messages started.
# CPI1E83 - Cleanup of user messages successfully completed.
# CPI1E84 - Cleanup of operator and work station messages started.
# CPI1E85 - Cleanup of operator and work station messages completed
# CPI1E86 - Cleanup of system journals and system logs started.
# CPI1E87 - Cleanup of system journals and system logs successfully completed.
# CPI1E88 - Cleanup of OfficeVision/400 calendar items started.
# CPI1E89 - Cleanup of OfficeVision/400 calendar items successfully completed.
# CPI1E91 - User cleanup program started.
# CPI1E92 - User cleanup program successfully completed.
# CPI1E93 - Cleanup of job logs and other system output started.
# CPI1E94 - Cleanup of job logs and other system output successfully completed.
CLASS *DISCARD*
SELECT
    1: $MSG_ID = CPI 1E23:1E23 1E81:1E89 1E91:1E94;
END

```

The SELECT statement here illustrates denoting a range of Message ID values using low:high pairs, thereby, identifying each of the Message IDs listed as comments above. For clarity, let us break this down pair by pair.

CPI 1E23:1E23 Matches only CPI1E23.

1E81:1E89 Matches message ID's CPI1E81 through CPI1E89.

1E91:1E94 Matches message ID's CPI1E91 through CPI1E91.

Using high:low pairs is useful when selecting consecutive messages. However, there is no requirement to use low:high pairs when specifying individual message IDs as shown in the next example, also used in the supplied CDS file.

```

# CPC1E1D - Cleanup has completed.
CLASS *DISCARD*
SELECT
    1: $MSG_ID = CPC1E1D;
END

```

Supplied class definitions

Following the Class DISCARD definitions, the supplied AS/400 Message Classes are defined. A complete listing of these is provided in Chapter 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507. We will show a couple examples here to illustrate how event classes are defined.

The following excerpt from the supplied CDS file illustrates the definition of AS400_Device_No_Longer_Communicating.

```
# CPF2677 - Device xxxxxxxx no longer communicating.
CLASS AS400_Device_No_Longer_Communicating
SELECT
  1: $MSG_ID = CPF2677;
END
```

Notice that this class definition simply selects events with a Message ID of CPF2677. It adds no new TEC attributes, nor does it alter any of the attributes' existing values.

Now, examine the following AS400_Controller_NotReplying Class Definition statement, which is also from the supplied CDS file.

```
# CPA58E0 - Controller XXXXXXXX not replying. Remote system or config problem
CLASS AS400_Controller_NotReplying
SELECT
  1: $MSG_ID = CPA58E0;
  2: ATTR(=, "arg24");
MAP
  arg24 = $V2;
END
```

In this case, events with a Message ID of CPA58E0 and defining an attribute, named arg24, will be selected. To better understand this arg24 attribute, it is helpful to review the CPA58E0 message description using.

DSPMSGD RANGE(CPA58E0)

See Figure 159 on page 290.

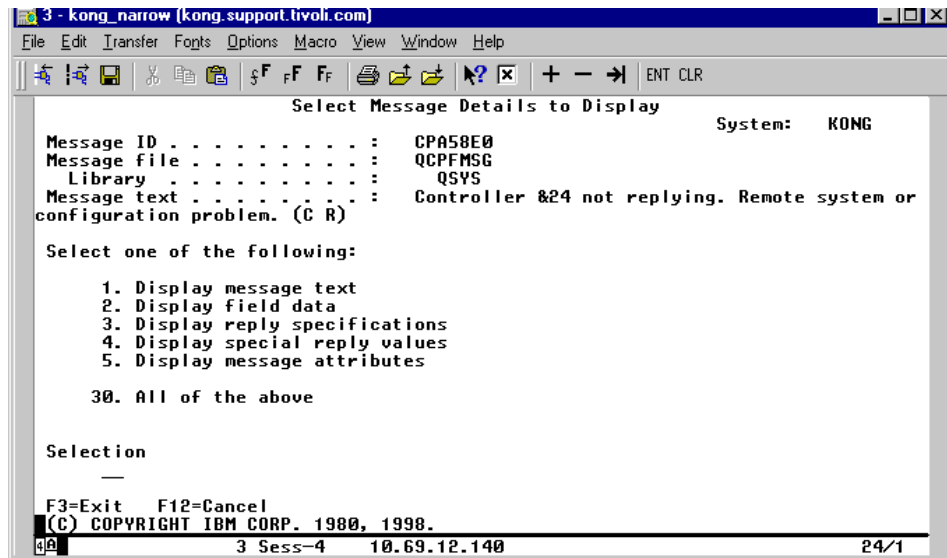


Figure 159. Display message description for CPA58E0

Notice that the Controller name in the message text is field data &24. Recall, that the DEFAULT_MAP statement only defined the TEC attributes (arg1 through arg8) for &1 through &8; so, in order for this CLASS to supply this, an attribute called *arg24*, the MAP statement, is used to set the attribute value to \$V2. \$V2 is a pseudo-variable signifying the output of SELECT statement 2:

```
2: ATTR(=, "arg24")
```

The output from this is the value of the arg24 attribute.

To summarize, when message ID CPA58E0 is sent and specifies a value for field data &24, it will be sent to TEC as CLASS AS400_Controller_NotReplying. Also, it will have an additional attribute, named arg24, which will contain the message field data for &24.

AS400_MSG Class Definition

Following the supplied Class Definition statements, the AS400_MSG Class is defined as follows:

```
CLASS AS400_MSG
SELECT
1: $MSG_ID = * *.*;
END
```

Any event that was not matched by any previous Class Definition will match this SELECT statement. Notice the use of “*” wildcards in the low:high pair notation to match any message IDs. This will also match all impromptu messages which, you may recall, do not have any Message IDs.

5. These formatted events are then filtered based on the FilterMode and Filter statements in the configuration file.
6. Events that are included (or not excluded) based on the above filtering are then sent to the specified SeverLocation and ServerPort using TCP/IP sockets.
7. If the TEC server is not available, the events will be cached based on the values specified for BufferEvents, BufEvtName, Filter, and any FilterCache statements in the configuration file.

When the event arrives at the TEC server, it is processed based on the classes and rules in the active rulebase. The supplied AS/400 Message Adapter baroc file on the TEC event server is as400msg.baroc. It is automatically compiled into the active rulebase when the event server is installed.

There is also a copy of this file as shipped on the AS/400 if you prefer to access them from there. Of course if you customize these files on the TEC server, you may want to FTP the updated files to the AS/400 so that the actual version being used will be accessible. This baroc file on the AS/400 is:

`/QSYS.LIB/QUSRSYS.LIB/QAYMABRC.FILE/MSGBRC.MBR`

Note

We have not attempted to provide a complete listing of the available configuration keywords. Rather, we have tried to identify those necessary for an understanding how the AS/400 message adapter functions. We have also tried to mention any of the keywords we will use in the remainder of this section. Refer to the “Adapter Files” sections of Chapter 1 and Chapter 4 of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, for a complete discussion of the available configuration file keywords as well as a detailed description of the adapter CDS file.

Likewise, we have not attempted to provide a complete discussion of the Class Definition Statement File. Rather, we have provided the basics necessary to understand the AS/400 message adapter as well as the customizations and scenarios to follow. A detailed discussion of the CDS file can be found in Appendix B, “Class Definition Statement File Reference”, of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507.

7.3.2.3 OS/400 Alert Support

Alerts are specific types of system messages that are used to identify problems or impending problems. When you configure alert support on your system, you receive an alert whenever a problem occurs. The alert attempts to provide analysis data about the cause and suggest possible recovery actions. OS/400 alert support is quite flexible and, therefore, complicated; so, our discussion here is intentionally limited to what is necessary to move onto an explanation of the OS/400 alert adapter. And, while we will point out the ability to create new alert tables and alert descriptions, we will not extend our discussion nor include any scenarios doing this. Appendix A of the *OS/400 Alerts Support Guide* provides sample procedures of this should you desire to review them.

Let us begin with a discussion of how alerts are created. The creation of alerts on an AS/400 is controlled by all of the following four values:

1. **The Alert Status Network Attribute** - This value can be displayed or changed using the Display Network Attributes command, `DSPNETA`, or the Change Network Attribute command, `CHGNETA`. The allowable values for this attribute are:
 - *ON
Alerts are created for all alert conditions, except unattended.
 - *UNATTEND

Alerts are created for all alert conditions including unattended. When this value is specified, the system is said to be *unattended* or in *unattended mode*.

- *OFF

No alerts are created.

The Alert Status Network Attribute value can be checked using the Display Network Attribute command as follows:

DSPNETA

This command will bring up the Display Network Attribute panels. The Alert Status value is displayed on the second page of these attributes as illustrated in Figure 160 on page 293.

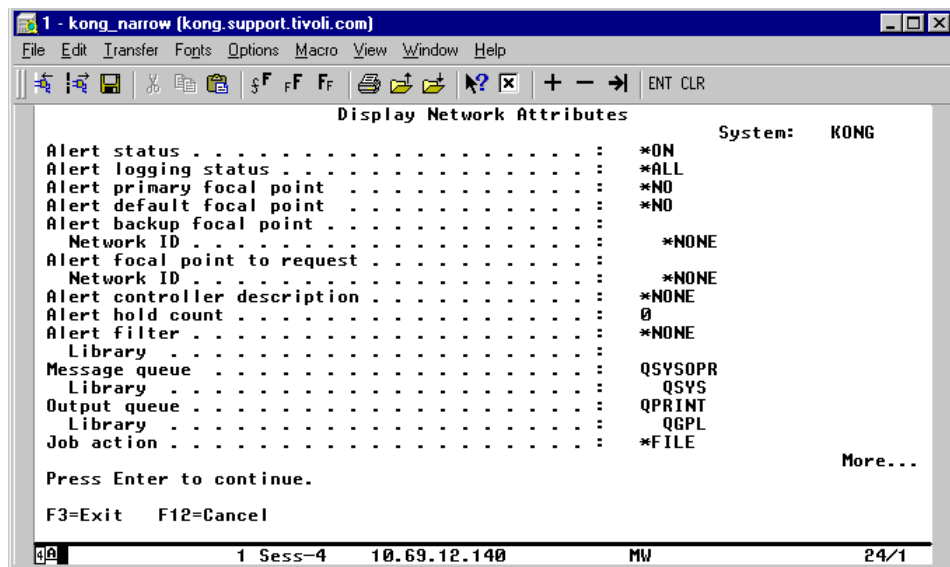


Figure 160. Display network attributes

Note

Note that there are numerous other values related to OS/400 Alert support. In this section, the only other value that will be examined is the Alert Filter (ALRFTR). In Section 7.3.4, “Troubleshooting TEC AS/400 Event Adapters” on page 313, we will discuss the Alert logging status (ALRLOGSTS). For an explanation of the other alert-related network attributes including alert focal points and sphere of control, we refer you to the *OS/400 Alerts Support V3.2.0*, SC41-3413.

2. **The Alert Option specified in the Message Description** - As shown in Figure 154 on page 278, this value can be displayed using the Display Message Description command, `DSPMSGD`. It can also be changed using the Change Message Description command, `CHGMSGD`. The allowable values for the Alert Option attribute are:

- ***NO**

No alert is created.

- ***IMMED**

An alert is created at the moment it is received on a message queue that allows alerts. This specification will be discussed in the following.

- ***UNATTEND**

An alert is created only if the Alert Status network attribute is set to ***UNATTEND**.

- ***DEFER**

An alert will be created only after problem analysis has been performed on the message in the Problem Log. The Problem Log can be accessed using the Work with Problems command, `WRKPRB`. The ***DEFER** option is valid only when the Log Problem (LOGPRB) parameter is set to ***YES**.

3. **The Allow Alerts specification on the Message Queue** - Recall that, when a message is sent, the sender specifies to which message queue the message is sent. One of the available parameters on a message queue is Allow Alerts (ALWALR). This can be set to:

- ***NO** - This does not allow alerts to be generated from this message queue.

- ***YES** - This allows alerts to be generated from this message queue. The QSYSOPR message queue allows alerts to be created.

4. **Alert Descriptions in an Alert Table** - In many ways, an Alert Table resembles a message file. For example, compare the QCPFMSG alert table with the QCPFMSG message file shown in Figure 152 on page 276. We access the first panel of alerts contained in the OS/400 QCPFMSG alert table by using the Work with Alert Descriptions command, `WRKALRD`:

```
WRKALRD MSGID(*FIRST) MSGF(*LIBL/QCPFMSG)
```

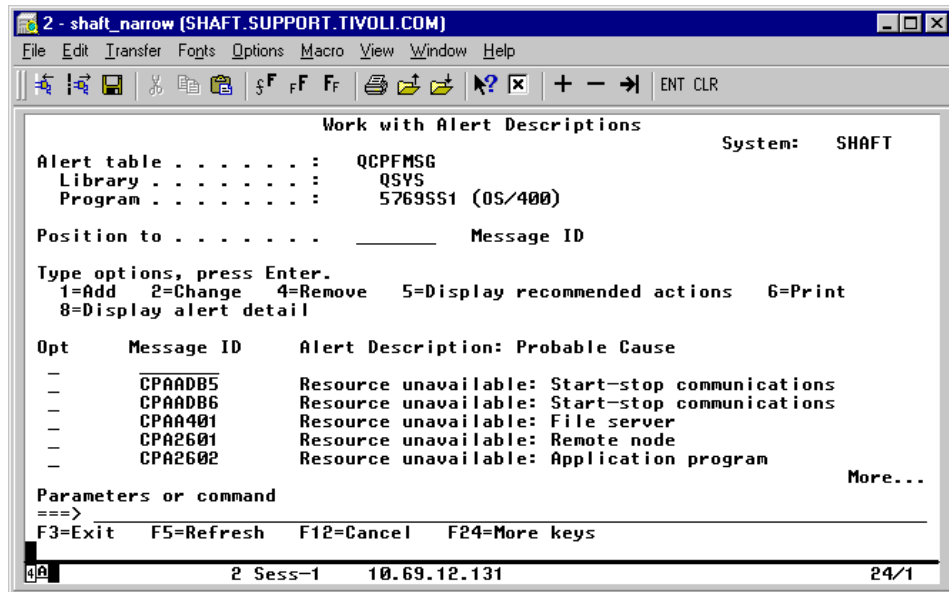


Figure 161. Work with Alert Descriptions

Notice, in Figure 161, that an option is provided to Change the alert. This can be used, for example, to change the alert type or even the code points associated with the alert description or probable cause. Also, note that new alerts can be added to the alert table using option 1. However, the IBM-supplied alert tables and alerts are very rarely changed. However, creating new alert tables for use with new application-created messages and message files is possible. By defining your own alert descriptions, you can provide the network operator with specific information about the causes of and recovery for application problems.

Now, let us examine the details of alert for CPF0907. As we did in the Work with Message Description panel, we can simply put this message ID into the *Position to...* option at the top of the panel and press Enter. This immediately scrolls the list to this message. We then select option 8 (Display alert detail) beside this message. The alert details include the alert type, alert description, alert option, and alert ID as shown in Figure 162 on page 296.

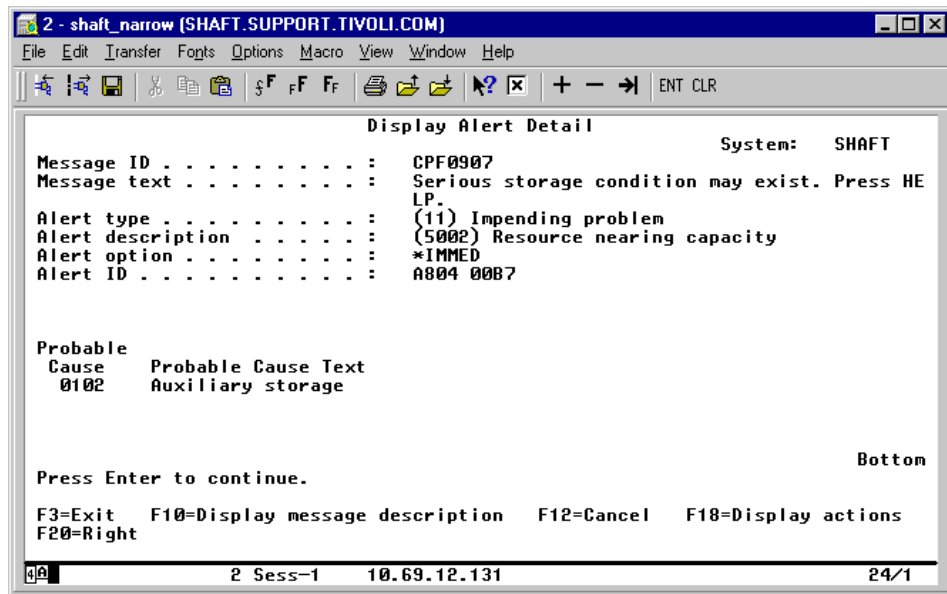


Figure 162. Display Alert Detail

From the Work with Alert Description panel, you can use option 5 to display the recommended action associated with an alert. The recommended actions associated with CPF0907 are shown in Figure 163 on page 297.

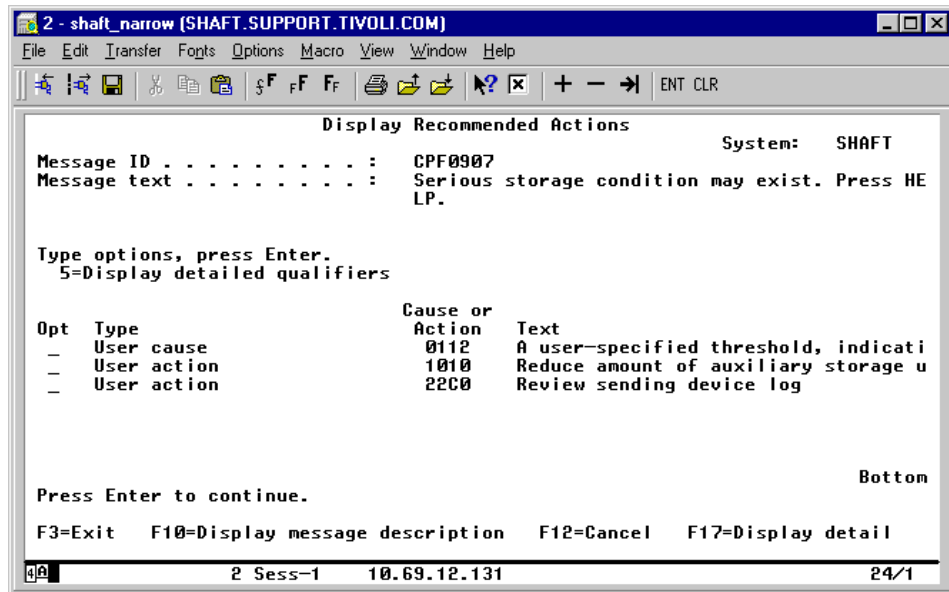


Figure 163. Display Recommended Actions

For the purposes of our discussion, we will not go into code point definitions on the AS/400. If you are interested in this level of information, refer to the *OS/400 Alerts Support V3.2.0*, SC41-3413.

The important point that we want to make is that there is a one-to-one correspondence between a message (defining an error) and an alert (defining a problem notification). An alert description for a message ID must exist for the message to generate an alert. That is, in order for message CPF0907 in the QCPFMSG message file to generate an alert, the alert must be defined for this message ID in the QCPFMSG alert table.

To summarize, when alerting is active on the AS/400 (#1), the AS/400 creates an alert when an alertable message (#2) is sent to a message queue that allows alerts to be created (#3), and an alert description corresponding to the message ID must exist in the alert table (#4). It is also possible to generate alerts in the following ways:

- By selecting to generate an alert when analyzing a problem. We will show an example of doing this in Section 7.3.4.6, "Generating a Test Alert" on page 332.

- By using the QALGENA and QALSENDA system API. These will not be discussed here. Refer to the *AS/400 System API Reference V3R2*, SC41-3801, for details.

The final object that we need to discuss to provide an understanding of the AS/400 alert adapter is an alert filter. An alert filter is an OS/400 object that can be used to assign alerts to a group and then process the alert according to the actions defined on the group. Figure 164 illustrates an Alert Filter.

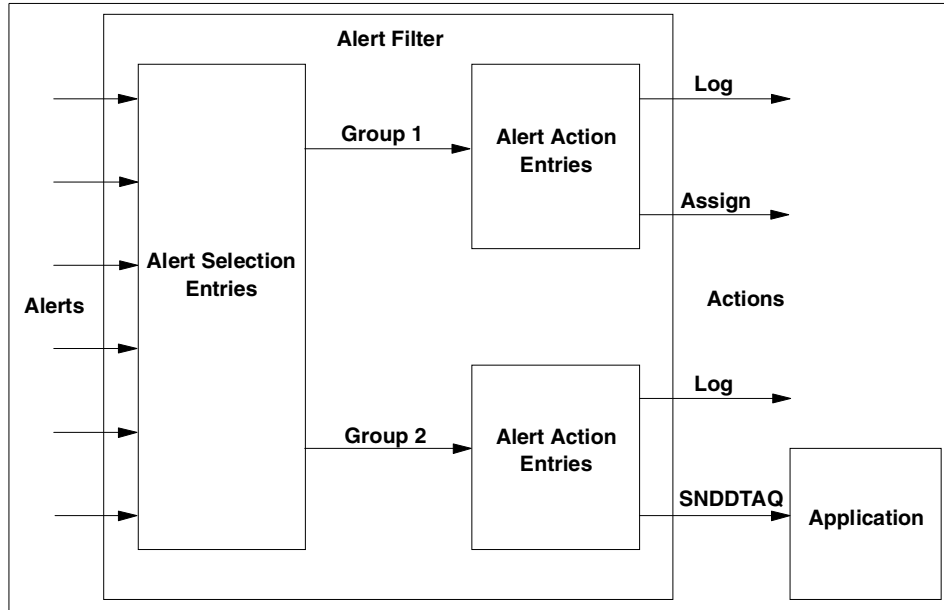


Figure 164. Alert Filter mechanism

Note

At this point, we have introduced the term data queue, but we have never defined it. Simply stated, a data queue is an OS/400 object that is used to communicate and store data used by programs. The purpose of sending some events, such as the FINANCE alerts to a data queue, is so that an application program monitoring that data queue can pick it up and process it as desired.

A data queue can be created using the Create Data Queue (CRTDTA) command.

To provide a little more detail, we shall create an example alert filter here. We shall refer back to this alert filter in our discussion of the AS/400 alert adapter.

An alert filter can be created using the Create Alert Filter (CRTFTR) command. We create the LPVFTR alert filter in LPVLIB library by issuing the following.

```
CRTFTR FILTER(LPVLIB/LPVFTR) TYPE(*ALR) TEXT('Example Alert Filter')
```

We then add the following selection entries to the LPVFTR alert filter using the following Add Alert Selection Entry (ADDALRSLTE) commands.

```
ADDALRSLTE FILTER(LPVLIB/LPVFTR) SELECT((IF *RSCTYPE *EQ PRT)) SEQNBR(10) GROUP(PRINTER)
ADDALRSLTE FILTER(LPVLIB/LPVFTR) SELECT((IF *RSCTYPE *EQ DSK) (OR *RSCTYPE *EQ TAP) (
*EQ OPT)) SEQNBR(20) GROUP(STORAGE)
ADDALRSLTE FILTER(LPVLIB/LPVFTR) SELECT((IF *MSGID *CT FIN)) SEQNBR(30) GROUP(FINANCE)
```

The first entry will assign any alert from printer resources into a group called PRINTERS. Similarly, the second entry will assign any alert from tape, disks, or optical devices into the STORAGE group. Finally, the third entry will assign any alert created from a message ID containing “FIN” into the FINANCE group. The assumption here is that we have an application on our system that has defined its own alertable messages with message IDs of the form FINxxxx.

Also, every alert filter created will automatically also contain a default selection entry for a group called *DEFAULT. The sequence number will be *LAST and the selection criteria *ANY. This entry will catch any alert without matching any other selection entry.

Note that, at this point, we have not specified that any actions be taken as a result of any of these alerts. The selection entries simply assign the alerts to a group. To specify how each group should be processed, we add the following action entries to our alert filter using the Add Alert Action Entry (ADDALRACNE) commands.

```
ADDALRACNE FILTER(LPVLIB/LPVFTR) GROUP(PRINTERS) LOG(*NO) ASNUSER(FRANKLIN)
SNDDTAQ((SUPPORT/PRINTERS))
ADDALRACNE FILTER(LPVLIB/LPVFTR) GROUP(STORAGE) LOG(*YES) GENIRAP(*YES)
ADDALRACNE FILTER(LPVLIB/LPVFTR) GROUP(FINANCE) LOG(*YES) SNDDTAQ((FINLIB/FINERRORS))
```

The first action entry will assign any alert in the PRINTERS group to the user FRANKLIN. It will also place the alert on a data queue called PRINTERS in the SUPPORT library. This alert will not be logged. The second action will log any alert in the STORAGE group and generate an SNMP trap for it. The final action entry will log any alert in the FINANCE group and place it on the FINERRORS data queue in the FINLIB.

Also, every alert filter created will automatically also contain a *DEFAULT action entry corresponding to the *DEFAULT selection entry; so, our example alert filter performs as illustrated in Figure 165.

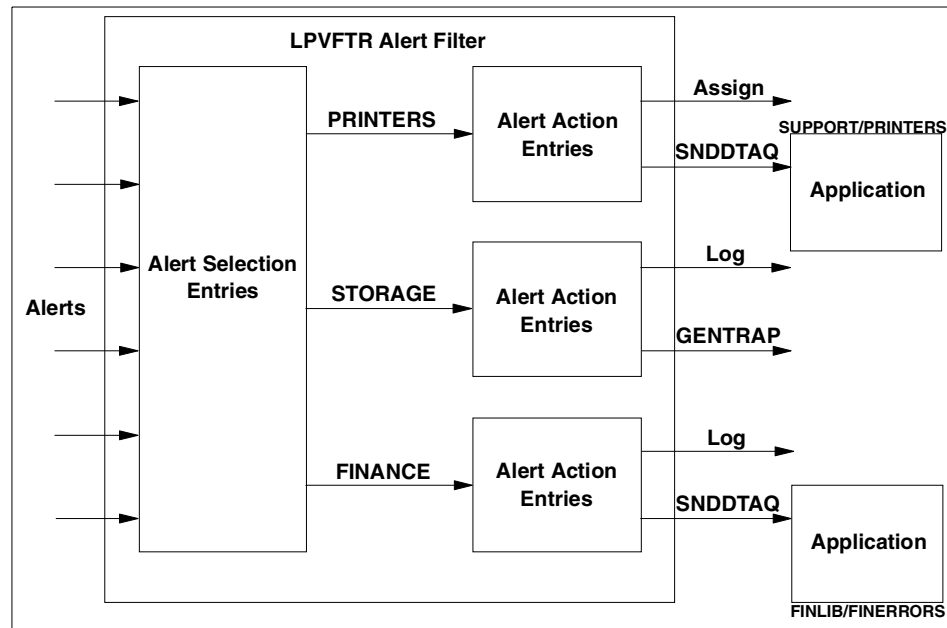


Figure 165. The LPVFTP Alert Filter mechanism

Notice that, while you can create numerous Alert Filters, only one Alert Filter can be active on the system at a time. The currently active Alert Filter is the one specified by the Alert Filter network attribute as shown in Figure 160 on page 293.

7.3.2.4 AS/400 Alert Adapter

The TEC AS/400 Alert Adapter functions in a manner that is actually very similar to the TEC AS/400 Message Adapter. In fact, we have already defined many of the keywords in the configuration and CDS file in Section 7.3.2.2, “AS/400 Message Adapter” on page 282. Therefore, while we will list them

again here for the sake of explanation, we will not detail any of the keywords that you can find detailed in that section.

The TEC AS/400 Alert Adapters forward events created from alerts arriving on the monitored data queue to a TEC event server. An AS/400 Alert Adapter functions in the following manner:

1. At startup, it reads the specified adapter configuration file using the keywords specified within it to determine such things as:

- **ServerLocation**

- **ServerPort**

- **Filter** - This specifies the OS/400 alert filter to be used. The alert filter must exist at the time the adapter is started. Only one alert filter can be specified. If you wish to specify multiple alert filters, you should start an alert adapter for each. The alert filter should be specified as:

library/alert_filter

The default value is QTMEYAA/QYAAFTR.

- **FilterDataQueue** - This specifies the data queue that the alert adapter should monitor for incoming alerts. This keyword is required if the Filter specified is already registered with the system. The Filter is registered with the system if it is the Alert Filter value on the Display Network Attributes panel or if it was registered using the QNMRGFN OS/400 system API. It would, for example, have been registered using this API if there is a currently-active alert adapter specifying this filter. Refer to the final example, which discusses running multiple adapters monitoring the same alert filter.

When this keyword is specified, the data queue specified must already have been created. If no filter is specified or if the filter specified is not already registered, this keyword is not necessary.

It is often easier to understand how the Filter and FilterDataQueue parameters function with a few examples.

- Default Values

Filter	QTMEYAA/QYAAFTR
---------------	-----------------

FilterDataQueue	none
------------------------	------

When the alert adapter starts, an alert filter called QYAAFTR and a data queue named DEFAULT are both created in the QTMEYAA library. The alert filter is created with the selection entry called QTECALERT. The QTECALERT selection criteria is set to *ANY so all it will catch are alerts. The corresponding QTECALERT action

entry specifies that the alerts be sent to a DEFAULT data queue. The alert adapter monitors this DEFAULT data queue for alerts.

- Specifying an existing alert filter without specifying a data queue

Filter LPVFTR

FilterDataQueue none

When the alert adapter starts, a data queue with the same name as the alert adapter is created in the QTMEYAA library. It is assumed that the user has already set up a QTECALERT selection entry as they desire. If there is a corresponding QTECALERT action entry, this action entry is updated to send the alerts to the created data queue. If a QTECALERT action entry is not found, one will be added.

Note

No QTECALERT selection entry will be added. If a Filter is specified, it is assumed that the user has created the QTECALERT selection criteria they desire. Obviously, if there is no QTECALERT selection entry the QTECALERT action entry will never be used and no alerts will arrive at the created data queue.

If the specified alert filter is the currently active system alter filter specified in the system network attributes as shown in Figure 160 on page 293, then the FilterDataQueue must be specified.

Let us give a brief example. Suppose we wish to send alerts to the TEC event server that do not fall into any of the groups (PRINTERS, STORAGE, or FINANCE) for which we have selection entries defined in the LPVFTR alert filter. Refer to Figure 165 on page 300, to review the LPVFTR filter. We would need to add a QTECALERT selection entry to the LPVFTR similar to the following.

```
ADDALRSLTE FILTER(LPVLIB/LPVFTR) SELECT(*ANY) SEQNBR(40)
GROUP(QTECALERT)
```

This would catch any alert that did not match any of the previous selection entries. Now, when the alert adapter starts, it will create a data queue by the event adapter name and add a corresponding QTECALERT action entry to send the alert to the data queue.

- Specifying both an alert filter and a data queue

Filter LPVFTR

FilterDataQueue FINLIB/FINERRORS

When the alert adapter starts, it will monitor the data queue specified. In this case, it is assumed that the user has already setup the alert filter, selection entries and action entries as they desired.

Note

The specified data queue must be created as follows:

```
CRTDTAQ DTAQ(library/dataqueue) TYPE(*STD) MAXLEN(592) FORCE(*NO)
SEQ(*FIFO)
```

If the data queue is not specified with these parameters, the adapter will not start.

We would use the example settings shown in the case where we wished to send only the alerts from the FINANCE application to the TEC event server. Refer to Figure 165 on page 300. In this case, the alert adapter makes no modifications to the alert filter, selection entries, or action entries. These were all previously set up as we defined earlier. When we created the FINERRORS data queue, we used the specifications noted previously.

If you wanted to monitor both the FINERRORS and PRINTERS data queues in this manner, you could start a second alert adapter with a separate configuration file which specified:

```
Filter                LPVFTR
FilterDataQueue       SUPPORT/PRINTERS
```

That is, it is allowable for multiple alert adapters to run specifying the same alert filter, provided that the **FilterDataQueue** keywords are unique.

- ProcessExistingAlerts

This specifies whether, when initially starting, the AS/400 alert adapter should process all existing alerts on the specified FilterDataQueue or just new ones. The default is NO.

- AdapterCdsFile

The default value is:

```
QSYS.LIB/QUSRSYS.LIB/QAYAACDS.FILE/ALRCDS.MBR
```

- FilterMode

- Filter

- BufferEvents

- **BufEvtName**

This file will reside in the /QIBM/UserData/Tivoli/TEC/Alert/Cache directory.

- **FilterCache**

- **TestMode**

2. Based on the **Filter** and **FilterDataQueue** keywords, the adapter may create or update an alert filter. A data queue may also be created if needed. If the adapter is not already registered with the system, it will be registered. Finally, if appropriate, the data queue to be monitored will be created. If necessary, review the examples provided in the FilterDataQueue keyword explanation for details.
3. If ProcessExistingAlerts is set to YES, the message adapter will extract all the alerts currently on the monitored data queue.
4. Whenever an alert is placed on the monitored data queue, the information from the alert is formatted into a TEC event using the specified Class Definition Statement file (AdapterCdsFile).

The CDS file defines how TEC events are constructed from OS/400 alerts using the SELECT, FETCH, and MAP statements to define all supported event classes. Refer to Section 7.3.2.2, "AS/400 Message Adapter" on page 282, for a description of these statements.

The default alert adapter CDS file provided can be broken down into the following sections.

MAP_DEFAULT statement

Some attributes have a constant value for all events generated by the AS/400 alert adapter. These attributes are defined in the MAP_DEFAULT statement so that each event does not have to specify them. The values are specified in *name=value* pairs where *name* specifies the attribute, or slot name, and *value* specifies the value placed in the attribute.

The following is the MAP_DEFAULT statement in the supplied alert adapter CDS file.


```

MAP_DEFAULT
source = $SOURCE;
origin = $ORIGIN;
sub_origin = $SUB_ORIGIN;
hostname = $HOSTNAME;
adapter_host = $ADAPTER_HOST;
date = $DATE;
severity = $SEVERITY;
msg = $MSG;
adapter_host_snanode = $ADAPTER_HOST_SNANODE;
event_type = $EVENT_TYPE;
arch_type = $ARCH_TYPE;
product_id = $PRODUCT_ID;
alert_id = $ALERT_ID;
block_id = $BLOCK_ID;
action_code = $ACTION_CODE;
alert_cdpt = $ALERT_CDPT;
self_def_msg = $SELF_DEF_MSG;
causes = $CAUSES;
actions = $ACTIONS;
detailed_data = $DETAILED_DATA;
event_correl = $EVENT_CORREL;
incident_correl = $INCIDENT_CORREL;
adapter_correl = $ADAPTER_CORREL;
END

```

Notice that most of the attributes are initialized to variable values based on the alert description. \$ALERT_ID and \$CAUSES, for example, refer to the Alert ID and Probable Cause listing as defined in the alert while \$DATE and \$PRODUCT_ID, for example, refer to the date and time the alert occurred and the product identifier associated with generating the alert. Refer to the previous discussion using the Display Alert Description and Work with Alerts example in Section 7.3.4.6, “Generating a Test Alert” on page 332, to see these values. Additionally, refer to Chapter 3 of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, for a complete list of the defined CDS keywords.

Note

The *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, includes a chart mapping the AS/400 Alert Descriptions (TEC Event Classes) to a TEC Severity. This information is inaccurate. The Severity of an AS/400 Alert Event on the TEC Console is set based on the AS/400 Alert Type field. A corrected chart is provided in the product document, *Tivoli Enterprise Console 3.6.1 Release Notes*, GI10-8020.

Class *DISCARD* Definitions

Unlike the AS/400 message adapter CDS file, the alert adapter CDS file, as supplied, discards no events.

Supplied Class Definitions

Following the MAP_DEFAULT statement, the supplied AS/400 Alert Classes are defined. A complete listing of these is provided in Chapter 3 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507. The supplied Class definition map the defined codepoints into Classes with related names as shown here.

```
CLASS SNA_Equipment_Malfunction
  SELECT
    1: ATTR(=,$ALERT_CDPT), VALUE(PREFIX,"10"); # 10xx codepoints
END
```

SNA_Undetermined Class Definition

Following the supplied Class Definition statements, the SNA_Undetermined Class is defined as follows.

```
CLASS SNA_Undetermined
  SELECT
    1: ATTR(=,$ALERT_CDPT);
END
```

Any event that was not matched by any previous Class Definition, will match this SELECT statement provided that a codepoint value is set.

5. These formatted events are then filtered based on the FilterMode and Filter statements in the configuration file.
6. Events that are included (or not excluded) based on the above filtering are then sent to the specified SeverLocation and ServerPort using TCP/IP sockets.
7. If the TEC server is not available, the events will be cached based on the values specified for BufferEvents, BufEvtName, Filter, and any FilterCache statements in the configuration file.

When the event arrives at the TEC server, it is processed based on the classes and rules in the active rulebase. The supplied AS/400 Alert Adapter baroc file on the TEC server is tecad_snaevent.baroc. as400msg.baroc. The supplied AS/400 Alert Adapter rules file on the TEC server is tecad_snaevent.rls. These files are automatically compiled into the active rulebase when the event server is installed.

The supplied `tecad_snaevent.rls` file performs duplication detections on SNA Events based upon the `dup_detect` values in the `tecad_snaevent.baroc` file and event correlation based upon specific defined SNA Formats.

There are also copies of these files as shipped on the AS/400 if you prefer to access them from there. Of course, if you customize these files on the TEC server, you may want to FTP the updated files to the AS/400 so the actual version being used will be accessible. The Alert Adapter baroc file on the AS/400 is:

```
/QSYS.LIB/QUSRSYS.LIB/QAYAABRC.FILE/ALRBRC.MBR
```

The Alert Adapter rules file on the AS/400 is:

```
/QSYS.LIB/QUSRSYS.LIB/QAYAARLS.FILE/ALRRLS.MBR
```

Note

We have not attempted to provide a complete listing of the available configuration keywords. Rather, we have tried to identify those necessary for an understanding of how the AS/400 alert adapter functions. We have also tried to mention any of the keywords we will use in the remainder of this section. Refer to the “Adapter Files” sections of chapters 1 and 3 of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, for a complete discussion of the available configuration file keywords as well as a detailed description of the adapter CDS file.

Likewise, we have not attempted to provide a complete discussion of the Class Definition Statement File. Rather, we have provided the basics necessary to understand the AS/400 message adapter as well as the customizations and scenarios to follow. A detailed discussion of the CDS file can be found in the appendix, “Class Definition Statement File Reference”, of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507.

7.3.3 Customizing the TEC AS/400 Event Adapters

In this section, we will discuss some customizations that can be made to enhance the TEC AS/400 Event Adapters. Some of these will include ways to simplify and automate starting the TEC Adapters. It is necessary to remember that, regardless of the method by which you select to start the TEC Adapters, you must ensure that the user profile that will actually issue the Start TEC Event Adapter command, `STRTECAD`, has the necessary authorities as listed in the “Starting the TEC Adapters” sections of chapters 3 and 4 of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507. In general, when running a program to start the TEC Adapter, as discussed in

Section 7.3.3.1, “Writing a program to start the TEC Adapter” on page 308, the user profile that called the program will also be the user profile that issues the `STRTECADP` command. However, this may not be the case if the program uses adopted authority, is called from another program, or is submitted using the Submit Job command. When starting a TEC Adapter as a subsystem autostart job, the user profile specified as part of the autostart job entry will issue the `STRTECADP` command.

Also, take note that there are also specific authorities listed in the Adapter’s Guide that are required to use the End TEC Event Adapter command, `ENDTECADP`.

7.3.3.1 Writing a program to start the TEC Adapter

Writing a short CL program to start the TEC Adapter is often one of the first activities an AS/400 administrator may perform. There are several reasons for this:

1. As shown in the scenario in Section 7.3.5.1, “Editing the configuration file” on page 341, the Start TEC Adapter command requires that the Configuration file path value be supplied in IFS format. This means that, each time the user wishes to start a TEC Adapter, they must type a string similar to the following in the Configuration file path parameter.

```
/qsys.lib/library.lib/filename.file/member.mbr
```

where:

<code>library</code>	Specifies the library containing the configuration file.
<code>filename</code>	Specifies the configuration file.
<code>member</code>	Specifies the member within the file containing the keyword data.

2. Moreover, for OS/400 users who typically specify file information into OS/400 commands as `FILE(library/file) MBR(member)`, using the IFS format can be quite error prone.
3. Finally, after the TEC Adapter had been configured and running for some time, the Configuration File path is easily forgotten. Also, in environments running multiple message adapters, it may not be easy to distinguish the appropriate configuration file path.

Having the configuration file path specified in a CL program helps alleviate these concerns. Figure 166 on page 309 shows the source of a simple example program to issue the `STRTECADP` command.

```

Columns . . . : 1 80
Edit
LPVLIB/QCLSRC
STRLPVMADP
SEU=>
FHT XX
***** Beginning of data *****
0001.00 /*****
0002.00 /X
0003.00 /X STARTING LPV'S TEC MESSAGE ADAPTER
0004.00 /X
0005.00 /*****
0006.00 PGM
0008.00 STRTECADP EVTADP(LPMMSGADP) +
0009.00 CFGFILE(' /QSYS.LIB/LPVLIB.LIB/QMMACFG.FILE+
0010.00 /QMMACFG.MBR')
***** End of data *****
ENDPGM
F3=Exit F4=Prompt F5=Refresh F9=Retrieve F10=Cursor F11=Toggle
F16=Repeat find F17=Repeat change F24=More keys
2 Sess-4 10.69.12.140 HW 27/20

```

Figure 166. STRLPVMADP program source

Note

Refer to Section 7.2.5.3, “Running a CL Program as a Run Program response” on page 243, for a complete example of how to create a CL program.

Once you have compiled your program, you will be able to start the TEC Adapter by simply using the Call Program command, `CALL`.

Notice that it is not necessary to use the Submit Job command, `SBMJOB`, when starting the TEC Adapter. The Start TEC Event Adapter command, `STRTECADP`, itself is written to submit a new job.

7.3.3.2 Setting the TEC Adapter up as an autostart job

Another task that you may wish to consider performing is to set up the TEC Adapters as a subsystem autostart job. This will cause the TEC Adapter to start automatically whenever the selected subsystem starts. Since the system supplied subsystem, `QSYSWRK`, starts automatically when the AS/400 IPLs, this is a good subsystem to use if you want the TEC Adapter to start whenever the AS/400 is IPLd. The steps necessary to do this are outlined in the “Starting the AS/400 Adapter after an IPL” sections of Chapter 3 and Chapter 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507. Note that the first step in this process is to create a program that

will start the TEC Adapter as shown in the preceding Section 7.3.3.1, “Writing a program to start the TEC Adapter” on page 308. Also, you may notice that the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, provides instructions for automatically starting the TEC Adapter as part of the AS/400 Startup Program. But, we prefer using an autostart job over editing the system-supplied startup job. Regardless of which method you select, be sure to assure that the user profile you decide to specify in the USER parameter of the Job Description you create has all the necessary object authorities listed in the command descriptions for the appropriate adapter type. Also, note that TCP/IP must be started before the Adapter is started.

7.3.3.3 Customizing the Adapter Configuration File

You can customize the adapter configuration files to define the behavior of the adapters in your environment. However, note that, as we show in our scenarios, we suggest making changes to copies of these files, thus, leaving the original intact should you wish to start with the original again.

Recall that, within the adapter configuration file, you can specify common adapter keywords, such as ServerLocation and BufferEvents. Additionally, you can also specify keywords specific to the AS/400 Adapter, such as MsgQueue or Filter. Along with these keywords, Filter statements can also be used to define events to be discarded.

Two common customizations made in the AS/400 adapter configuration file are:

1. One customization is using the MsgQueue keyword to monitor message queues other than QSYSOPR.

For example, you may wish to consider monitoring the QSYSMSG message queue. This is an optionally created message queue in the QSYS library that can be used to monitor specific system messages that indicate potentially severe system conditions. Refer to the book, *AS/400 System Operation V3.6.0*, SC41-4203, for more details on this message queue.

Additionally, in some cases, applications may choose to provide their own message file. They may even decide to send messages to a message queue created strictly for their use. If this is the case, you can configure a TEC message adapter to monitor this message queue. In scenario Section 7.3.5.5, “Using TEC to display User Admin UMBO_SKEL1 joblog” on page 371, we will create our own message queue to use.

Recall that, while each message adapter can only monitor one message queue. By creating multiple configuration files specifying other message

queues, you can start a message adapter for each message queue you wish to monitor.

Note

- In like manner, a custom application may have created its own custom alerts and alert table to correspond with its own messages. In this case, the message queue they create will probably already be set up to allow alerts. Therefore, you may wish to create an alert filter to select only these alerts and send them to a data queue, which you can have in an alert adapter monitor. Refer to the example provided of monitoring only the FINERRORS data queue in the discussion of the **Filter** and **FilterDataQueue** alert adapter keywords in section Section 7.3.2.4, “AS/400 Alert Adapter” on page 300.

As with the message adapters, you can run multiple alert adapters, each monitoring a different data queue.

- A common misconception the belief that the AS/400 Message Adapter can be used to monitor any file on the AS/400. Although a message queue named QHST may exist, the AS/400 History Log cannot be accessed using the `DSPMSG` command or the QMHRCVM system API. Therefore, the AS/400 message adapter cannot monitor it. Also, it is not possible for the AS/400 message adapter to monitor a joblog.

We suggest that, in TEC, it is possible to obtain AS/400 History Log data by using a Tivoli Task to issue the Display Log command, `DSPLOG`, and store this output in a file on a system that supports a Logfile Adapter. We foresee that you would need to work out details, such as how often to run the Display Log task to update the file contents.

Additionally, the scenario in Section 7.3.5.5, “Using TEC to display User Admin UMBO_SKEL1 joblog” on page 371, illustrates a way to display Tivoli User Administration UMBO_SKEL1 joblogs from the TEC console. You may be able to adapt this example to display joblogs for other applications.

2. The second customization is filtering out message and alert events using Filter statements. Here are a few example filter statements.

```
Filter:Class=AS400_MSG;severity=HARMLESS
```

The above statement is specified in a message adapter configuration file and will filter out all events of the AS400_MSG class that are of severity HARMLESS. The scenario in Section 7.3.5.1, “Editing the configuration file” on page 341, will use the following filter statement:

```
Filter:Class=SNA_Equipment_Malfunction;origin=1.2.3.4
```

This statement is specified in an alert adapter configuration file and will filter out all events of the SNA_Equipment_Malfunction class that specify an origin 1.2.3.4.

Chapter 9, “AS/400 management scenarios” on page 437, illustrates using filtering in the AS/400 alert adapter configuration file.

7.3.3.4 Customizing the Adapter CDS File

Additionally, you can customize the adapter CDS files. Again, as illustrated in our scenarios, we suggest making changes to copies of these files and leaving the original as supplied.

You can customize the adapter CDS files to do such things as:

- **Add additional DISCARD messages** - For example, let us say you wish to discard any CPF1240 “Job &3/&2/&1 ended abnormally” where &1 is TEST1. This could be done by adding the following to the CDS file:

```
CLASS *DISCARD*  
  SELECT  
  1: $MSG_ID = CPF1240;  
  2: $ARG1= TEST1  
END
```

- **Alter the value of an attribute** - For example, you may wish to map a message ID to a specific TEC Severity. An example like this will be provided in the scenario in Section 7.3.5.1, “Editing the configuration file” on page 341. Other attributes that are frequently alerted include:

- **host** - The host attribute is set based on TCP/IP configuration settings. These values can be viewed and changed using the Change TCP/IP Domain command, CHGTCPDMN. Most often, the host attribute will be something similar to:

```
KONG.SUPPORT.TIVOLI.COM
```

However, if the AS/400 is installed as an Endpoint, the Endpoint name is often made something simple, such as:

```
kong
```

This situation, in and of itself, is not a problem. However, if you wish to run a Task based on the specifications in the TEC event, it is beneficial to set the host attribute to the same name as the endpoint. Refer to the scenario in Section 7.3.5.4, “Responding to an AS/400 Inquiry Message from TEC” on page 353, for an example of this.

- **sub_origin** - The attributes used in Event Group filtering include event class, source, sub_source, origin, and sub_origin. Since the sub_origin

is the only one of these fields that is not included in the MAP_DEFAULT statement in the message adapter CDS file, this attribute can be used to create additional granularity in filtering event groups.

- Add a new attribute and/or a new Class. In Section 7.3.2.2, “AS/400 Message Adapter” on page 282, we discussed an example of defining a Class (AS400_Controller_NotReplying) and an attribute (arg24) for the CPA58E0 message.

Note

If an attribute is added or removed, or if a new Class is defined in the CDS file, you must assure that the TEC rulebase is also updated with matching definitions in the appropriate baroc files; otherwise, the TEC event server will not be able to process the event.

7.3.3.5 Creating custom rules and tasks

Two of the benefits of the Tivoli Enterprise Console include its ability to process events using a rules engine and its ability to perform automated tasks based on event arrival. As mentioned in Section 7.3.2, “Understanding the TEC AS/400 Event Adapters” on page 273, the tecad_snaevent.rls rules file performs duplication detection and event correlation of SNA events. You can, of course, further customize the TEC rulebase to perform additional processing as desired. Refer to the TEC event server documentation for complete details of manipulating the TEC rule base.

As discussed in Section 5.1.1, “Using tasks on an AS/400 Endpoint” on page 99, the AS/400 Task Library provides AS/400 Tasks that can be used from TEC. Additionally, you can add new tasks or even create additional task libraries as we illustrated in Chapter 6, “AS/400 security management” on page 139. In the scenario in Section 7.3.5.4, “Responding to an AS/400 Inquiry Message from TEC” on page 353, we show examples of replying to Inquiry Messages using tasks. In the scenario in Section 7.3.5.6, “Creating an AS/400 Tivoli Inventory menu” on page 382, we use automated tasks to perform an Inventory scan as a response to a message sent from the AS/400. Also, Chapter 9, “AS/400 management scenarios” on page 437, illustrates the setting up of an automated task as a response to an application message.

7.3.4 Troubleshooting TEC AS/400 Event Adapters

This section provides some useful information for troubleshooting and problem determination for TEC AS/400 Adapters.

7.3.4.1 Verifying the TEC AS/400 Event Adapter installation

The T/EC AS/400 Event Adapters are packaged as an AS/400 LPO (Licensed Program Offering). An LPO consists of a base package and option packages (1....n). Each product is packaged with a product ID, version, release, and modification identifier. The T/EC AS/400 Adapters, for instance, have the product ID, 5733TA1, and version, release, and modification identifier of V3R1M0.

For the T/EC Event Adapters for AS/400, the Event Interface Facility (EIF) is packaged as the base portion of the product. The TEC Message Event Adapter for AS/400 is packaged as option 1 and the TEC Alert Event Adapter for AS/400 is packaged as option 2. It is important to note that the installation of any option portion of the LPO requires that the base portion of the LPO be already installed on the system.

Knowing this, we can use the Display Software Resources command, DSPSEWRSC, to verify that the appropriate product options are installed as shown in Figure 167.

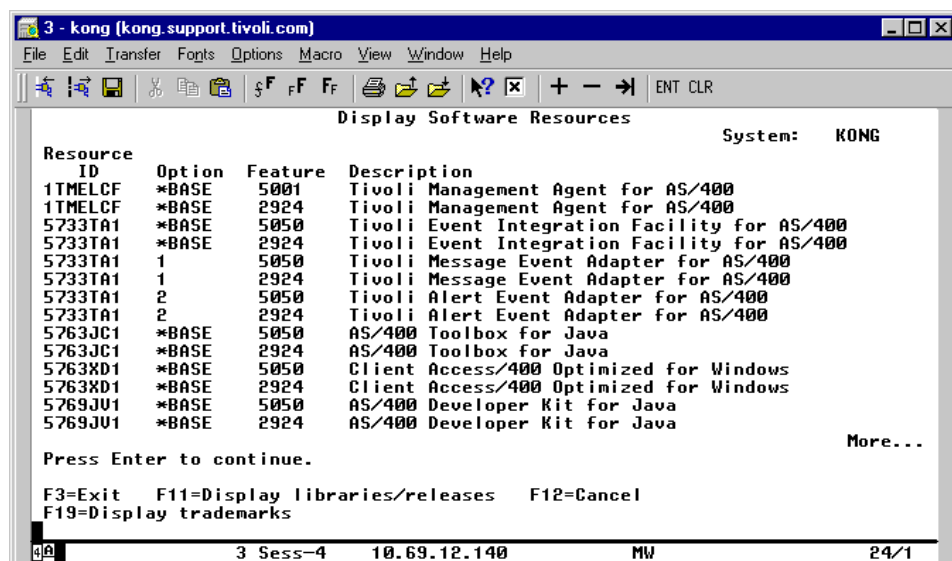


Figure 167. Display Software Resources

The above display shows that all three products (EIF, Alert Adapter, and Message Adapter) are installed. The two feature numbers, 5050 and 2924, differentiate between the program-related objects (5050) and the language objects (2924 = English) for the products. Both features must be installed for the product to function properly.

Also note, as mentioned earlier, that EIF (the BASE option) must be installed on the system before either of the Adapters, options 1 and 2, can be installed.

If these appropriate entries are not present, return to the installation instructions contained in Section 7.3.1, “Installing the TEC AS/400 Event Adapters” on page 272, to install the necessary license program options.

Assuming that the products are correctly installed as illustrated above. You can now use the Check Product Option command, `CHKPRDOPT`, to detect any differences between the correct structure and the actual structure of these products.

`CHKPRDOPT PRDID(5733TA1) OPTION(*BASE)` Will check the EIF option.

`CHKPRDOPT PRDID(5733TA1) OPTION(1)` Will check the Message Adapter option.

`CHKPRDOPT PRDID(5733TA1) OPTION(2)` Will check the Alert Adapter option.

If these commands run successfully, the message

“No errors detected by `CHKPRDOPT`”

will result. If an error is found, you will want to delete the failing option and reinstall it. Refer to the installation and uninstallation instructions provided in Chapter 2 of the *TME 10 Enterprise Console Adapter's Guide*, SC31-8507.

Note

After verifying that the TEC AS/400 Event Adapters are properly installed, remember to verify that any prerequisite PTFs (as discussed in Section 7.3.1, “Installing the TEC AS/400 Event Adapters” on page 272) are also installed. This can be verified using the Display Program Temporary Fix command, `DSPPTF`.

7.3.4.2 STRTECADP/ENDTECADP error messages

If you receive an error after issuing either the Start TEC Event Adapter command, `STRTECADP`, or the End TEC Event Adapter command, `ENDTECADP`, you will want to review your joblog for errors. Before doing so, however, we recommend first issuing a Change Job (`CHGJOB`) command to ensure that all possible messages are being logged:

`CHGJOB LOG(4 00 *SECLVL)`

Following this, attempt the failing command again. Assuming the command fails again, you should review your joblog for errors. A sample of doing this is shown in Figure 168 on page 316.

We begin by issuing the `STRTECADP` command and receiving an error. The error appears at the bottom of the screen from which we entered the command as shown in Figure 168.

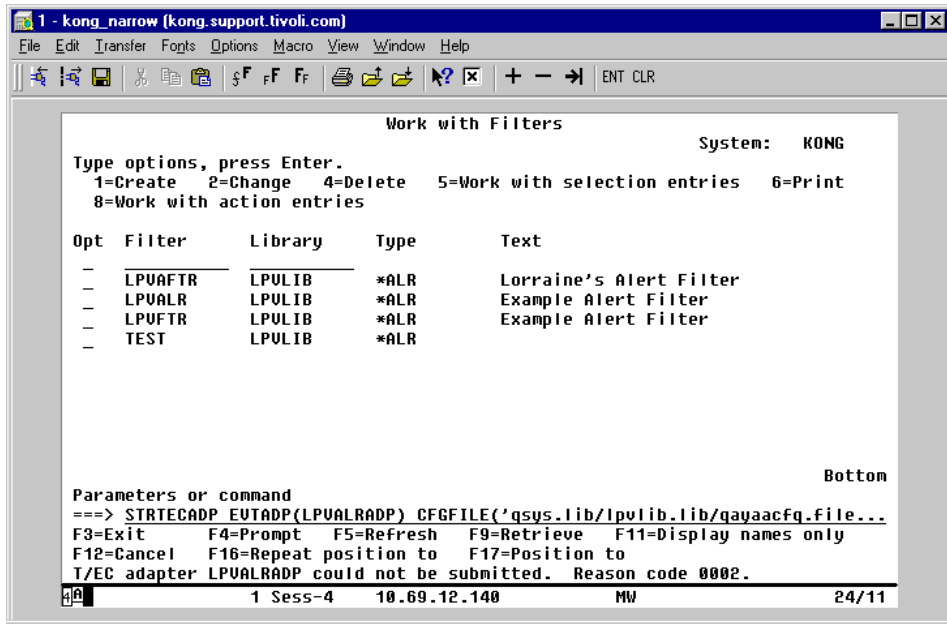


Figure 168. `STRTECADP` command resulting in an error message

With the error message displayed on the screen, move your cursor onto the error message and press the **F1** (Help) key. This will bring up the Additional Message Information panel, shown in Figure 169 on page 317, for the error message.

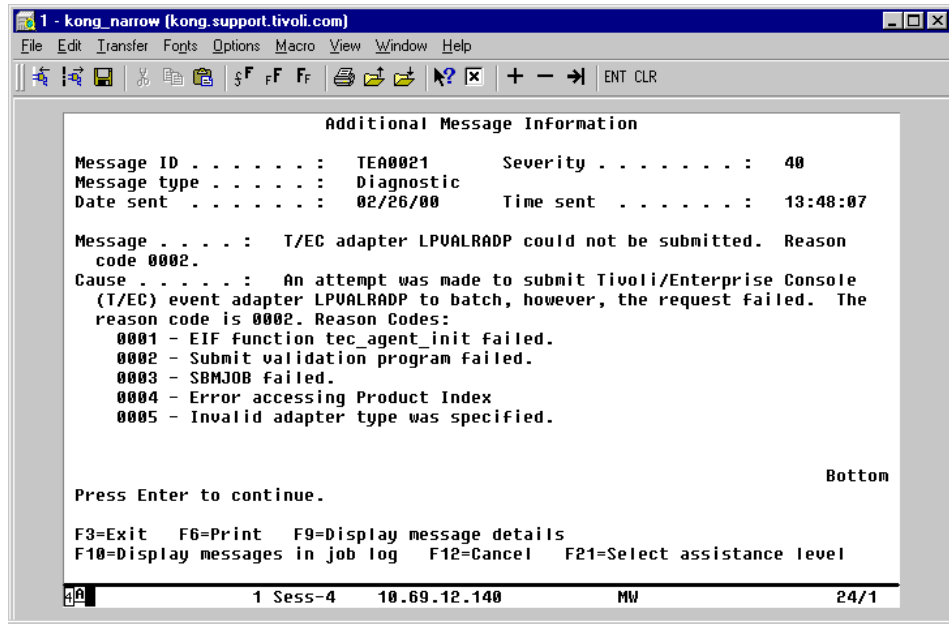


Figure 169. Additional Message Information (TEA0021)

While the message details here may be helpful, oftentimes, there are other previous errors that have resulted in the displayed error message. Press **F10** to display messages in a job log. This will bring up the Display All Messages panel as shown in Figure 170 on page 318.

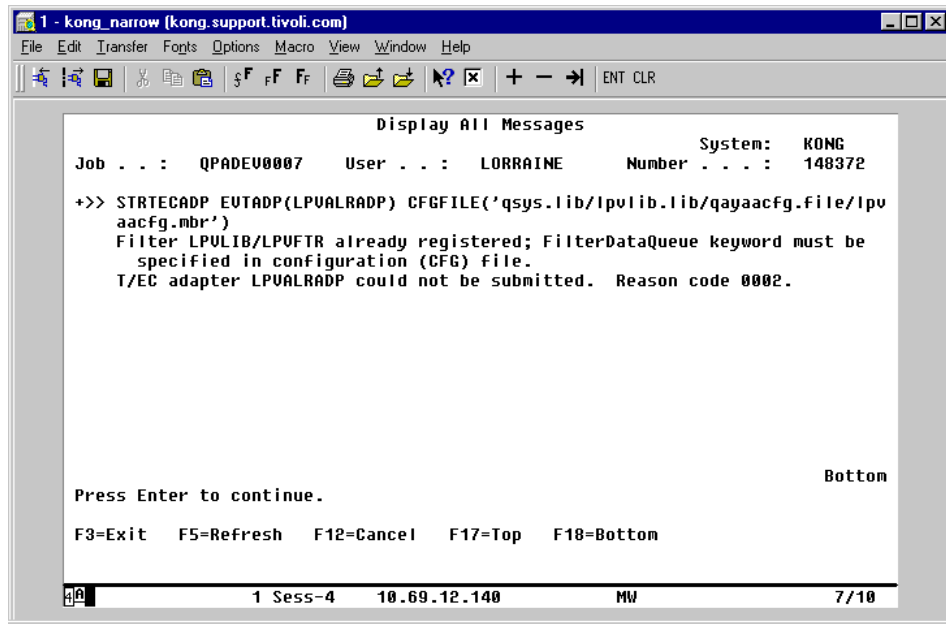


Figure 170. Display All Messages (STRTECADP error)

Notice that, as suspected, there is an error message prior to the TEA0021 message. To bring up the Additional Message Details for this message, place your cursor on it and press the **F1** (Help) key again. The Additional Message Information screen is shown in Figure 171 on page 319.

Note

The above discussion is based on the premise that the `STRTECADP` or `ENDTECADP` commands are being issued in the currently active user's job. If these commands are being run elsewhere, for example, as a subsystem autostart job, you will need to locate and examine the joblog for the appropriate job in which this command is being run. For example, if running this as an autostart job, the user issuing the `STRTECADP` command will be the user profile specified in the `USER` parameter of the Job Description in the Autostart Job entry. Therefore, you can use the Work with Spooled Files command, `WRKSPLF`, to locate the spooled files for this user profile. Specifically, the command issued to find these would be:

```
WRKSPLF SELECT(user_profile)
```

The following Function keys on the Work with Spooled Files screen are very useful, particularly if there are a large number of spooled files.

- F18 (Bottom) will take you to the last screen of spooled files because the spooled files, in general, are listed from first to last.
- F11 (View 1/2/3/4) will display other spoolfile attributes. We find View 2 to be particularly helpful because it provides the spoolfile creation date and time, and this can often be helpful to confirm that the desired joblog is being examined.

In the following list, we provide a sampling of some of the possible message IDs that could be issued in the event of an error condition. We also provide additional common causes where appropriate.

TEA0021 "TEC Adapter &1 could not be submitted. Reason code &2"

Reason Code	0001 EIF function tec_agent_init failed.
Common Cause	The configuration file specified does not exist, or the user is not authorized to use it.
Reason Code	The 0002 Submit validation program failed.
Common Cause	The message queue or alert filter specified to does not exist, or the user is not authorized to use it.
Reason Code	0003 SBMJOB failed.
Common Cause	The job description specified in the adapter configuration file does not exist, or the user is not authorized to it.

Reason Code	0004 Error accessing Product Index
Reason Code	0005
Common Cause	Attempting to start a Message or Alert Adapter without the appropriate License Product Option installed. Refer to Section 7.3.4.1, "Verifying the TEC AS/400 Event Adapter installation" on page 314, for instructions on verifying the TEC AS/400 Adapter installation.

TEA0024 "Submit validation for &1 failed. Reason code &2"

Reason Code	0001 EIF function tec_agent_init failed.
Reason Code	0002 EIF function tec_create_handle failed.
Common Cause	The specified ServerLocation or ServerPort cannot be identified.
Reason Code	The 0003 CDS file could not be opened.
Common Cause	The specified CDS file does not exist or the user profile is not authorized to it.
Reason Code	The 0004 CDS file has syntax errors.
Common Cause	Review the joblog for a TEA0032 message preceding this. This will give further details of the syntax error and its location.
Reason Code	0005 Error using data queue API function.

Note

As shown in the example, be sure to locate error messages preceding these in the joblog. Some of the messages preceding these may be listed below.

TEA0022 "TEC adapter &1 could not be ended. Reason code &2"

Reason Code	The 0001 Event adapter with name &1 is not running.
Reason Code	0002 *ALL is specified but no event adapters are running.

CPD0032 "Not authorized to command &2 in library &3."

Common Cause The user profile does not have the necessary authority to run the STRTECADP or ENDTECADP commands. Refer to chapters 3 or 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507, for a listing of the authorizations necessary to run these commands.

CPD0161 "Not authorized to program &2 in library &3."

Common Cause The user profile does not have the necessary authority to the STRTECADP or ENDTECADP command processing program. Refer to Chapter 3 or Chapter 4 of the *TME 10 Enterprise Console Adapter's Guide*, SC31-8507, for a listing of the authorizations necessary to run these commands.

TEA0324 "Server location &1 not found."

Common Cause No route to the server specified by the **ServerLocation** keyword in the configuration file was found. If there is no name server available, you may need to add the TEC event server in the TCP/IP host table using the Add TCP/IP Host Table Entry (ADDTCPHTE) command as mentioned in the "TCP/IP Considerations" sections in Chapters 3 and 4 of the *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507.

TEA0032 "CDS file syntax error at line &1. Reason code &2."

Reason Code	0001 Invalid MAP_DEFAULT statement.
Reason Code	0002 SELECT keyword must be first element of CLASS.
Reason Code	0003 Invalid SELECT statement.
Reason Code	0004 Invalid keyword specified.
Reason Code	0005 Invalid ATTR operator specified.
Reason Code	0006 Invalid VALUE operator specified.
Reason Code	0007 Invalid pseudo variable specified.
Reason Code	0008 Invalid FETCH statement.
Reason Code	0009 Invalid MAP statement.

Reason Code	0010 Invalid line; expecting MAP_DEFAULT or CLASS entry.
Common Cause	Syntax error at the specified line number in the CDS file. Review the CDS file at this location to correct the error.

7.3.4.3 Adapter job problems

The Start TEC Event Adapter command, `STRTECADP`, actually submits a new job. In fact, if you examine the joblog in which the `STRTECADP` command was run, you will see an informational message, CPC1221, as shown in Figure 172.

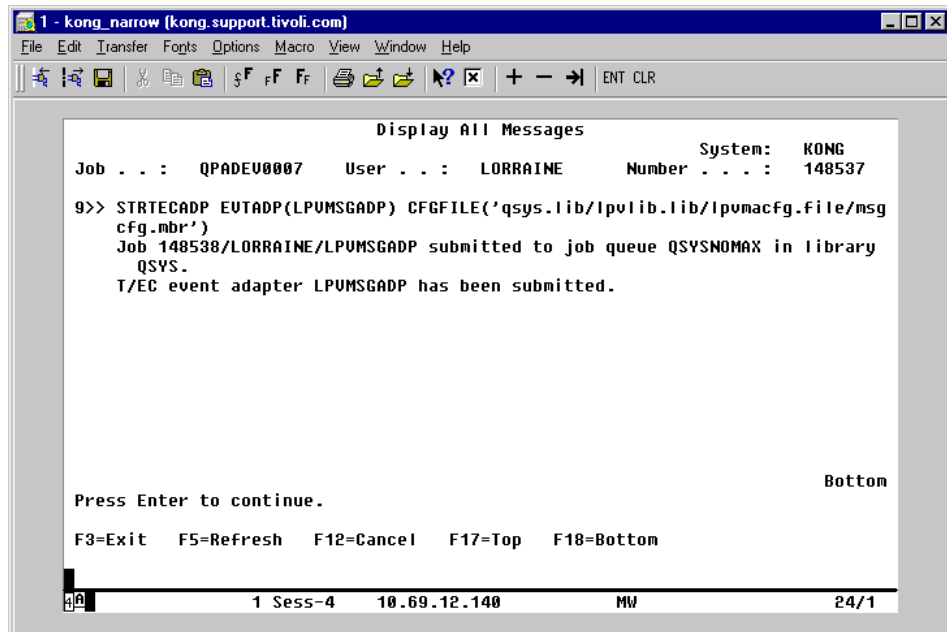


Figure 172. Joblog of a successful `STRTECADP` command

Once this command has been issued, you should be able to locate the adapter job using the Work with Active Jobs command, `WRKACTJOB`. This command will list all the active jobs on the system. By scrolling through the list, you should be able to identify the adapter job running in the `QSYSWRK` subsystem as shown in Figure 173 on page 324.

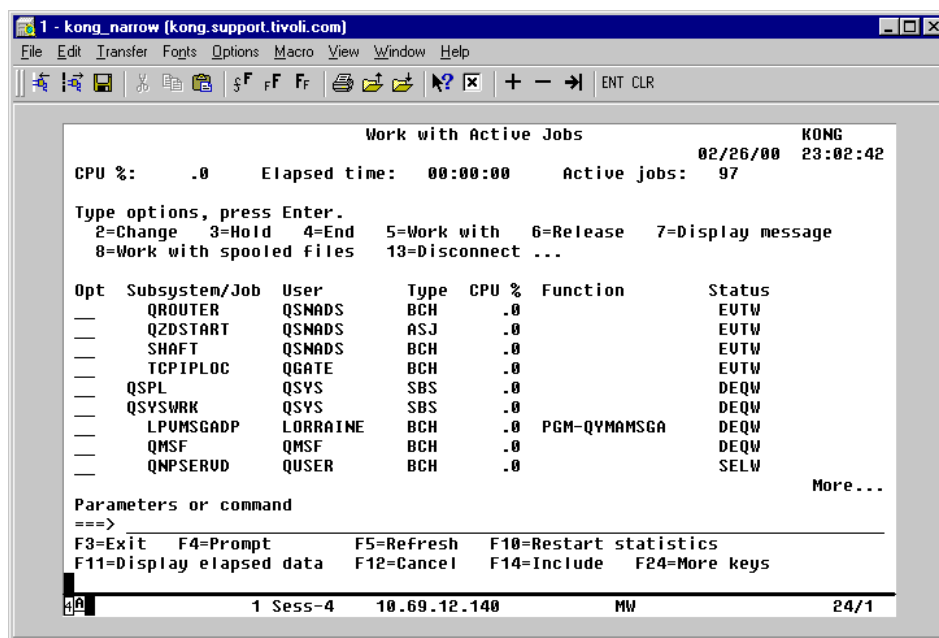


Figure 173. WRKACTJOB showing LPVMSGADP job active

However, if, for some reason, the adapter job ends and cannot be found on the Work with Active jobs display, use the qualified job name, as provided in the CPC1221 message, to easily find the job on the system using the Work with Job command in the following manner:

```
WRKJOB JOB JOB (number/user/jobname)
```

The values specified here are those provided in the CPC1221 message. From the Work with Jobs panel, you are provided various options to determine the status of the job. If the job started and ended for some reason, you should be able to find the joblog and examine it by using option 4, *Work with spooled files*.

If the TEC Event Adapter job is running but not functioning as expected, you should first review its joblog. From the Work with Active Jobs panel, the joblog can be accessed by selecting option 5 (Work with) for the job and then selecting option 10 to display the joblog from the Work with Job panel. Once in the joblog, begin by pressing **F10** (Display detailed messages) to view the complete joblog. You should see a joblog very similar to the one shown in Figure 174 on page 325.

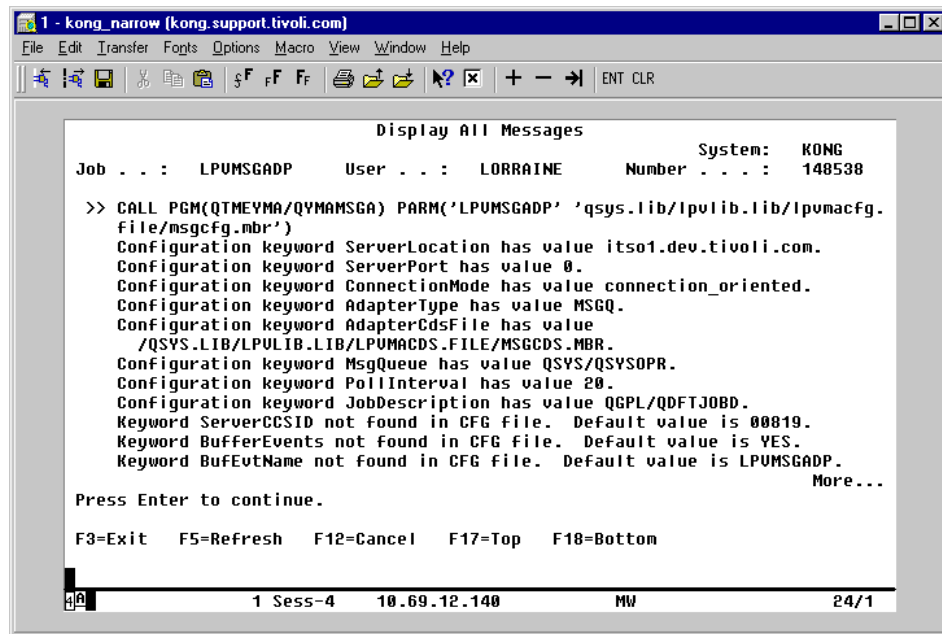


Figure 174. Display All Message (LPVMSGADP)

The joblog will explicitly show each configuration keyword used. Check that you are using the expected configuration and CDS file. It is often the case that changes have been made in a copy of the configuration file, but the `STRTECADP` command (in a program perhaps) still specifies the previous configuration file. In like manner, changes are often made in one copy of the CDS file, but the Configuration File being used to start the TEC Adapter still specifies the previous CDS file.

Once the configuration keywords are verified, continue to review the joblog for any errors. The following list provides a few samples of some of the possible message IDs that could appear in the TEC Event Adapters joblog.

TMA0031 "Event discarded because it did not match any CDS CLASS record; msgid = &1 msgf &3."

and

TAA0031 "Event discarded because it did not match any CDS CLASS record; alert code point"

and

TAA0032 "Alert processing is not turned on (Network Attributes); ALRSTS is &1, ALRLOGST

and

TEA0323 "Syntax error in CFG file. Line number &1."

Cause A syntax error was detected in configuration (CFG) file &2 at line number &1. This error does not prevent the event adapter from starting but may cause undesired behavior.

TEA0325 "Existing T/EC events in the buffer cache are being sent."

Cause A Tivoli Enterprise Console (TEC) Event Adapter found events in the buffer cache. These events will be sent to the event server.

TEA0328 "Unable to buffer T/EC event adapter event."

Cause Tivoli Enterprise Console (TEC) events are not being cached in the buffer event file. Either the buffer event file is being used by another adapter job, or there is a security problem with the buffer file.

7.3.4.4 Events not appearing on the TEC console

Let us continue our discussion of troubleshooting the TEC AS/400 Event Adapters by examining in more detail a situation where events from either an AS/400 message adapter or an AS/400 alert adapter are not appearing on the TEC console. At this point, we assume that the adapter job is running. As discussed in the preceding section, begin by examining the joblog for errors. If there are none, take careful note of the ServerLocation and ServerPort. Assure that these values are set correctly.

If this is a message adapter, verify that the MsgQueue keyword specifies the message queue to which the messages are being sent. If you are unfamiliar with AS/400 message support, you can use the procedure in Section 7.3.4.5, "Generating a Test Message" on page 332, to generate a test message. In like manner, if this is an alert adapter, verify that the Filter and FilterDataQueue keywords specify the data queue to which the alerts are

being sent. If you are unfamiliar with AS/400 message support, you can use the procedure in Section 7.3.4.6, “Generating a Test Alert” on page 332, to generate a test alert.

If desired, you can put the adapter in test mode. In test mode, the events will not be sent to a TEC server. Rather, they are written to an AS/400 database file. By examining the database file, you can verify that events are actually being generated. To place an adapter in test mode, adjust the configuration file to specify:

- TESTMODE = YES
- ServerLocation = *library/filename*

where *library* specifies an existing library, and *filename* specifies the file to which to write the events. If the specified file does not exist, it will be created.

Examine the example of using testmode shown in Figure 175. We first update our configuration file.

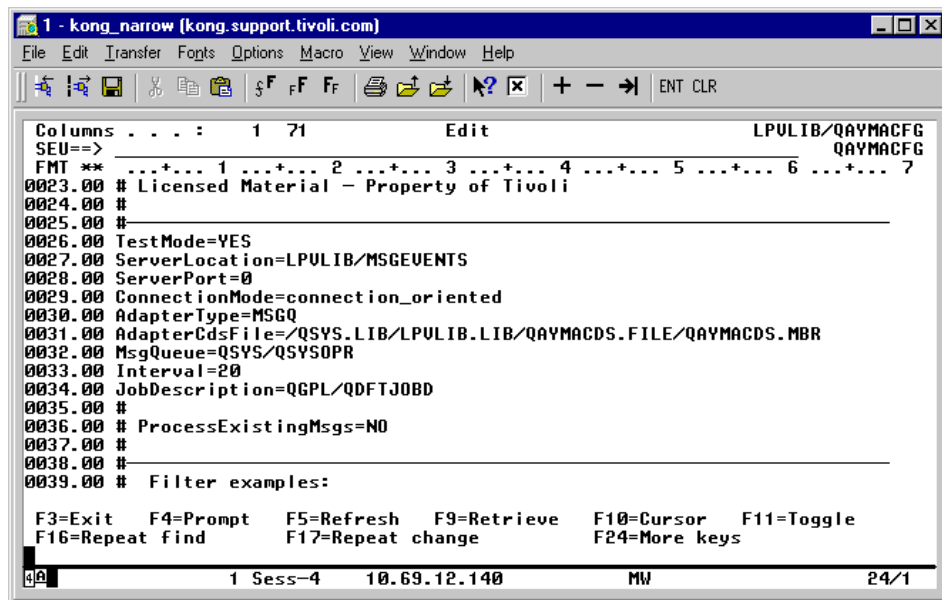


Figure 175. Configuration file specifying TestMode

We must then stop and restart the TEC adapter job by issuing the appropriate End TEC Event Adapter, `ENDTECADP`, and Start TEC Event Adapter, `STRTECADP`, commands. In our case, we issue the following:

```
ENDTECADP EVTADP (LPVMSGADP) OPTION(*IMMED)
```

and

```
STRTECADP EVTADP (LPVMSGADP)  
CFGFILE ('/QSYS.LIB/LPVLIB.LIB/LPVMACFG.FILE/MSGCFG.MBR')
```

Now, we send a few test messages to the QSYSOPR message queue using the following Send Message commands.

```
SNDSMSG MSG('This is test message #1') TOUSR(QSYSOPR)  
SNDSMSG MSG('This is test message #2') TOUSR(QSYSOPR)
```

In actual practice, at this point, you may be repeating some procedure or running a particular application to generate a particular message or alert.

To verify that the message is received at the QSYSOPR message queue, we run the Display Message command.

```
DSPMSG MSGQ(QSYSOPR)
```

This confirms that the messages are on the message queue as shown in Figure 176.

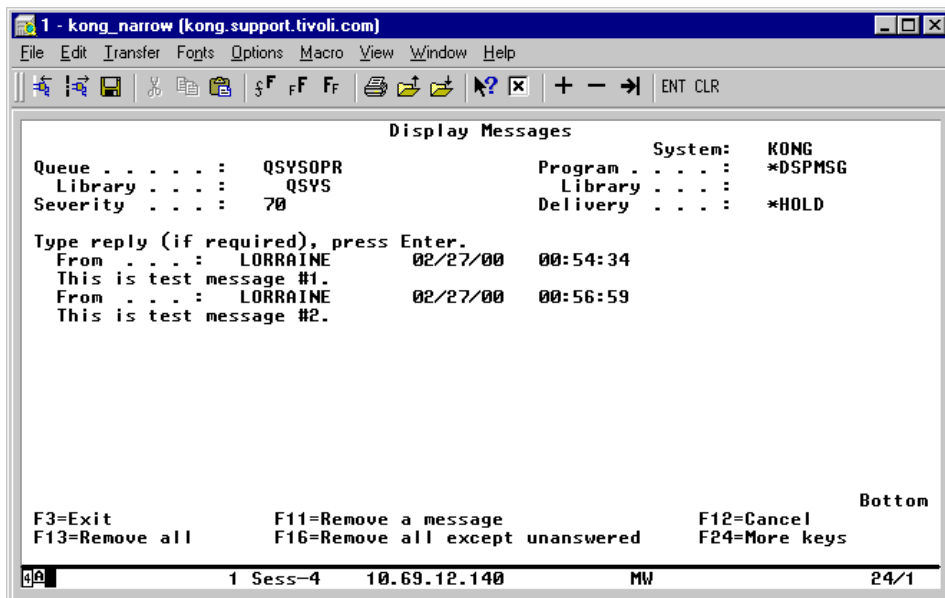


Figure 176. DSPMSG MSGQ(QSYSOPR) test messages

Finally, to confirm that our message adapter is generating alerts from these messages, we issue the following Display Physical File Member (DSPPFM) command:

Our file appears as shown in Figure 177.

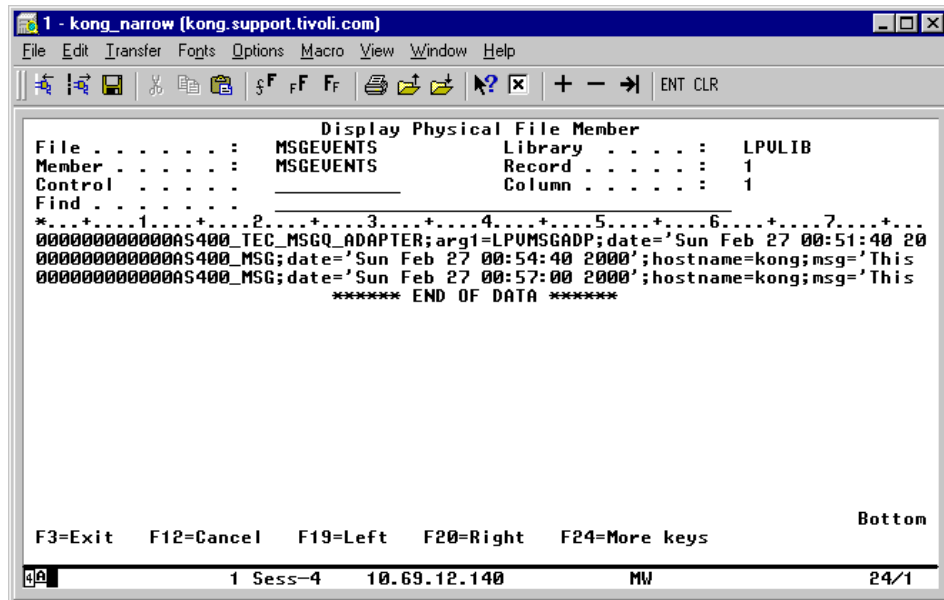


Figure 177. DSPPFM FILE(LPVLIB/MSGEVENTS)

Note that you can use the F20 (Right) key to view additional event attribute values. Nonetheless, the event will be truncated at the record length of the physical file. If the file does not exist beforehand, the record length is set to 266 when it is created.

In addition to using Testmode, the following actions can be performed to assist in troubleshooting the TEC Event Adapters:

1. Consider and check the following networking issues:
 - Is TCP/IP active?
 - Can you ping the TEC event server using the hostname and/or IP address?
 - Do both UDP and IP protocols have entries in the protocol table?
 - Do you have entries in the TCP/IP host table for both the event server and the AS/400?
 - Issue NETSTAT OPTION(*CNN) to verify that connections to the TEC event server port are active and/or ended as appropriate.

2. If this problem involves an Alert Adapter, review and verify that all the necessary settings to generate alerts are set correctly. Refer to Section 7.3.2.4, “AS/400 Alert Adapter” on page 300. Additionally, set the ALRLOGSTS Network Attribute to *ALL so that alerts will be logged. Then, you can verify that alerts are created using the Work with Alerts command, `WRKALR`. Review the alert filter selection entries and action entries using the Work with Filters command, `WRKFTR`. Refer to Section 7.3.2.4, “AS/400 Alert Adapter” on page 300, for details of how these entries will be updated based on the Filter and FilterDataQueue keyword specifications. Verify that the alert filter is created or updated as expected while the alert adapter is active.
3. Check to see if events are being cached using the Work with Object Links command, `WRKLNK`, to locate the buffer file as specified in the BufEvtName keyword. As shown in Figure 178, option 8 can be used to Display the attributes of the buffer file.

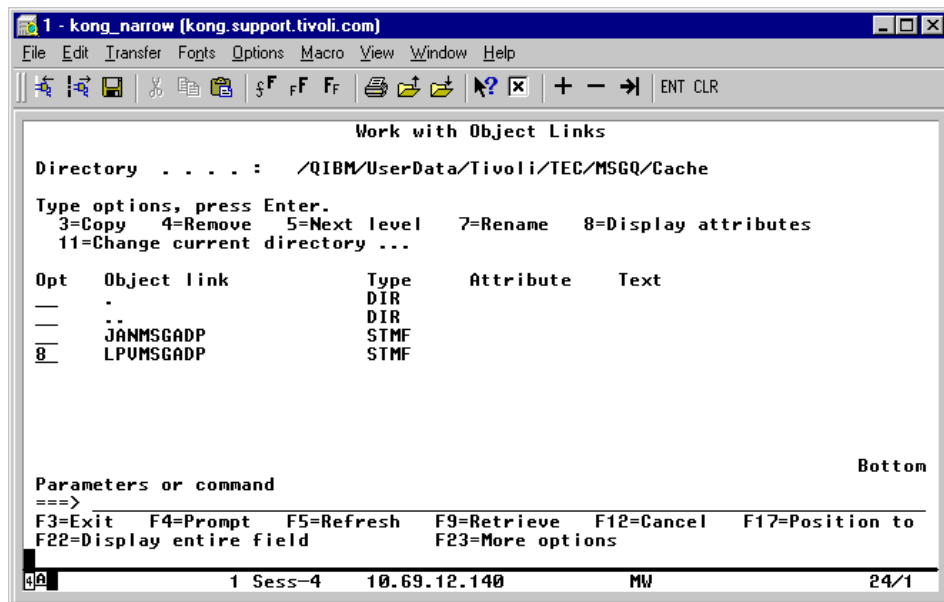


Figure 178. `WRKLNK` of /QIBM/UserData/Tivoli/TEC/MSGQ/Cache directory

If the Size of object data in bytes attribute as shown on the Display Attribute panel in Figure 179 on page 331 is 0, there are no events in the cache file. On the other hand, a value of greater than 0 indicates that events are in the buffer file.

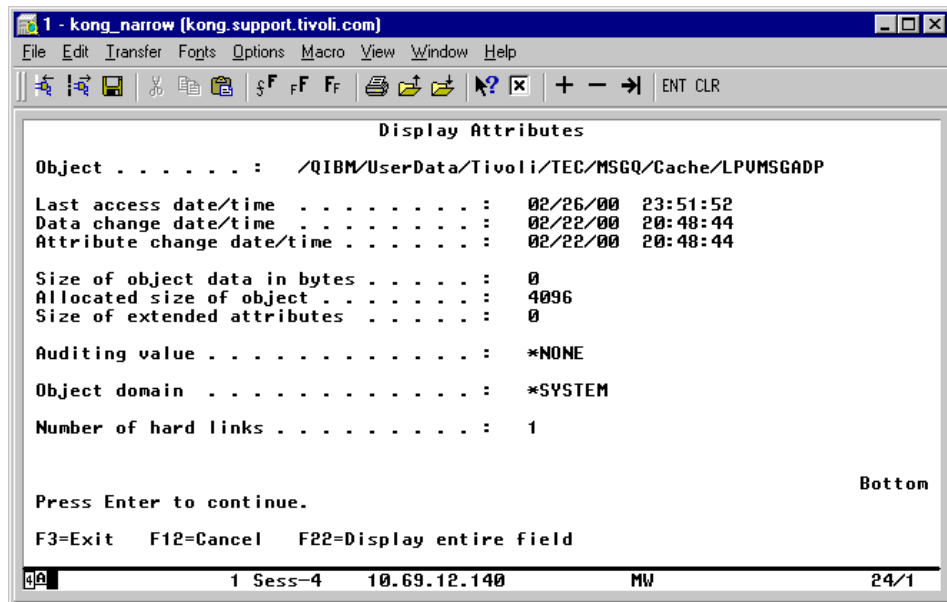


Figure 179. /QIBM/UserData/Tivoli/TEC/MSGQ/Cache/LPVMMSGADP attributes

It is often useful to check whether this value increases or decreases in response to values being changed or actions being taken.

Note

Because the cache file is a stream file, you cannot view it directly by using native OS/400 commands. However, you can view the contents by moving it to another platform type using FTP or by copying the stream file to an AS/400 physical file as shown in Section 5.1.2, “Working with Tivoli Files on the AS/400” on page 106.

4. If all indications show that events are being sent to the TEC event server, but, there are still no events shown on the TEC console, use the `wtdumprrl` command to view the TEC reception log. It may be that the event is actually arriving at the TEC event server. The `wtdumprrl` command will indicate whether the message has been processed successfully or not. The event may be in a waiting or queued state. If the event is being sent to the TEC event server, perform standard TEC troubleshooting techniques to determine why the event is not being displayed on the TEC console. Refer to the *Tivoli Enterprise Console User's Guide* and the *TME 10 TEC Reference Guide 3.6*, SC31-8505, as needed for assistance.

7.3.4.5 Generating a Test Message

Because there are no `wpostmsg` or `postmsg` commands supplied with the AS/400 adapters, you may wish to send a simple impromptu message as a means of testing the AS/400 Message Adapter. From our initial discussion, it should be fairly obvious how to send a message, but we include steps here simply for completeness. We are going to use the Send Message command, `SNDMSG`, to send a message to the message queue which is being monitored by our AS/400 message adapter.

1. Review the configuration file to determine which message queue is being monitored. Look for the `MsgQueue` keyword value:

```
MsgQueue = library/name
```

By default, this will be the AS/400 System Operator message queue, which is the `QSYSOPR` message queue in the `QSYS` library.

2. Run the Send Message command, `SNDMSG`, specifying to send the message to the above message queue:

```
SNDMSG MSG('This is a test message') TOMSGQ(library/name)
```

3. Use the Display Message command, `DSPMSG`, to verify that the above message arrived in the specified queue:

```
DSPMSG MSGQ(library/name)
```

If the OS/400 message adapter is active and monitoring this message queue, the message should now be sent to the TEC event server specified in the configuration file. If the event is not sent, review the previous troubleshooting topics to attempt to resolve the situation.

Note

As mentioned in our previous discussions, you cannot specify a message ID when using the Send Message command. So, if you are experiencing a problem with a particular message not producing the desired results, you can either perform whatever actions result in the message being issued or write a simple CL program to send the message using another command like Send User Message (`SNDUSRMSG`).

7.3.4.6 Generating a Test Alert

Generating a test OS/400 alert may not be as obvious as using the Send Message command to send a message. Nonetheless, here, we will illustrate a fairly simple way of generating a user-created alert on the AS/400 using the Analyze Problem command, `ANZPRB`. Since we are not interested in analyzing or reporting a real problem using this command, we will not provide details on

any of the options you will see on these panels. Rather, we are simply going to illustrate using these as a means of generating an alert.

1. Before generating the alert, use the Display Network Attributes command, `DSPNETA`, to verify that:
 - The Alert status is set to `*ALL` or `*UNATTEND`.
 - The Alert logging status is specified as `*LOCAL` or `*ALL`.

Note

While the Alert logging status value is not necessarily required to generate our test alert, we want to set it in this manner so that the alert will be logged on the AS/400. By logging it on the AS/400, we will be able to see it in the Alert database using the Work with Alerts (`WRKALR`) command and thereby verify that the alert actually was created as expected.

2. Once these values are set as desired, we will create an alert using the Analyze Problem command, `ANZPRBAS`, illustrated in the following screens.
 - a. Enter the `ANZPRB` command and press **Enter**. This will bring up the panel shown in Figure 180.

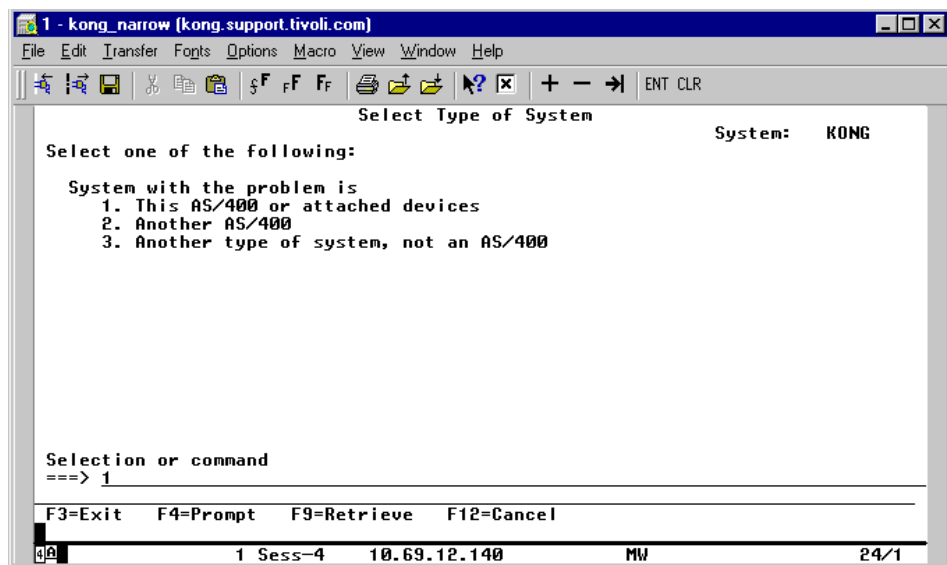


Figure 180. ANZPRP: Select Type of System

- b. Select option **1** and press **Enter**. This will bring up the Analyze Problem panel shown in Figure 181 on page 334.

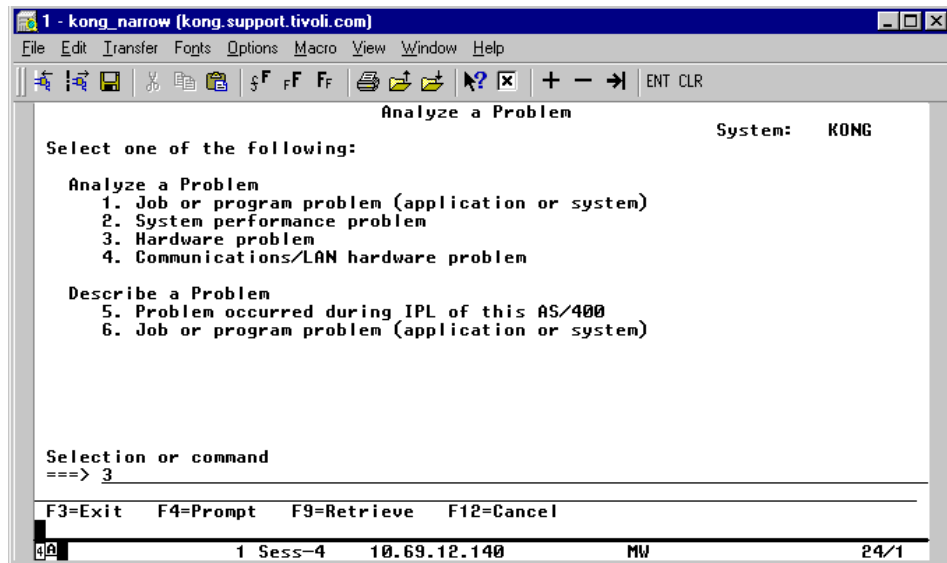


Figure 181. ANZPRB: Analyze a Problem

- c. On this panel, select option **3**, as shown, and press **Enter**. This will bring up the Problem Frequency panel shown in Figure 182.

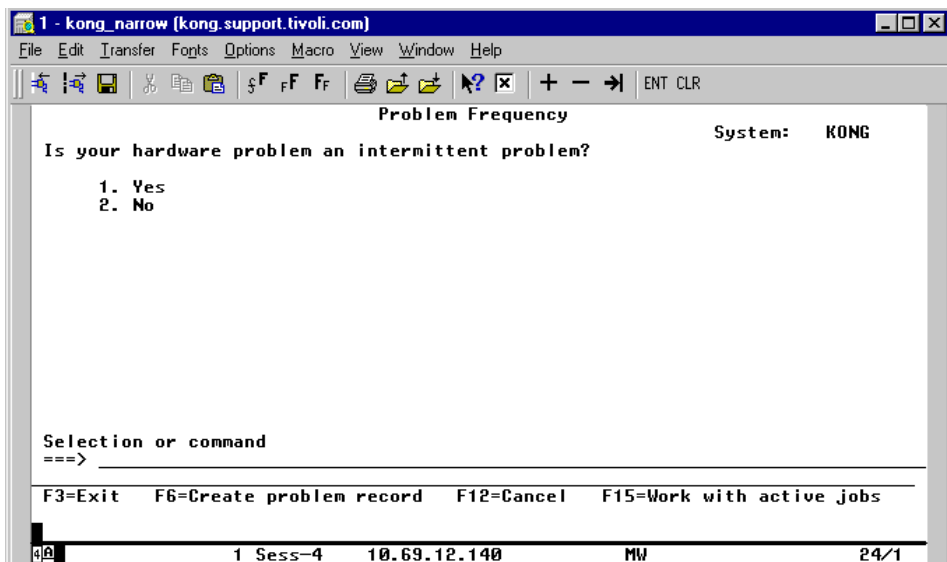


Figure 182. ANZPRB: Problem Frequency

- d. On this panel, press the **F6** (Create) problem record key to proceed to the Define Resource panel shown in Figure 183.

1 - kong_narrow (kong.support.tivoli.com)

File Edit Transfer Fonts Options Macro View Window Help

Define Resource

System: KONG

Note: If a serial number is requested, it must be entered exactly as it appears on the serial number tag. Type zeros if not known.

Type information, press Enter.

Resource:

Type 0000

Serial number 00-0000

Machine type and serial number not recognized.

F3=Exit F5=Refresh F12=Cancel

Function key not allowed.

1 Sess-4 10.69.12.140 MW 24/1

Figure 183. ANZPRB: Define Resource

- e. On this panel, you may enter the type and serial number of your AS/400. If you don't know them, you may enter 0s for these values. If you enter 0s, as we have, you may see the message, *Machine type and Serial number not recognized*, as we show here. If this is the case, you will need to press **Enter** twice to proceed to the Enter Problem Description panel shown in Figure 184 on page 336.

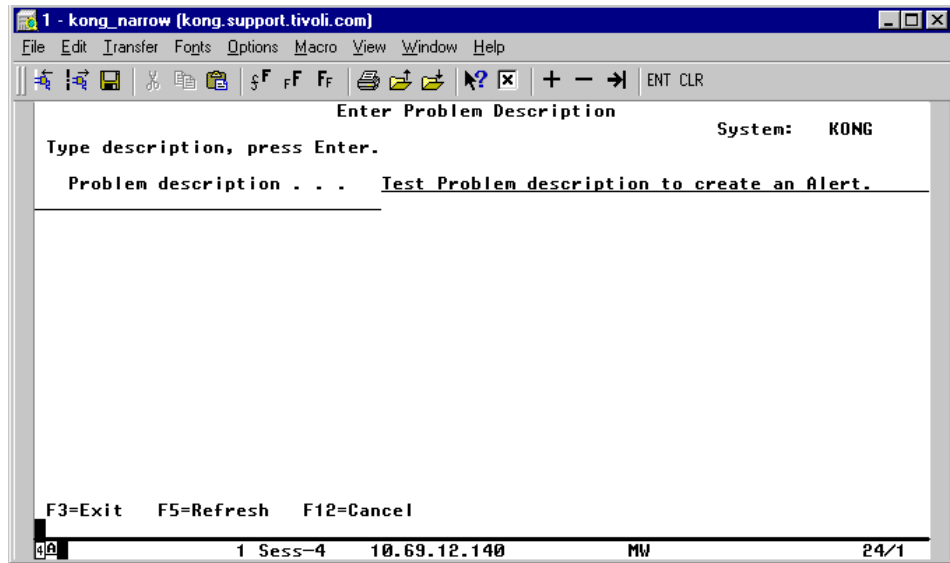


Figure 184. ANZPRB: Enter Problem Description

- f. Enter any text you like on this panel, and press **Enter**. This will bring up the Save Problem Data screen shown in Figure 185.

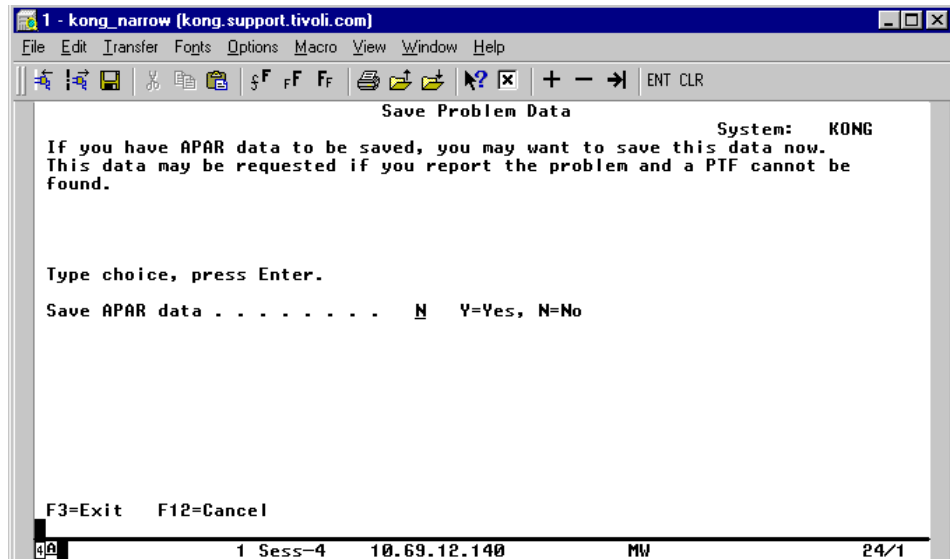


Figure 185. ANZPRB: Save Problem Data

- g. Since we are not reporting an actual problem, specify *N* at the Save APAR data parameter, and press **Enter**. The Report Problem panel shown in Figure 186 will appear.

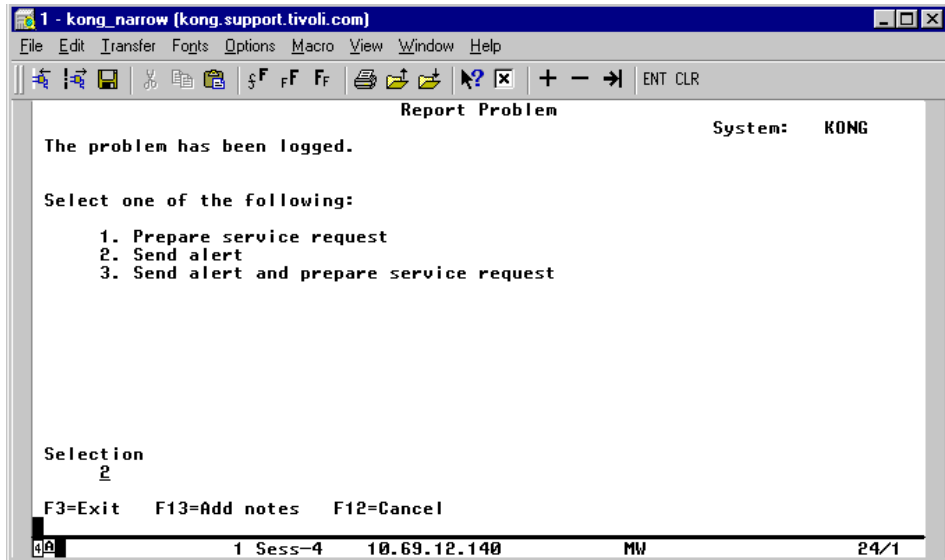


Figure 186. ANZPRB: Report Problem

- h. Select option **2**, as illustrated, to bring up the Define Message Text for Alert panel shown in Figure 187 on page 338.

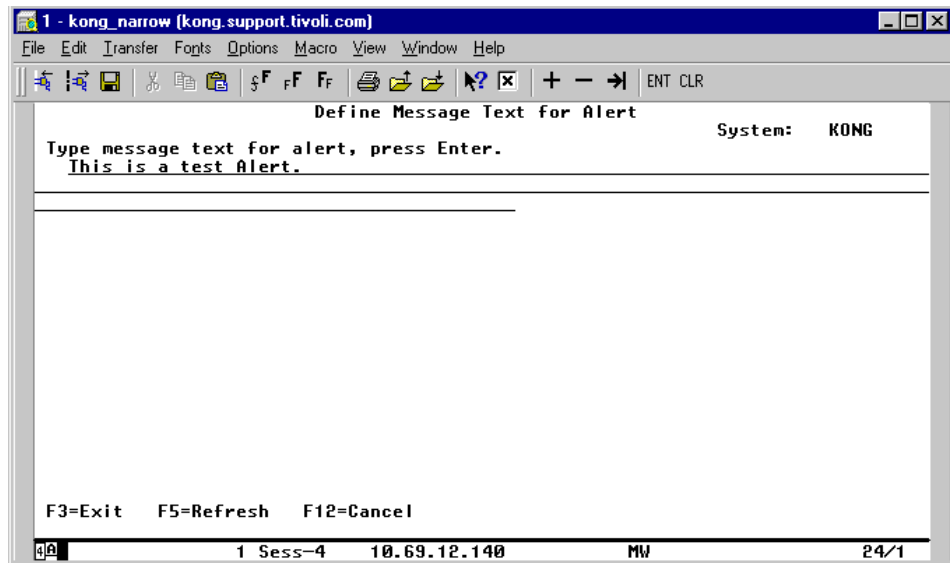


Figure 187. ANZPRB: Define Message Text for Alert

- i. Enter any desired text, as we have shown, and press **Enter**.

At this point, an OS/400 alert will be generated.

3. Issue the Work with Alerts command, `WRKALR`, to verify that your alert has been created as expected. A panel similar to the one shown in Figure 188 on page 339 may appear.

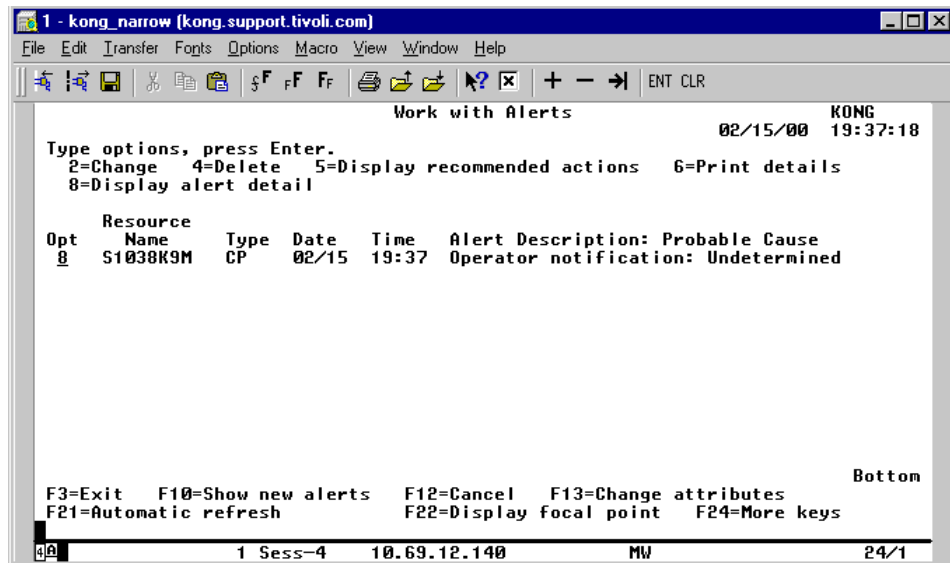


Figure 188. Work with Alerts

Selecting option **8** (Display Alert Details) and then paging down should allow you to confirm which alert is actually the one created from the above actions by displaying the Message text (as entered in Figure 183 on page 335) as shown in Figure 189 on page 340.

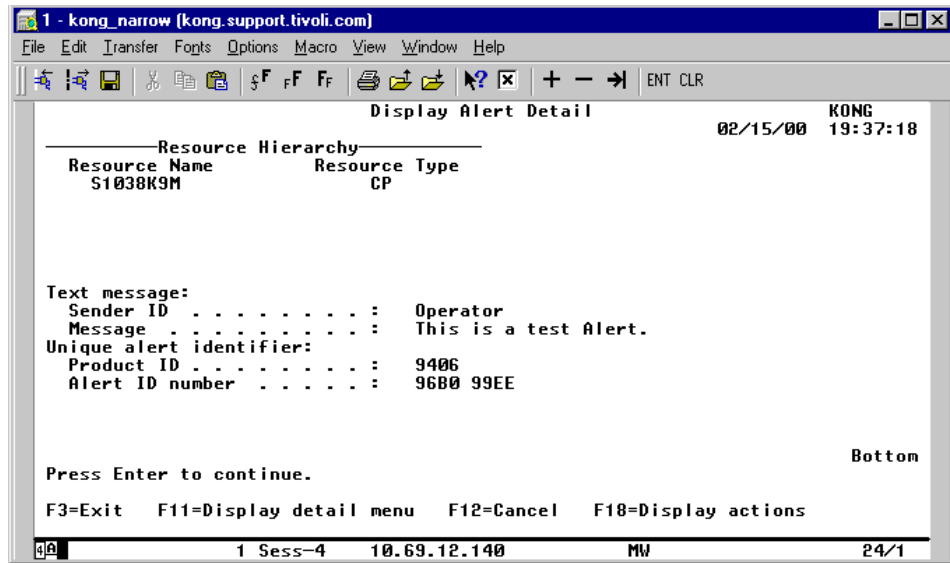


Figure 189. Display Alert Detail (page 2 of 2)

If the OS/400 alert adapter is active and functioning as expected, the alert should now be sent to the TEC event server specified in the configuration file. If the event is not sent, review the previous troubleshooting topics to attempt to resolve the situation.

7.3.4.7 Recreating the Event Adapters Files

Recall that it was suggested that you first make a copy of the supplied event adapter files before editing them. Nonetheless, in the event that you need to recreate these files with their original values, we provide instructions to do this here.

Since a backup copy of the Message Adapter files is contained in the QAYMAMSG file in the QTMEYMA library, you can issue the following Copy File (CPYF) commands to recreate the initial files as supplied in the QUSRSYS library:

```
CPYF FROMFILE (QTMEYMA/QAYMAMSG) TOFILE (QUSRSYS/QAYMACFG) FROMMBR (MSGCFG)
TOMBR (MSGCFG) MBROPT (*REPLACE) CRTFILE (*YES)
```

```
CPYF FROMFILE (QTMEYMA/QAYMAMSG) TOFILE (QUSRSYS/QAYMACDS) FROMMBR (MSGCDS)
TOMBR (MSGCDS) MBROPT (*REPLACE) CRTFILE (*YES)
```

```
CPYF FROMFILE (QTMEYMA/QAYMAMSG) TOFILE (QUSRSYS/QAYMABRC) FROMMBR (MSGBRC)
TOMBR (MSGBRC) MBROPT (*REPLACE) CRTFILE (*YES)
```

Likewise, since a backup copy of the Alert Adapter files is contained in the QAYAAALR file in the QTMEYAA library, you can issue the following Copy File (CPYF) commands to recreate the initial files as supplied in the QUSRSYS library.

```
CPYF FROMFILE (QTMEYAA/QAYAAALR) TOFILE (QUSRSYS/QAYAACFG) FROMMBR (ALRCFG)
TOMBR (ALRCFG) MBROPT (*REPLACE) CRTFILE (*YES)
```

```
CPYF FROMFILE (QTMEYAA/QAYAAALR) TOFILE (QUSRSYS/QAYACDS) FROMMBR (ALRCDS)
TOMBR (ALRCDS) MBROPT (*REPLACE) CRTFILE (*YES)
```

```
CPYF FROMFILE (QTMEYAA/QAYAAALR) TOFILE (QUSRSYS/QAYABRC) FROMMBR (ALBRC)
TOMBR (ALBRC) MBROPT (*REPLACE) CRTFILE (*YES)
```

```
CPYF FROMFILE (QTMEYAA/QAYAAALR) TOFILE (QUSRSYS/QAYARLS) FROMMBR (ALRLS)
TOMBR (ALRLS) MBROPT (*REPLACE) CRTFILE (*YES)
```

7.3.5 TEC AS/400 Event Adapter scenarios

For our adapter scenarios shown here, we have selected to demonstrate using the AS/400 message adapter for continuity between the scenarios. We make reference to other places in this Redbook where similar functions were performed using the AS/400 alert adapter. Also, since the layout and processing of the adapter files is actually quite similar, you can easily adapter the scenarios shown here to use the alert adapter should you have an application that will be generating OS/400 alerts.

As we have suggested several times, we begin by first making our own copy the original AS/400 message adapter configuration and CDS files. Using the Copy File (CPYF) commands shown here, we create LPVMACFG and LPVMACDS files in the LPVLIB. These will be our customized configuration and CDS files respectively.

```
CPYF FROMFILE (QUSRSYS/QAYMACFG) TOFILE (LPVLIB/LPVMACFG) FROMMBR (MSGCFG)
TOMBR (MSGCFG) MBROPT (*REPLACE) CRTFILE (*YES)
```

```
CPYF FROMFILE (QUSRSYS/QAYMACDS) TOFILE (LPVLIB/LPVMACDS) FROMMBR (MSGCDS)
TOMBR (MSGCDS) MBROPT (*REPLACE) CRTFILE (*YES)
```

7.3.5.1 Editing the configuration file

Now, that we have our own copies of these files, we will illustrate editing them. First, we will edit our LPVMACFG configuration file. We want to make the following adjustments:

1. Specify our TEC event server.
2. Specify to use our own CDS file.

- Specify to filter out messages where the severity is Harmless.

Note

In our scenarios, we will be using the Programming Development Manager menu. Selecting 2 (Change) from this menu will bring up the Source Entry Utility as appropriate. Of course, if you prefer, you can use the Start SEU command, `STRSEU`, to go directly into your file from the command line.

If you don't have SEU available, you may use DFU. A simple example of using DFU is shown in Section 7.2.4, "Troubleshooting Tivoli Distributed Monitoring for AS/400" on page 227. You may wish to use FTP to move the file to another AS/400 that does have SEU, edit there, and FTP back to the original AS/400.

If you have numerous AS/400 Endpoints that will all use the same Configuration and/or CDS file, consider using Tivoli Software Distribution for AS/400 Endpoints to distribute this to the AS/400 Endpoint.

To begin, we enter the following Work with Objects using PDM command:

```
WRKOBJPDM LIB(LPVLIB) OBJ(LPVMA*)
```

This command brings up the menu shown in Figure 190.

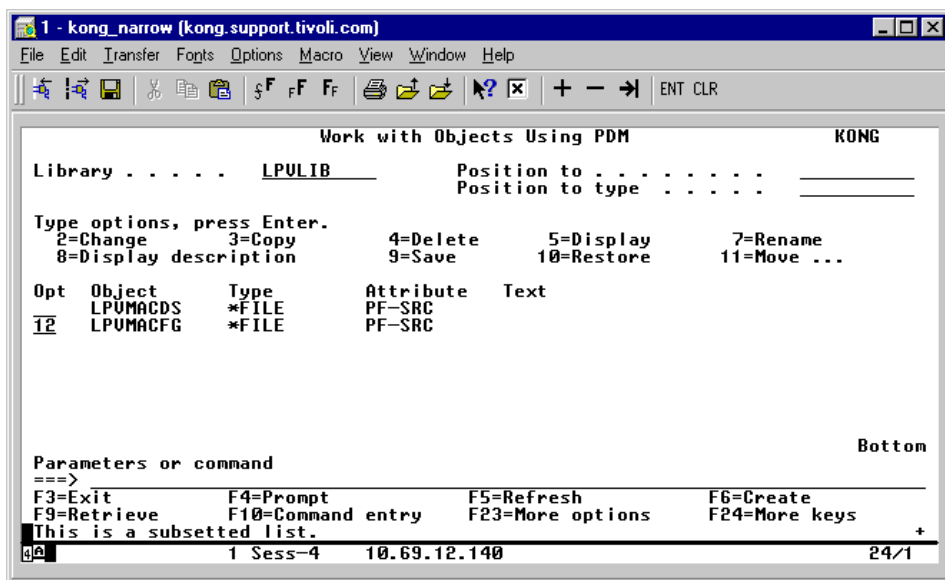


Figure 190. Work with Objects Using PDM (LPVLIB/LPVMA*)

Since we will be editing the configuration file, LPVMACFG, we select option 12 (Work with) beside this file. This brings us into the Work with Members Using PDM screen, shown in Figure 191, where we will see the MSGCFG member in our file. Recall that this member was previously created as part of the Copy File command.

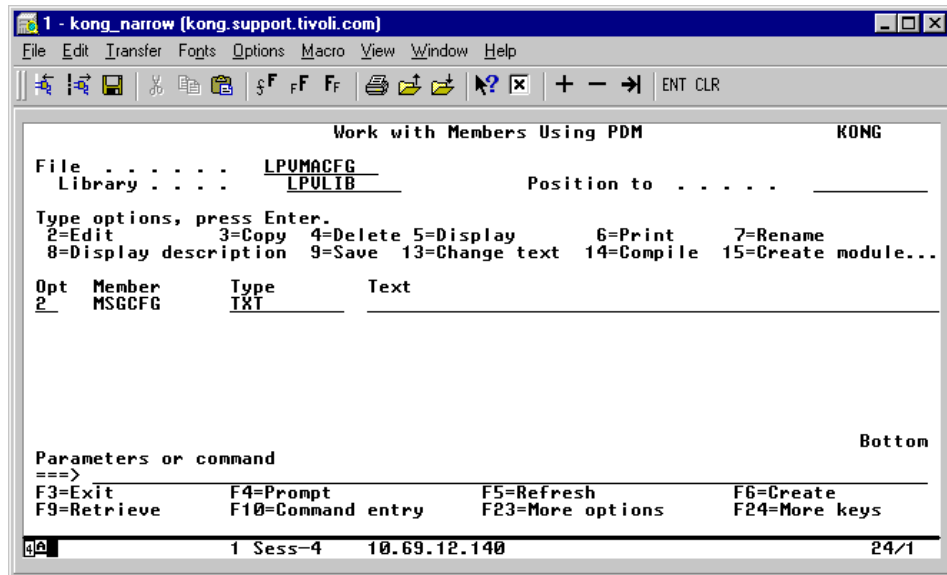


Figure 191. Work with Members Using PDM (LPVLIB/LPVMACFG)

Beside the MSGCFG member, select option 2 (Change). As we mentioned, this will bring you into the Source Edit Utility. We edit the file to appear as shown in Figure 192 on page 344.

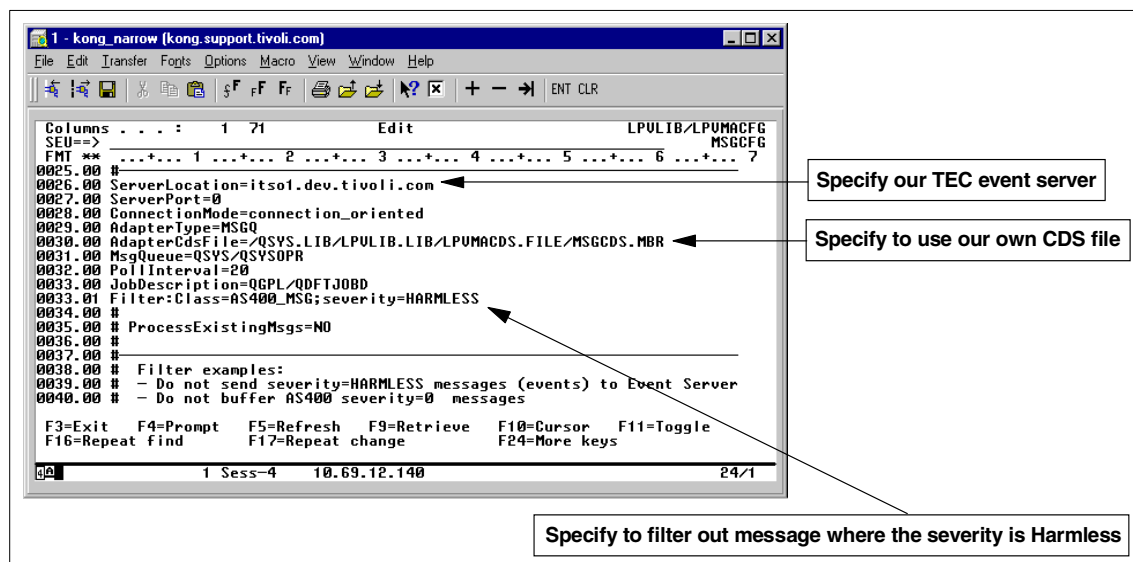


Figure 192. Source Entry Utility (MSGCFG member)

Note the changes made corresponding to our specifications.

Note

As our TEC server has a port mapper available, we can leave the ServerPort set at 0.

When our changes are complete, we press **F3** (Exit), which brings up the Exit panel shown in Figure 193 on page 345.

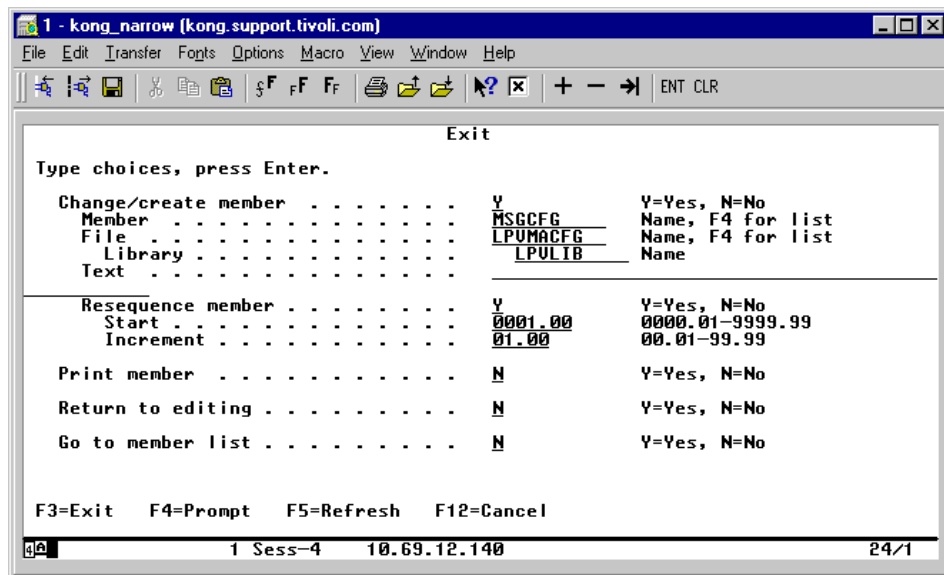


Figure 193. Source Entry Utility Exit (MSGCFG)

We must specify Y at the Change/create member parameter to save our changes.

Note

You can make similar adjustments in the AS/400 alert adapter configuration file. Refer to Section 7.3.2.4, “AS/400 Alert Adapter” on page 300, for a discussion of setting the AlertFilter and AlertDataQueue because these may be keywords that you will want to change to use a customized OS/400 alert filter. Refer to Chapter 9, “AS/400 management scenarios” on page 437, which shows filtering events in the Alert Adapter’s config file.

7.3.5.2 Starting and stopping a TEC adapter

Now that we have set up our message adapter configuration file, we will start our TEC message adapter using the `STRTECADP` command as shown in Figure 194 on page 346.

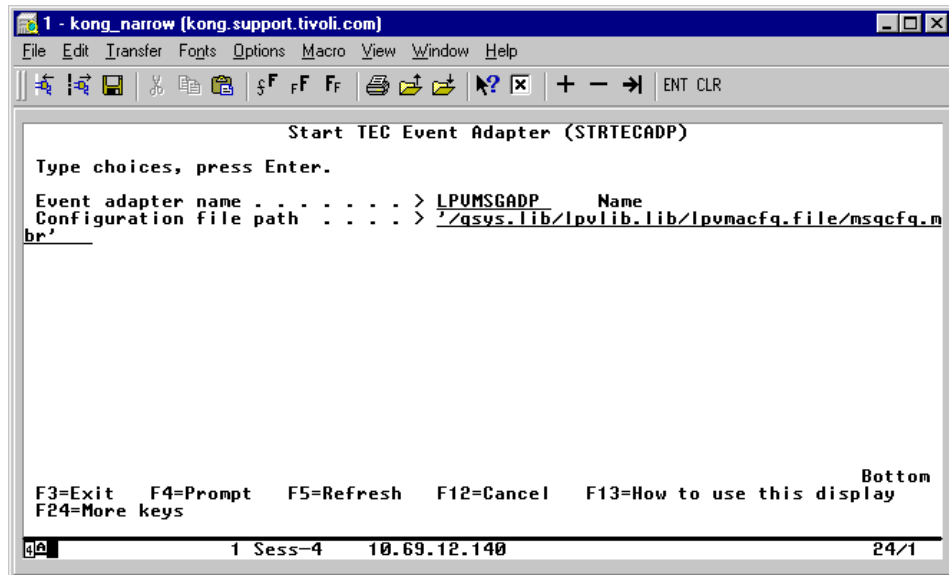


Figure 194. STRTECADP LPVMSGADP

Refer to Section 7.3.3.1, “Writing a program to start the TEC Adapter” on page 308, and Section 7.3.3.2, “Setting the TEC Adapter up as an autostart job” on page 309, for a discussion of creating a program to issue this command and details on setting this up and running it as an autostart job.

We verify that our LPVMSGADP job is running using the Work with Active Jobs () command, `WRKACTJOB`:

```
WRKACTJOB JOB (LPVMSGADP)
```

Recall, from Section 7.3.4, “Troubleshooting TEC AS/400 Event Adapters” on page 313, that we can review the joblog from here to verify that the configuration settings are as we expected and to look for any errors.

Ending a TEC adapter

For completeness, in Figure 195 on page 347, we illustrate the use of the End TEC Event Adapter command, `ENDTECADP`, to end our LPVMSGADP job.

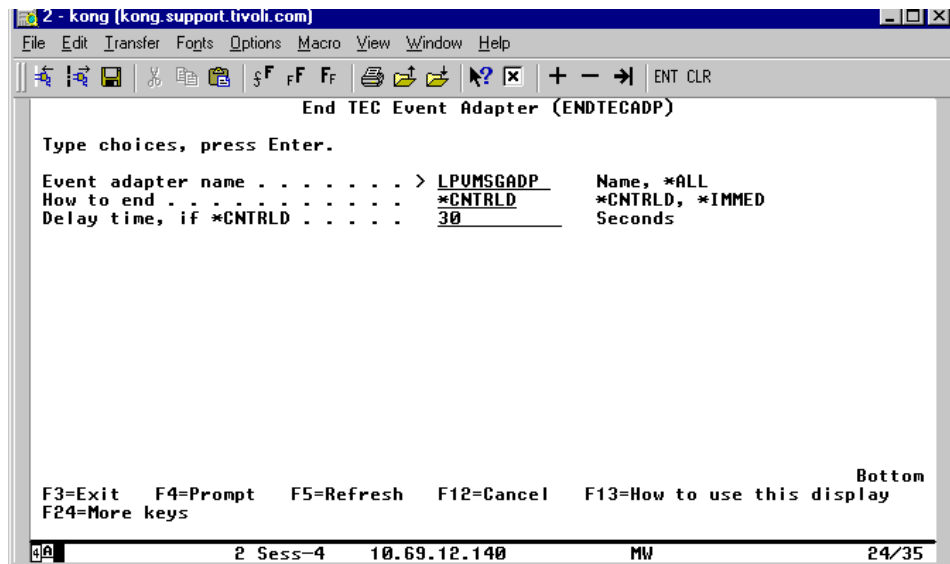


Figure 195. ENDTECADP LPVMSGADP

7.3.5.3 Editing the CDS file

After successfully starting our message adapter, we make the following two observations:

1. Messages CPF0995 “Subsystem &1 ending in progress.” and CPF1103 “Subsystem &1 is starting” are no longer being sent to TEC.

Because both of these messages are, by default, shipped as Severity 0, this is a natural consequence of Filter out Harmless messages. As the section “Event Class Structure” in Chapter 4 of the *TME 10 Enterprise Console Adapter’s Guide Version 3.6*, SC31-8507, shows, messages of severities between 0 and 19 will be Harmless.

2. AS/400 messages with a severity of 99 are not appearing on our TEC console.

This is, actually, the result of a known problem in the current product and the product documentation. If you look carefully at this same place in the *TME 10 Enterprise Console Adapter’s Guide*, SC31-8507, you will notice that severity 99 is not included. Because of this error, severity 99 messages are also mapped to the default value of Harmless.

In this scenario, we will show customizing the CDS file so that CPF0995 and CPF1103 messages will be mapped to Warning. We will also show adjusting

the CDS file so that severity 99 messages will be considered Fatal as appropriate.

Note

The known problem which we mention here is being worked on by Tivoli support as APARs IX89491 and IX89493. IX89491 reports the product problem while IX89493 reports the documentation error. The circumvention here will not be necessary once a patch is available for this.

Also, you might consider changing the severity of the CPF0995 and CPF1103 messages using the Work with Message Description command as shown in Figure 151 on page 275. However, since these are IBM-supplied messages, we suggest using the CDS customizations shown here instead.

The following figures illustrate editing the LPVMSCDS file to make these customizations.

We begin, as we did in the scenario in Section 7.3.5.1, “Editing the configuration file” on page 341, at the Work with Objects using PDM panel by entering the following command:

```
WRKOBJPDM LIB(LPVLIB) OBJ(LPVMA*)
```

This command brings up the menu shown in Figure 196 on page 349.

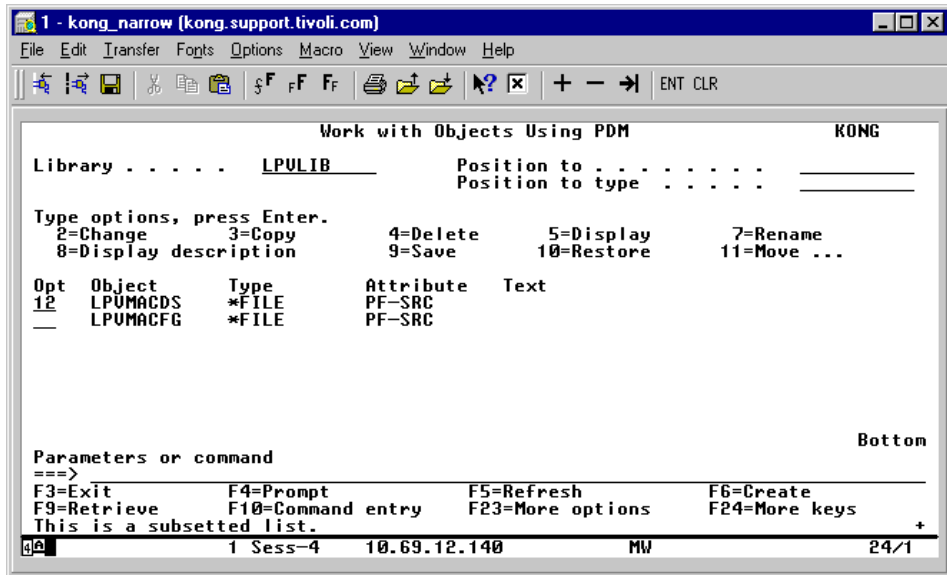


Figure 196. Work with Objects Using PDM (LPVLIB/LPVMA*)

We select option 12 (Work with) on the LPVMACDS file, which brings us into the Work with Members Using PDM screen. The MSGCDS member will be in our file as it was created by the Copy File command we initially ran to begin this section.

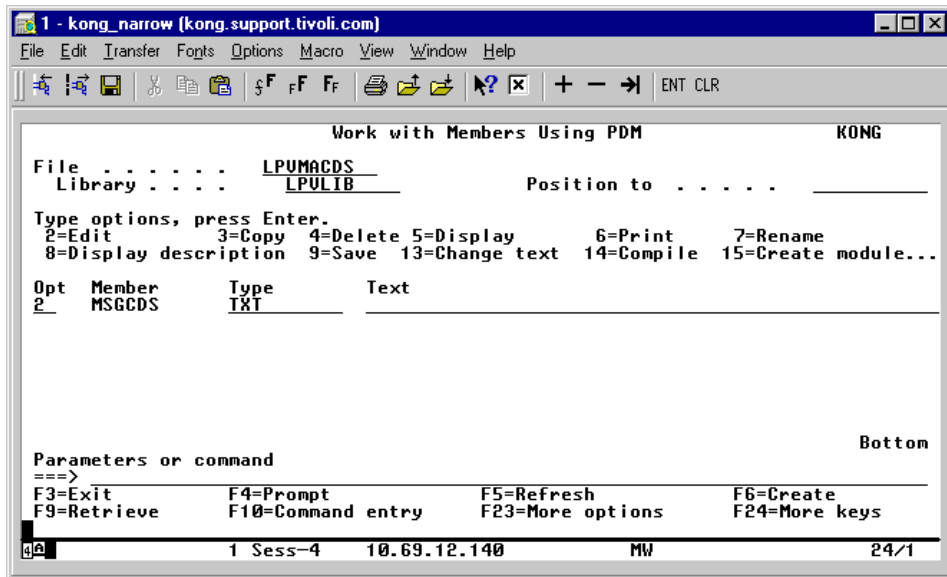


Figure 197. Work with Members Using PDM (LPVLIB/LPVMACFG)

Selecting option 2 (Change) on the MSGCDS member puts us into the Source Edit Utility. We edit the file to appear as follows.

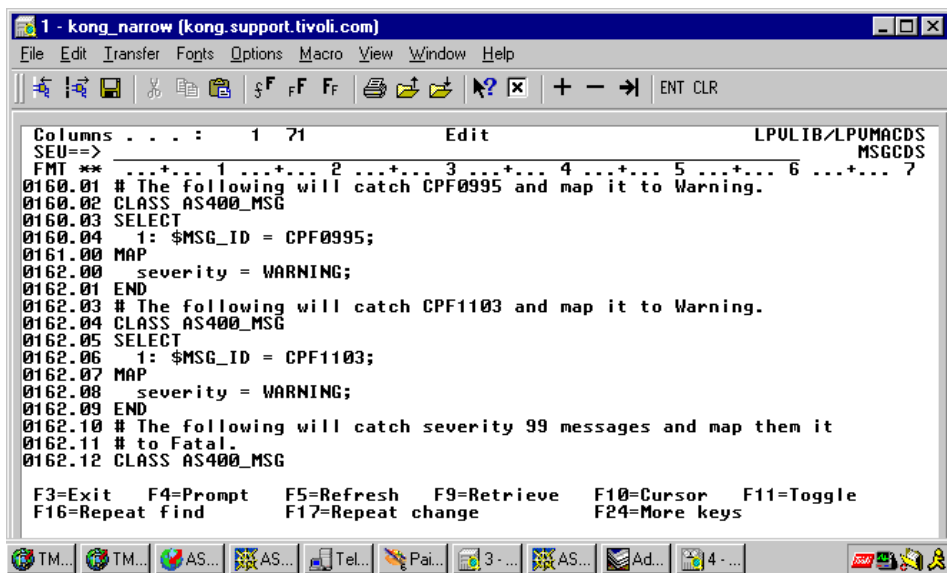


Figure 198. Editing MSGCDS file (1 of 2)

Figure 199 shows the next page of this screen.

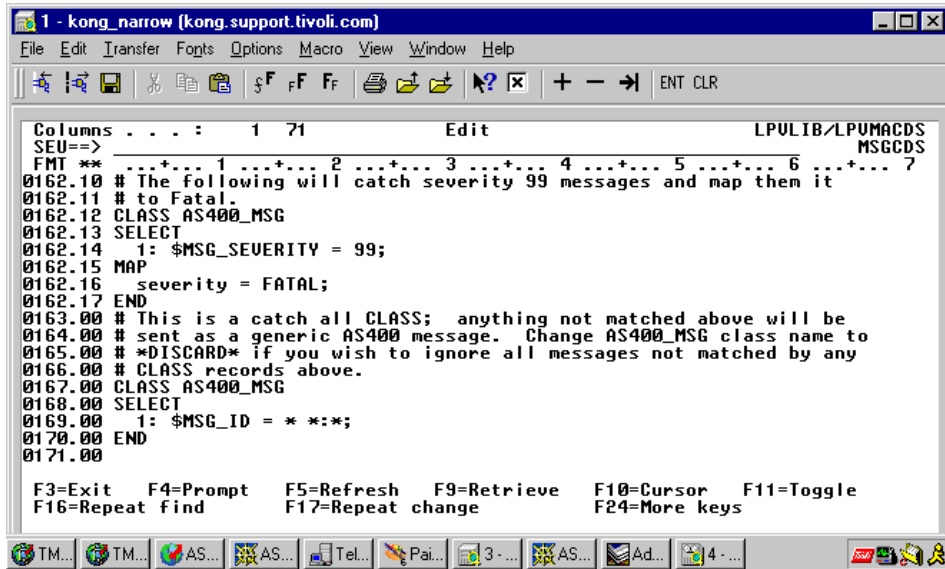


Figure 199. Editing MSGCDS file (2 of 2)

Notice that we place these additions above the *catch all* CLASS as event processing is performed based on the first matching class definition.

When updates are complete, we Press F3 (Exit). On the Exit panel, we assure that Y is specified at the Change/create member parameter.

To make our changes take affect, we must restart the LPVMSGADP job; so, we issue the following commands as illustrated in the scenario in Section 7.3.5.2, “Starting and stopping a TEC adapter” on page 345.

```
ENDTECADP EVTADP (LPVMSGADP) OPTION(*IMMED)
```

and

```
STRTECADP EVTADP (LPVMSGADP)
CFGFILE('qsys.lib/lpvlib.lib/lpvmacfg.file/msgcfg.mbr')
```

To test that, we will now receive our CPF0995 and CPF1103 messages as desired; we use the End Subsystem (ENDSBS) and Start Subsystem (STRSBS) commands to stop and restart the QSNADS subsystem.

Note

Never stop any active subsystem without first checking with the AS/400 System Administrator.

As shown in Figure 200, we observe that both of these messages are again arriving on our TEC console as we desired.

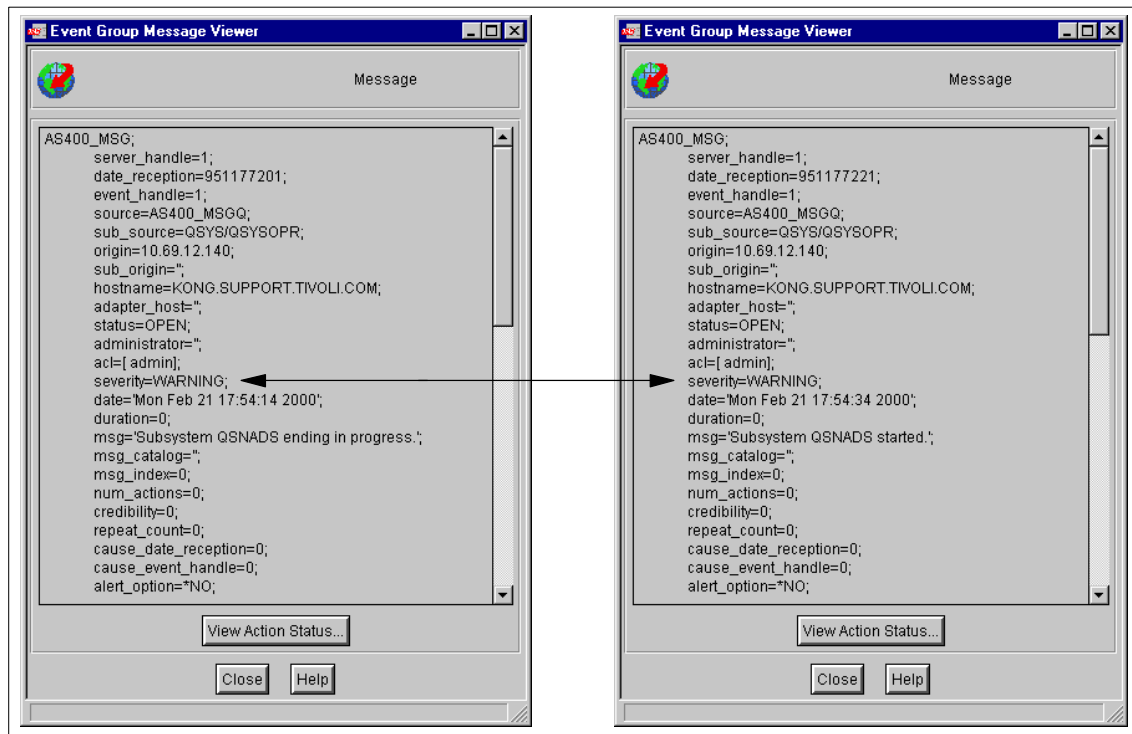


Figure 200. View message CPF0995 (left) and CPF1103 (right)

Additionally, at the next occurrence of a severity 99 message arriving at the QSYSOPR message queue, we note that it arrives on the TEC console with the proper FATAL severity as shown in Figure 201 on page 353.

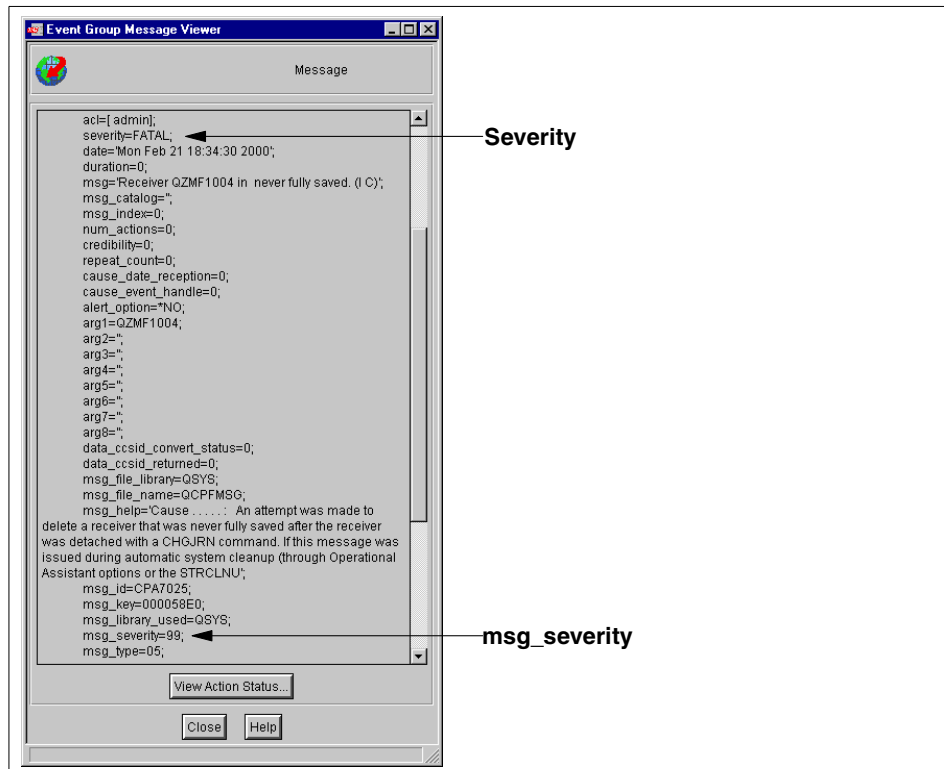


Figure 201. View Message (CPA7025)

7.3.5.4 Responding to an AS/400 Inquiry Message from TEC

At the end of the previous section, we illustrated CPA7025 being sent to QSYSOPR message queue as an Inquiry Message. The message was then sent to TEC console. This brings us to the next scenario in which we will illustrate various ways to reply to inquiry messages that arrive at the TEC console. In all our examples, we will illustrate replying to this CPA7025 message with an *I*.

In each of the following examples, the OS/400 Send Reply command, `SNDRPY`, is being used. As illustrated in Figure 202 on page 354, along with the reply, this command requires that you provide a message key and the message queue to which the reply will be sent.

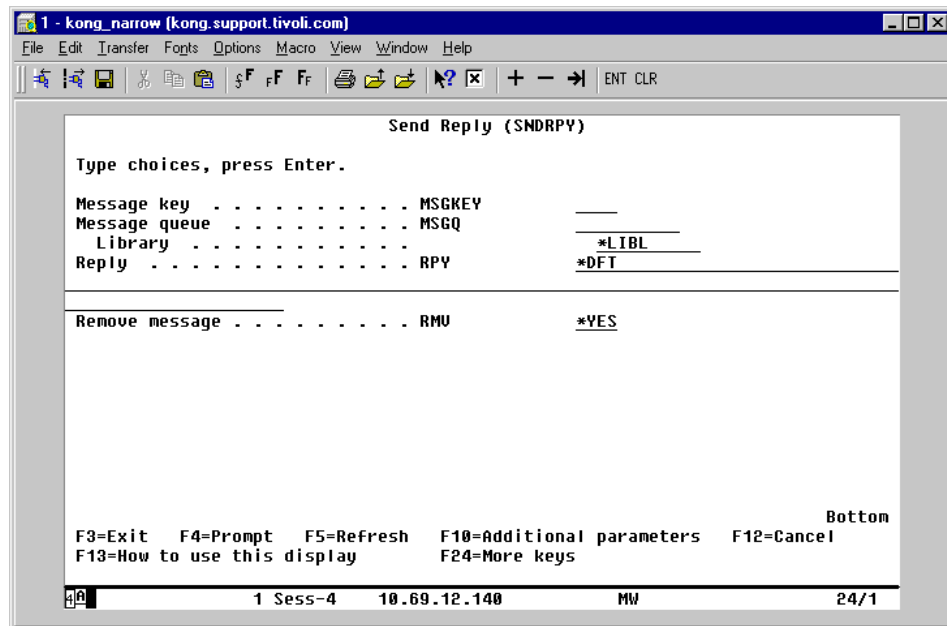


Figure 202. Send Reply (SNDRPY) command and keywords

You can also specify whether to remove the inquiry message from the queue or not. By default, the inquiry message will be removed.

Certain message event attributes become significant when replying to inquiry messages.

- msg_type** An inquiry message has a msg_type of **05**.
- msg_key** This is the message key value in HEX.
- sub_source** Sets to the OS/400 message queue in the form *library/msg_queue*.

In each section, we will begin with a new occurrence of the CPA7025 inquiry message on the TEC console; so, you may note different date/time-related values and/or different msg_key attributes.

Method 1: Replying using FTP

We begin with the CPA7025 inquiry message on the TEC console shown in Figure 203 on page 355.

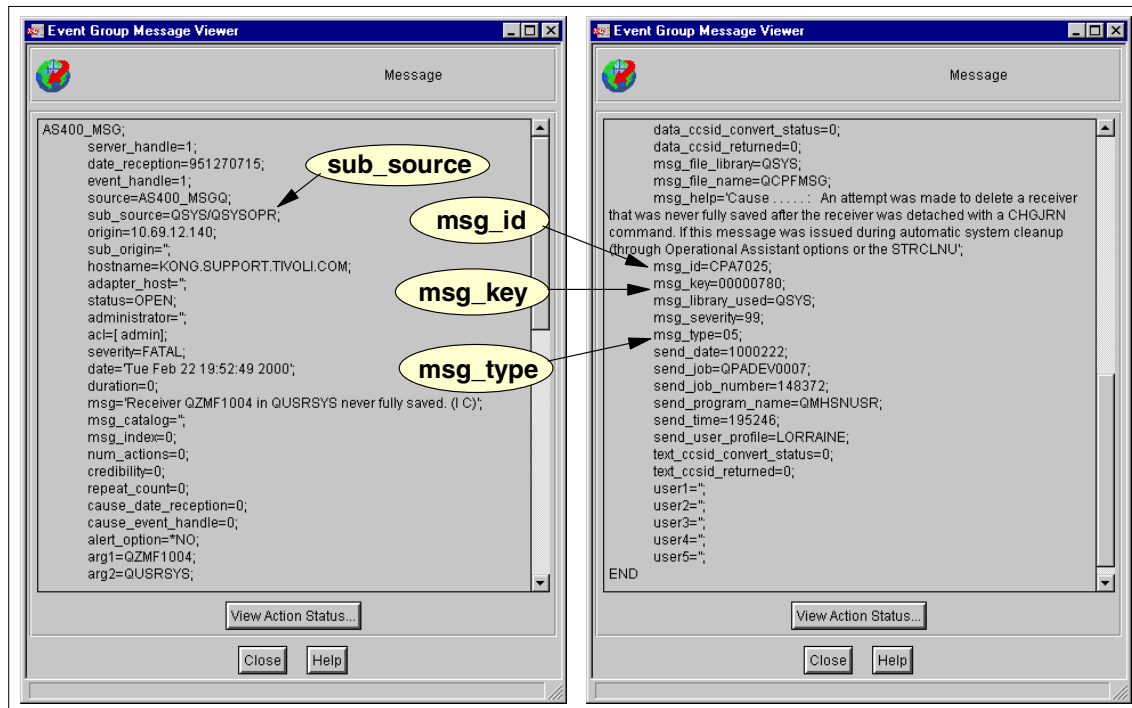


Figure 203. Method 1: CPA7025 inquiry message

If you are only using the TEC AS/400 adapters and do not have the AS/400 defined as an Endpoint in your TMR (recall that the AS/400 adapters are non-TME and, therefore, do not require that the AS/400 be an Endpoint), you can reply to an inquiry message using ftp as shown in Figure 204.

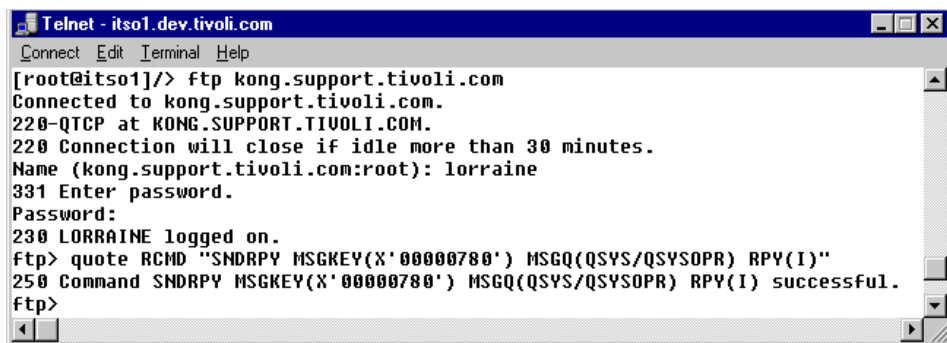


Figure 204. SND RPY command issued in an FTP session

If you plan on replying to messages frequently, you may wish to automate this using a script that takes the necessary Send Reply parameters as input.

Method 2: Using the supplied *Send_Reply* task

In this example, we begin with the following CPA7025 inquiry message on the TEC console as shown in Figure 205.

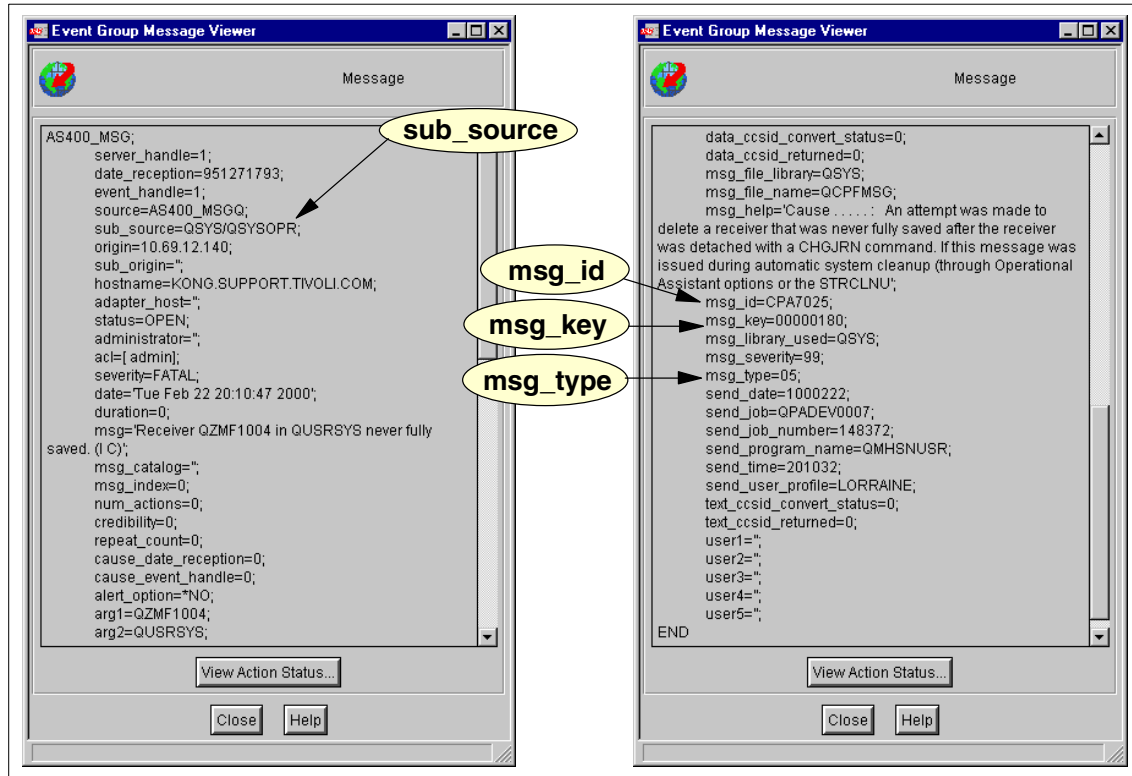


Figure 205. Method 2: CPA7025 inquiry message

If the AS/400 from which this inquiry message has been received is installed as an Endpoint, you can reply to this message using the Send_Reply task provided in the AS/400 Task library. Refer to Section 5.1.1, “Using tasks on an AS/400 Endpoint” on page 99, for a discussion of the AS/400 Task library.

We begin at the TEC Event Console in the AS/400 Messages Source Group. We highlight the message to which we wish to reply and select **Execute on Selected Event** from the Task pull-down as shown in Figure 206 on page 357.

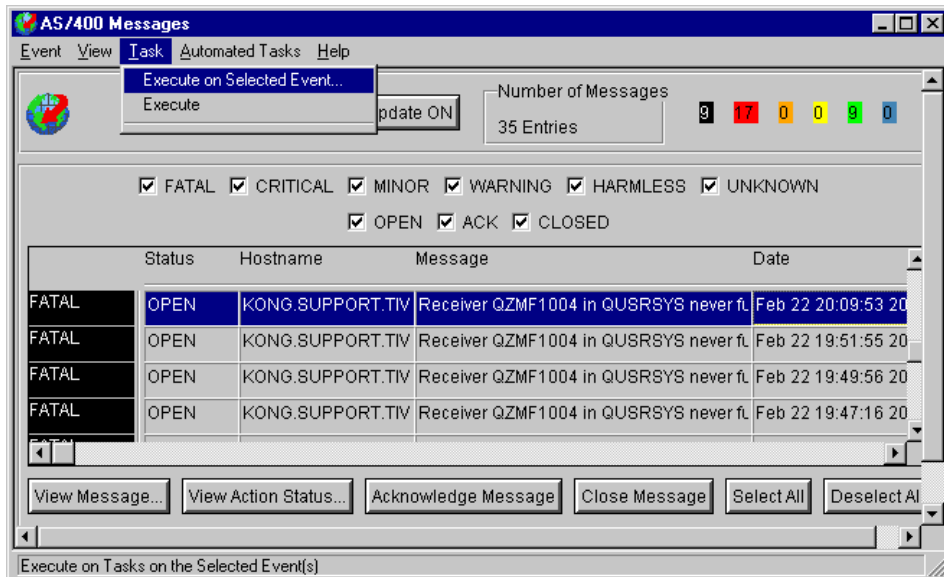


Figure 206. Method 2: Execute task on selected event

Before moving on to the next screen, we wish to mention the hostname attribute. The value of this attribute can be seen on the AS/400 Messages panel here and also on the previous View Message panels.

Of interest to us here is our AS/400 hostname, KONG.SUPPORT.TIVOLI.COM. As shown in this attribute, based upon the TCP/IP configuration values, it is not the same as our Endpoint name. For simplicity, we decided to name the AS/400 Endpoint *kong*. In upcoming panels, the significance of this discrepancy will be pointed out.

That being said, we proceed to the TEC tasks panel where we select the AS/400 task library and the Send_Reply task. We then complete the Send_Reply task arguments using the appropriate msg_key and sub_source attribute values for the Message Key and Message queue library/name arguments. We then provide the I as our reply and leave the remaining parameters as defaulted as shown in Figure 207 on page 358.

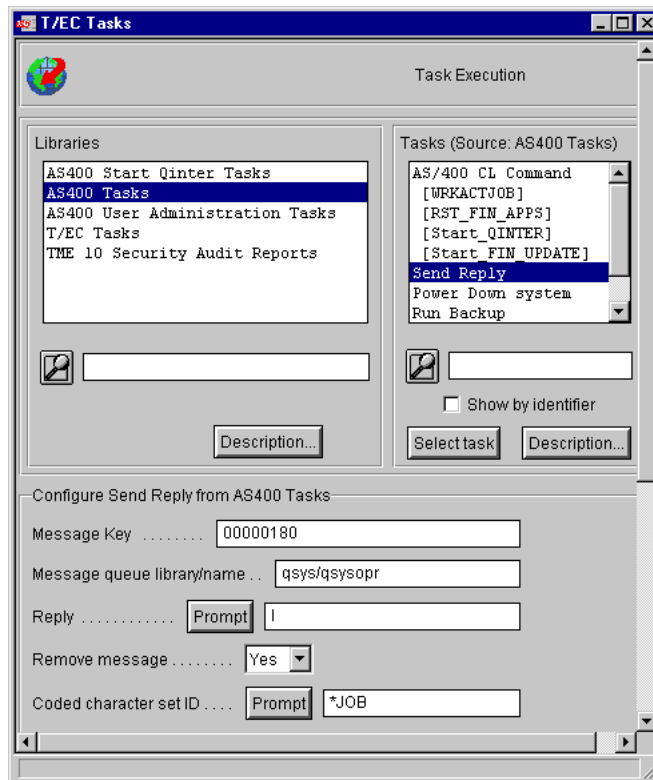


Figure 207. Method 2: TEC tasks (1 of 2)

Now, at the bottom of the TEC tasks panel, we must specify where to execute the specified task. Three choices are available:

- Execute on Selected Events
- Execute on Selected Managed Nodes
- Execute Local

Execute on Selected Events would result in attempting to run the task on the endpoint specified by the hostname attribute. However, as we pointed out, our hostname attribute and our endpoint name are not the same; so, the task cannot be completed successfully using this option.

Clearly, we do not wish to execute the Send Reply command on the local machine; so, we must specify the Execute on Selected Managed Nodes option as shown in Figure 208 on page 359.

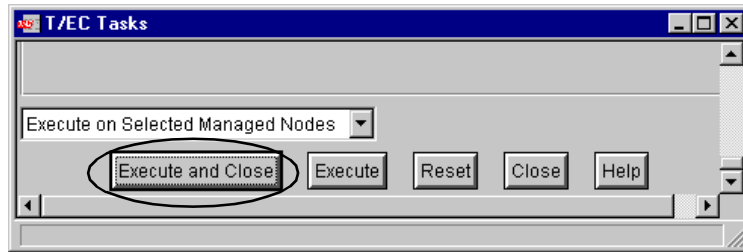


Figure 208. Method 2: TEC tasks (2 of 2)

From the Select the Managed Nodes panel, we select our Endpoint, kong, as shown in Figure 209.

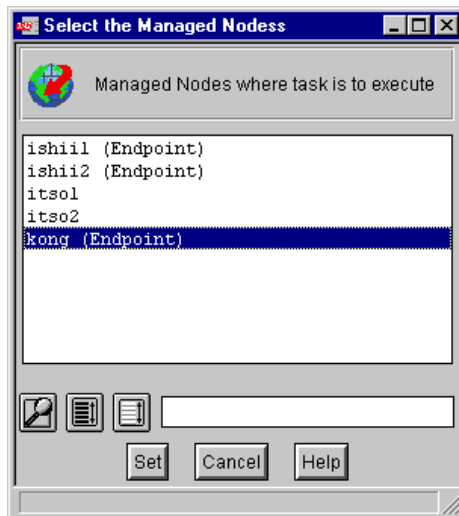


Figure 209. Method 2: Select the Managed Nodes

Press **Set**. This causes the Send_Reply task with the specified parameters to be run on the Endpoint, kong. The task output panels in Figure 210 on page 360 show that our task has completed successfully.

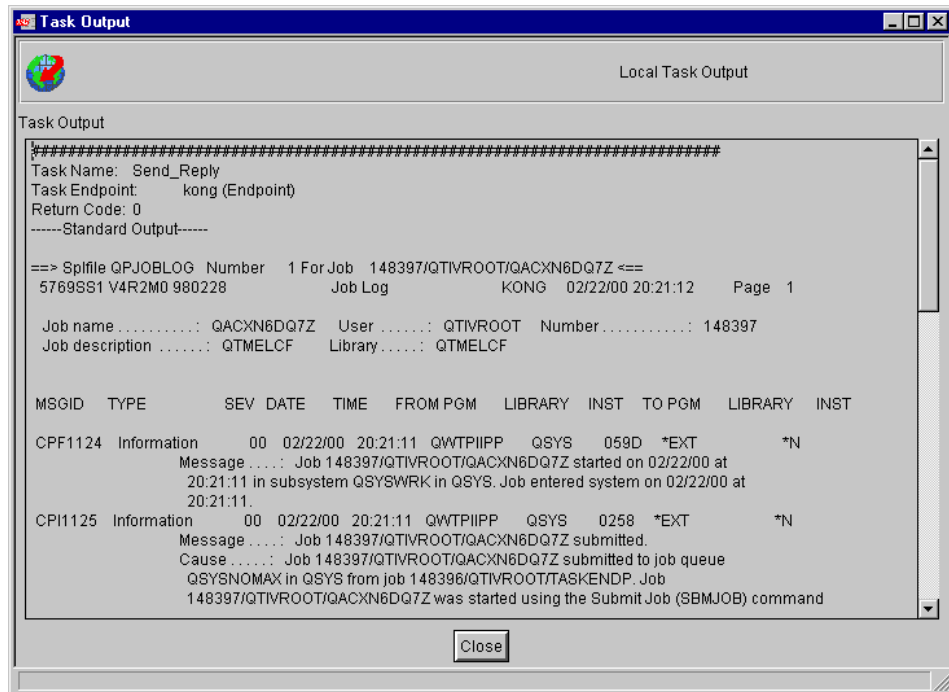


Figure 210. Method 2: Task output (1 of 2)

Figure 211 on page 361 shows the next page of the screen.

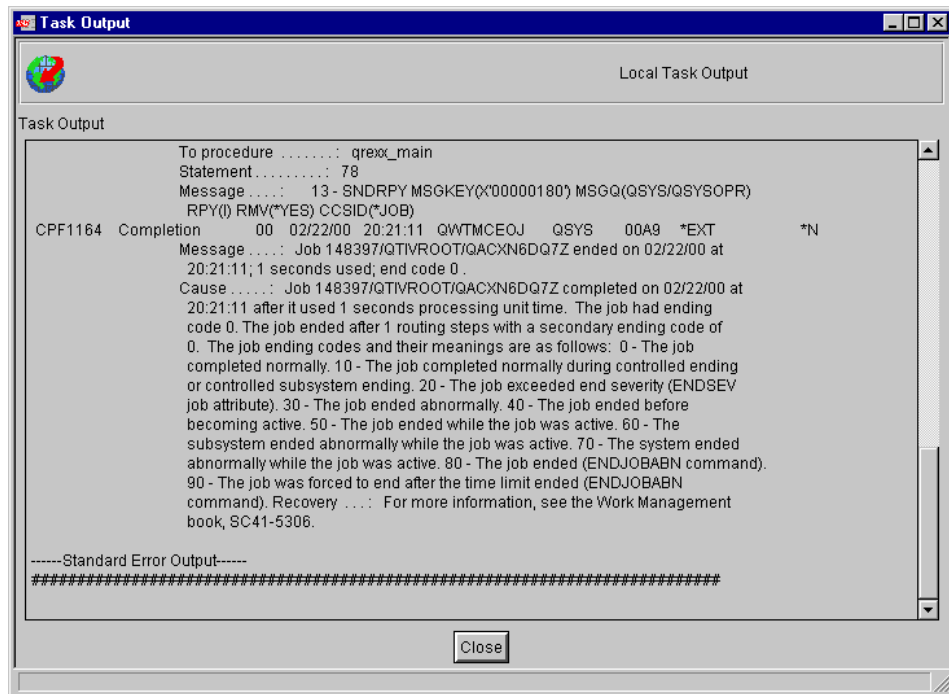


Figure 211. Method 2: Task output (2 of 2)

Method 3: Using variables with the Send_Reply task

The next method we illustrate is basically the same as method 2 with the following two exceptions.

- We customize our message adapter CDS file on the AS/400 so that the hostname will match the endpoint name. The benefit of this is that we will no longer have to select the endpoint on which we want the task to run. Obviously, changing the endpoint name to match the TCP/IP hostname value would provide the same benefit.
- Rather than initially taking note of the msg_key and sub_source attribute values, we will use variables to supply these values into the Send_Reply task for us.

We begin by first editing our message adapter CDS file using SEU as shown in Figure 212 on page 362.

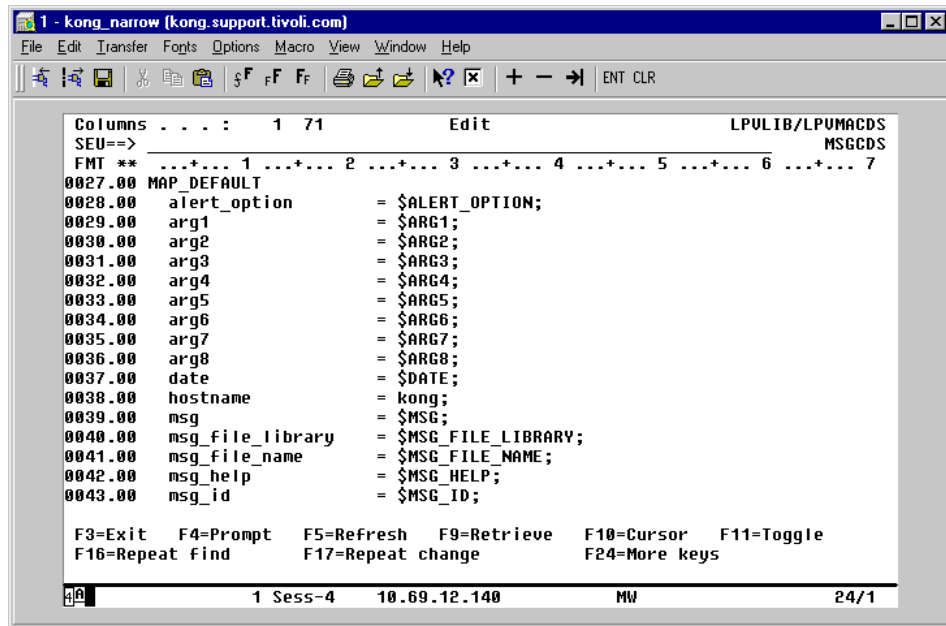


Figure 212. Method 3: Edit LPVLIB/LPVMACDS member MSGCDS

Notice that we have set the hostname attribute to be *kong*. Refer to Section 7.3.5.3, “Editing the CDS file” on page 347, for additional details of editing the CDS file.

We restart our AS/400 message adapter so the change will take affect.

Now, our hostname attribute will be set to kong as shown in the AS/400 Messages panel in Figure 213 on page 363.

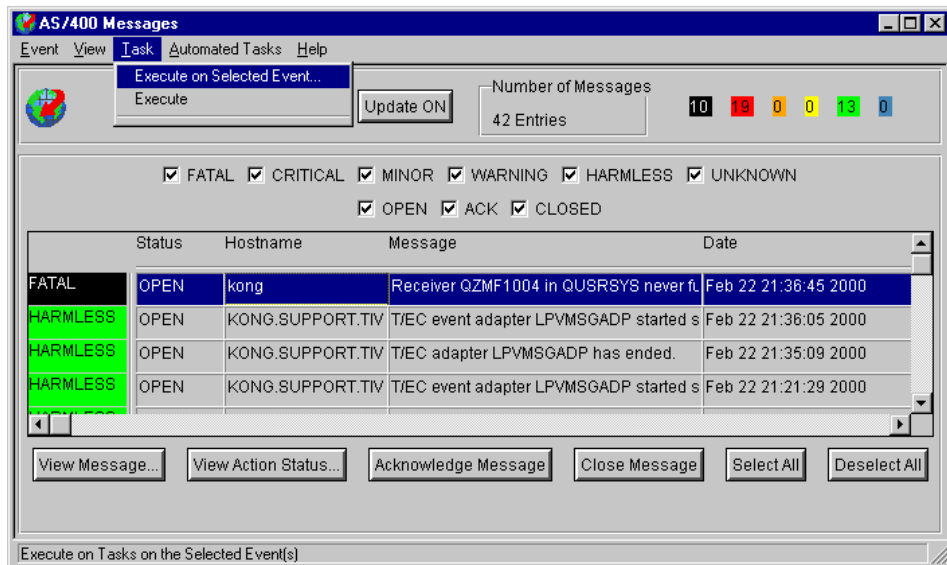


Figure 213. Method 3: Execute task on selected event

As in method 2, we highlight the inquiry message to which we want to reply and select **Execute on Selected Event** from the Task pull-down.

The TEC Tasks panel will result. Take note that, while we select the AS/400 Task library and the Send_Reply task as we did previously, we use variable names for the appropriate slot values of the Send_Reply task arguments as shown in Figure 214 on page 364.

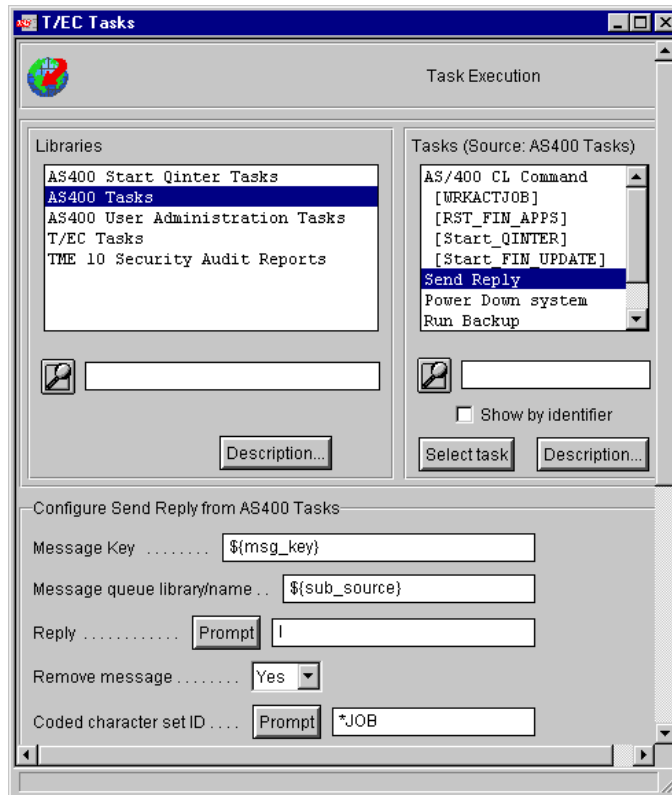


Figure 214. Method 3: TEC tasks (1 of 2)

Additionally, since we have our hostname attribute set to match our Endpoint name, we can now specify to **Execute on Selected Events** as shown in Figure 215.

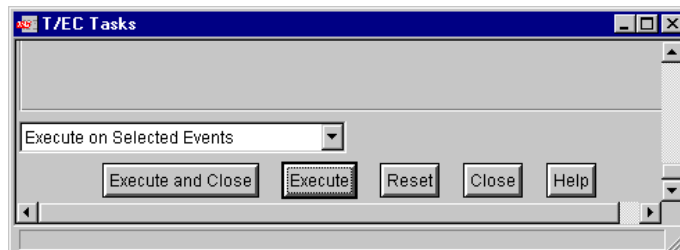


Figure 215. Method 3: TEC tasks (2 of 2)

Pressing the **Execute and Close** button will now run the task on the endpoint without us having to select the Endpoint as was required in Method 2.

The Task Output panel is shown in Figure 216 and confirms that the Send Reply command, `SNDRPY`, ran successfully.

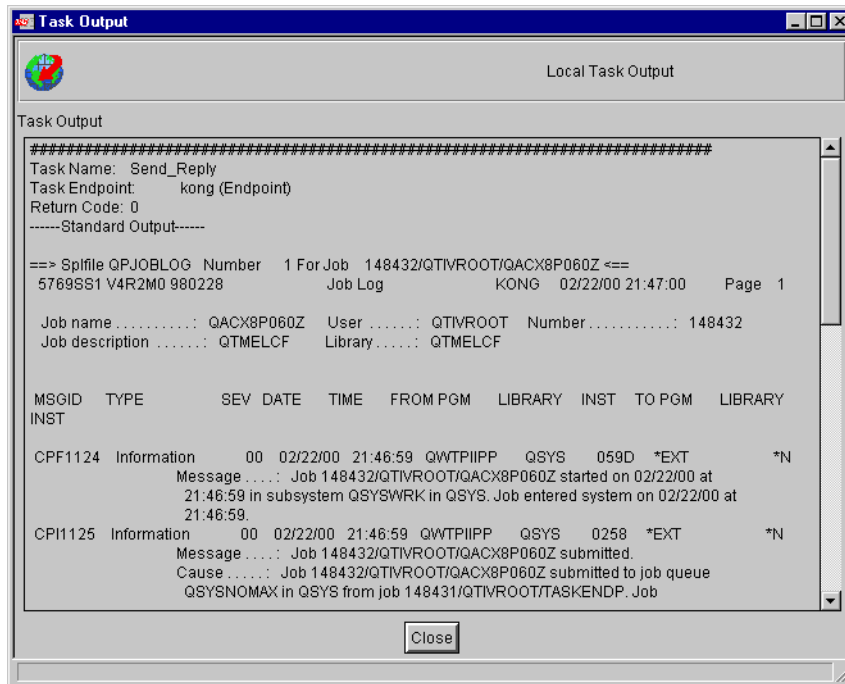


Figure 216. Method 3: Task output (1 of 2)

Figure 217 on page 366 shows the next page of the screen.

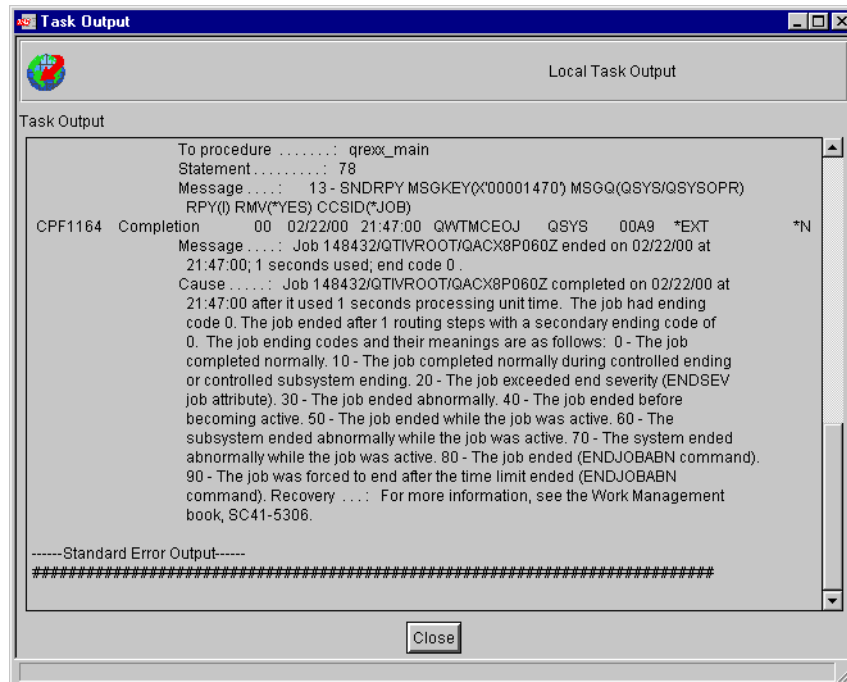


Figure 217. Method 3: Task output (2 of 2)

Method 4: Creating a task to set variables as default values

The method we illustrate here results from the realization that, since we can use event attribute variables as the Send_Reply task, there is really no need for us to type these in each time; so, we have altered the Send Reply task so that these values are filled by the default values. We also move the Reply argument so that it will be listed first. We do this because, in most cases, we expect that this may actually be the only value that may be altered from its default.

Here, we call our new task LPV_Send_Reply1 and define it as follows in the LPV_Send_Reply1.ttl file in the next screen.

```

Task LPV_Send_Reply1 {
  Description = ("", "LPV_Send_Reply1",1);
  HelpMessage = ("", "The LPV_Send_Reply1 task sends a reply to an AS/400
inquiry message. The reply argument is placed first as this is the mostly to be
specified. The message key and message queue arguments are defaulted to use th
e TEC event attribute values. The Remove and CCSID arguments are default based
on the Send Reply command defaults.",1);
  Roles = ("_!", "admin",1);
  Argument ("as400task","Reply . . . . . ",9) {
    DefaultValue = ("","DFT",1);
    Layout = ("","Reply",1);
  };
  Argument ("","Message Key . . . . . ",1) {
    DefaultValue = ("","${msg_key}",1);
    Layout = ("","Simple_Text",1);
  };
  Argument ("","Message queue library/name . . ",1) {
    DefaultValue = ("","${sub_source}",1);
    Layout = ("","Simple_Text",1);
  };
  Argument ("","Remove message . . . . . ",1) {
    DefaultValue = ("","*YES",1);
    Layout = ("","Yes_no",1);
  };
  Argument ("","Coded character set ID . . . . ",1) {
    DefaultValue = ("","*JOB",1);
    Layout = ("","CCSID",1);
  };
  Implementation ("os400-v3r7")
.Trace On
.Parse Arg reply msgkey msglibq rmvmmsg ccsid

.CALL ON ERROR NAME error_handler
.CALL ON FAILURE NAME error_handler

.msgkey = "x" || msgkey
.msgkey = msgkey || ""

      "SNDRPY MSGKEY("msgkey") MSGQ("msglibq,
      ") RPY("reply") RMV("rmvmmsg") CCSID("ccsid")"

.Exit 0
.Exit 1
};

```

Note

The task shown here is intended as a simple sample. It is supplied *as is* and is intended only as a sample. You may need to adjust it to meet the requirements of your environment.

We create the LPV_Send_Reply1 in the existing AS400 Task library by issuing the following command.

```
wtll -P /bin/cat -i -l "AS400 Tasks" LPV_Send_Reply1.tll
```

We use our task in exactly the same manner as we did previously. That is, we highlight the inquiry message to which we wish to respond and select “Execute on Selected Events” as shown in Figure 218 on page 368.

This time, however, we select the LPV_Send_Reply1 task in the AS400 Tasks library. Notice when the panel comes up the values are all filled in as shown except for Reply which we have set to “I”.

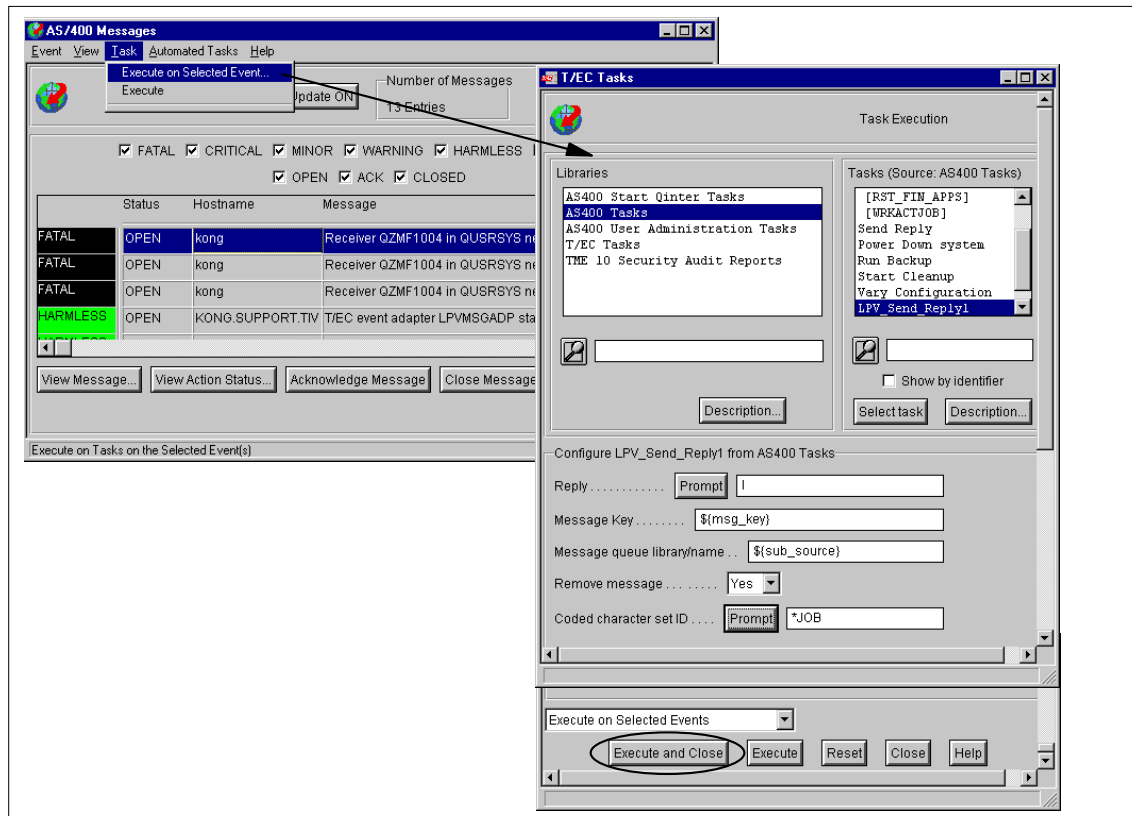


Figure 218. Method 4: Execute task on selected event

We then click on **Execute and Close** at the bottom of the TEC Tasks panel, leaving the Execute on Selected Events as set. The task runs, and we see the output and verify that the Send Reply command ran successfully.

Method 5: Creating a task to use only the reply argument

The final method we will illustrate builds on the previous one in that we wish to use the attributes from our inquiry messages as our Send Reply arguments. However, we take this a step further. Instead of setting the default values to be these attribute variables, we will completely remove the arguments from the TEC Tasks panel. Additionally, in our environment, we always leave the default values for the Remove message and CCSID parameters; so, we will remove these parameters as well. Since the only

parameter the administrator really needs to supply is the Reply value, this is the only argument that will be shown.

The major concern in actually removing these arguments is that our AS/400 implementation was actually being passed these argument values as shown in the following statement from the LPV_Send_Reply1.tll file.

```
.Parse Arg reply msgkey msglibq rmvmsg ccsid
```

Furthermore, at the time the arguments were passed, the variables, \$msg_key and \$sub_source, had already been resolved. Clearly, our REXX procedure running on the AS/400 will not be able to resolve these variables.

So, what this final task actually does is:

1. Obtain the Reply from the user.
2. Then, it simply runs the supplied Send_Reply task using the `wruntask` command supplying the Reply, the \$msg_key value, and the \$sub_source values as parameters. Recall that we are not specifying the Remove or CCSID values since they are defaulted as desired.

Here is the LPV_Send_Reply2.tll file that will perform this:

```
Task LPV_Send_Reply2 {
  Description = ("","LPV_Send_Reply2",1);
  HelpMessage = ("","The LPV_Send_Reply2 task sends a reply to an AS/400
inquiry message. Only the Reply value is an input argument. The message key a
nd message queue will be pulled from the selected TEC event. The remove and CCS
ID values are left as default by the Send Reply command.",1);
  Roles = ("!", "admin",1);
  Argument ("as400task","Reply . . . . . ",9) {
    DefaultValue = ("","*DFT",1);
    Layout = ("","Reply",1);
  };
  Implementation ("aix4-r1")
  .#!/bin/sh
  .rpy="$1"
  .wruntask -t Send_Reply -l "AS400 Tasks" -h $hostname -a "$msg_key $sub_source $rpy"
}
};
```

Note

The task shown here is intended as a simple sample. It is supplied *as is* and is only intended as a sample.

Notice that our implementation is actually a script running `wruntask` and `wsetmsg`. Therefore, the script itself does not run on the AS/400 Endpoint. Instead, as shown, we specify to run it locally.

We create the LPV_Send_Reply2 in the existing AS400 Task library by issuing the following command.

```
wtll -P /bin/cat -i -l "AS400 Tasks" LPV_Send_Reply2.tll
```

We use our task in exactly the same manner as previously shown. That is, we highlight the inquiry message to which we wish to respond and select “Execute on Selected Events”.

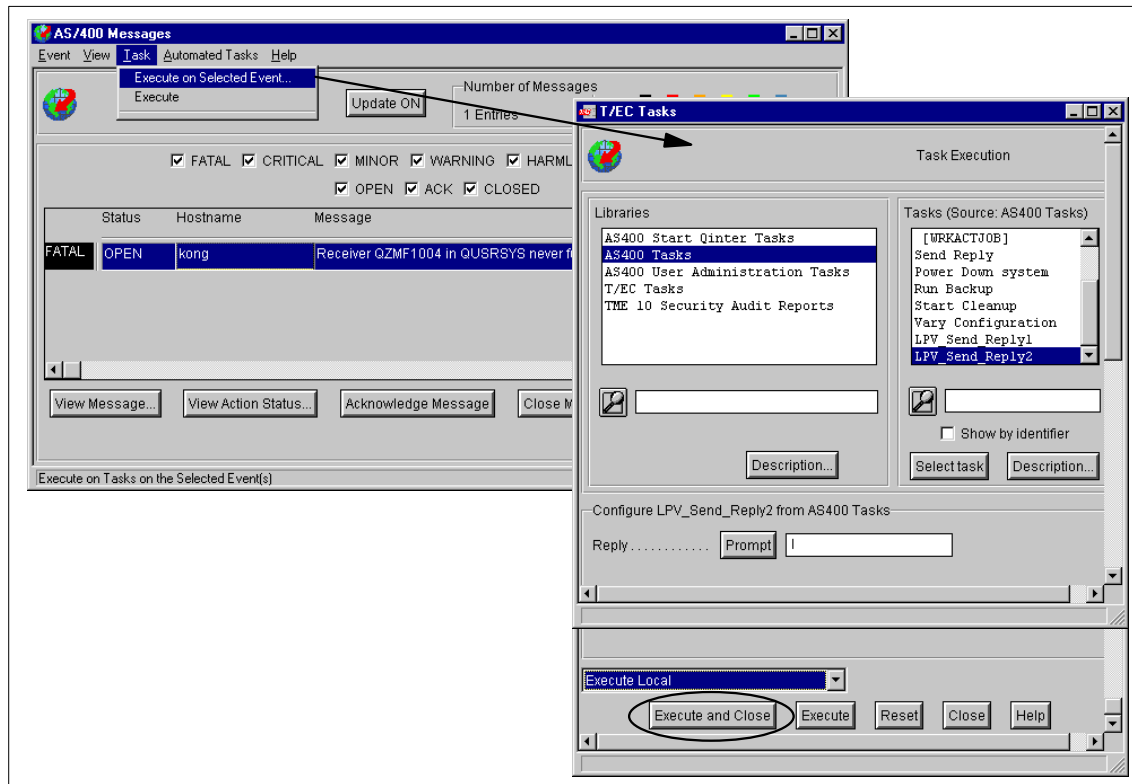


Figure 219. Method 5: Execute task on selected event

Select **Execute Local** on the TEC task panel.

Note

If you find yourself making an automated task or a rule, such as: *If message pppnnnn is received, send reply xx*, consider using the system reply list on the AS/400 to automate the reply to this message and then filtering out the message so that it doesn't get sent to TEC.

7.3.5.5 Using TEC to display User Admin UMBO_SKEL1 joblog

In this scenario, we will be adding a routing entry to the QSYSWRK subsystem description. We highly recommend that you have a thorough understanding of OS/400 Work Management before doing this on any AS/400.

Our goal in this scenario is to be able to run a task from the TEC console that will show us the User Administration UMBO_SKEL1 joblog. Refer to Section 6.2, “Tivoli User Administration for AS/400” on page 141, for a discussion of this job.

To accomplish this, we first need to send a message to a message queue on the AS/400 that will include the qualified job name. The TEC AS/400 message adapter monitoring this message queue will then forward the message to our TEC event server. Finally, from the TEC console, we will run a task that will use the qualified job name to ftp the joblog to the managed node and display it for us.

Here are the precise steps:

1. Use the Create Message File command, `CRTMSGF`, to create a message queue called `TIVMSGF` in the `LPVLIB` library as follows:

```
CRTMSGF MSGF (LPVLIB/TIVMSGF)
```

2. To access our message file, use the Work with Message Descriptions command, `WRKMSGD`, as follows:

```
WRKMSGD MSGF (LPVLIB/TIVMSGF)
```

This will bring up the Work with Message Descriptions panel as shown in Figure 220 on page 372.

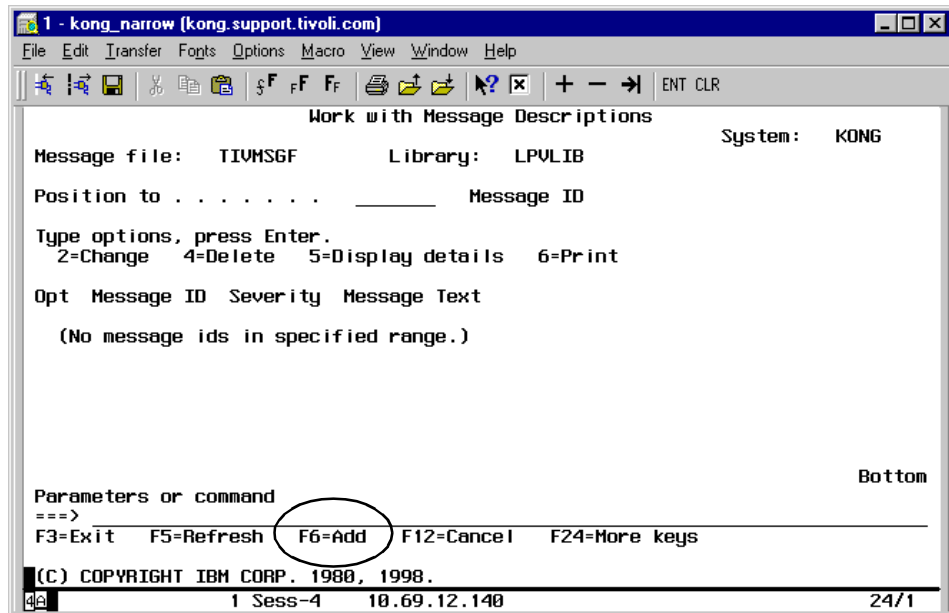


Figure 220. Work with Message Descriptions

3. Press **F6** (Add) to add the TIV0001 message description that we will use to send the qualified job name.
4. Complete the first screen, Add Message Description (ADDMSGD) panel, as we shown in Figure 221 on page 373.

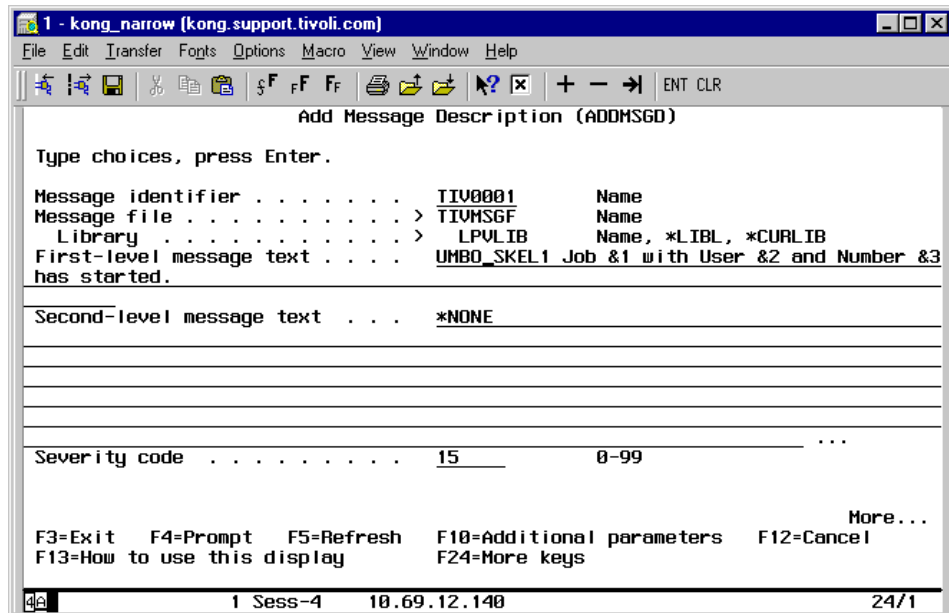


Figure 221. Add Message Description (ADDMSGD)

Notice the use of field data values &1, &2, and &3. When we send the message, we will use these fields to contain the actual UMBO_SKE1 job name, user, and number.

5. Page down, and complete the field data formats as shown in Figure 222 on page 374.

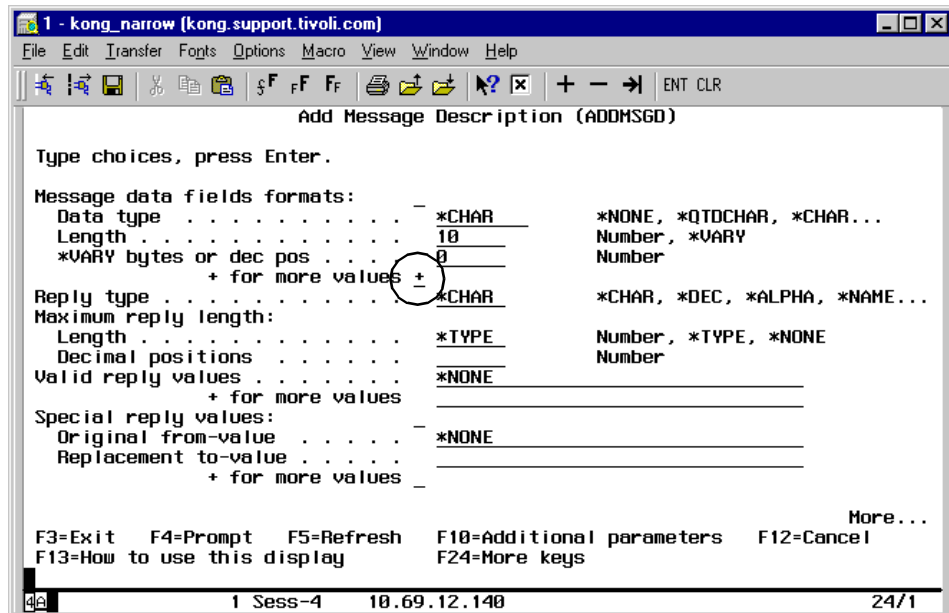


Figure 222. Filling out the Add Message Description panel

Since we have more than one data field, we enter a plus sign (+) for more values. When we press Enter, we can enter the remaining two defer fields as shown in Figure 223 on page 375.

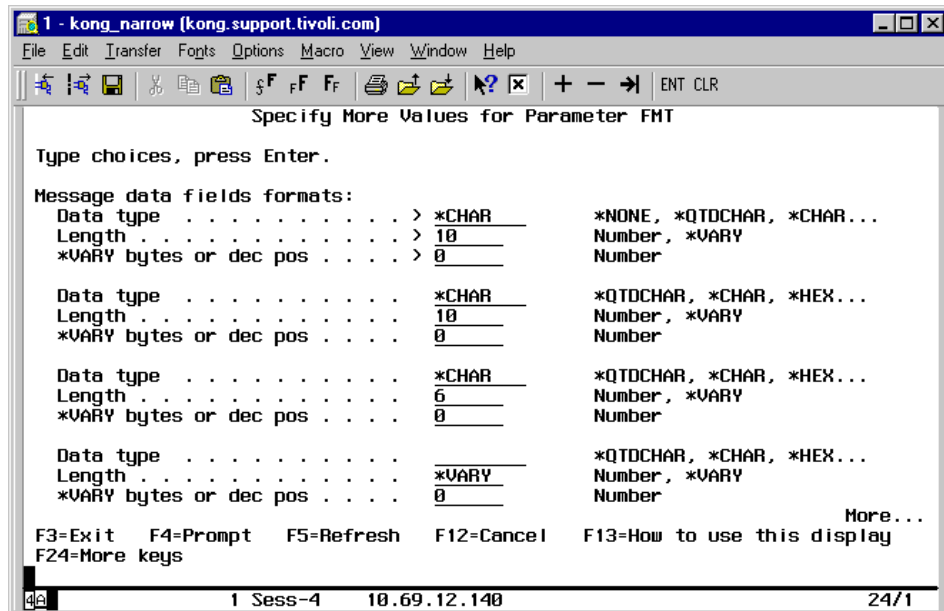


Figure 223. Specifying more values for parameter format

Notice that the job name and job user are 10 characters on the AS/400 while the job number is only six. Press **Enter** to complete the entry, and then press **Enter** again to complete adding the TIV0001 message description.

The Work with Message Descriptions panel, shown in Figure 224 on page 376, now shows our TIV0001 message.

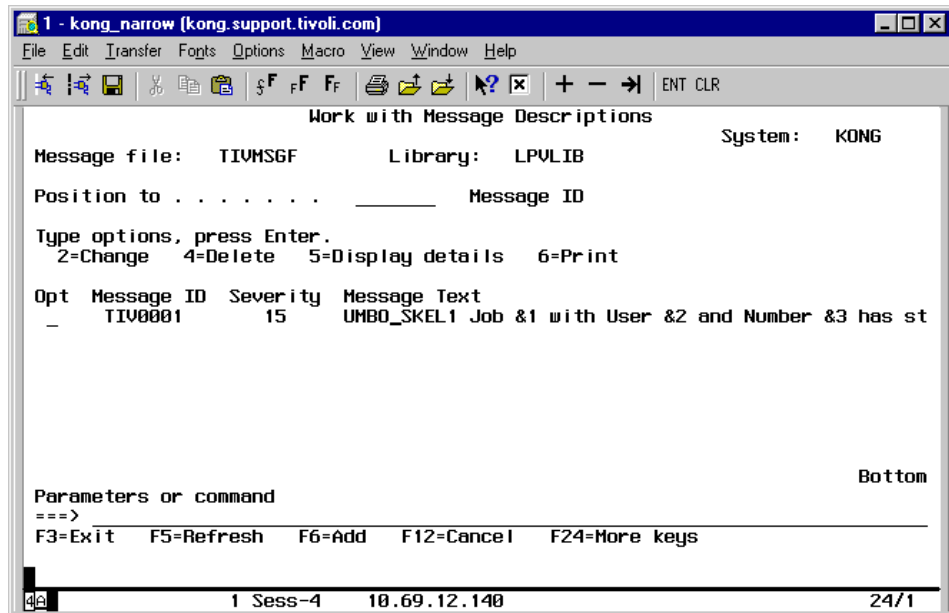


Figure 224. The TIV0001 message description

6. Now that we have our message description to use, we use the Create Message Queue () command, `CRMSGQ`, to create a message queue to which to send our TIV0001 messages:

```
CRMSGQ MSGQ (LPVLIB/TIVADM)
```

7. Next, we write and compile a program that retrieves the job's attributes (job name, user, and number) storing them as CL variables. It then checks to see if the job name is UMBO_SKEL1. If so, we send our TIV0001 message to our TIVADM message queue. The qualified job name (&JOBNAME) is included as the field data value. QCMD is then called to process the job as shown in Figure 225 on page 377.

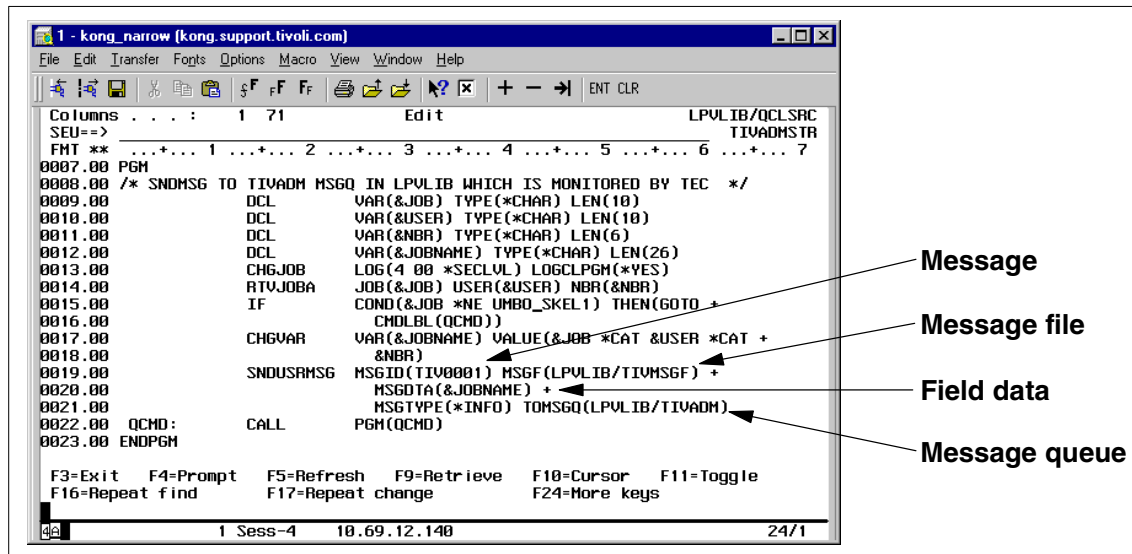


Figure 225. Editing the program that retrieves the job attributes

Note

The task shown here is intended as a simple sample. It is supplied *as is* and is intended only as a sample.

8. Since the UMBO_SKEL1 job currently uses QCMDB as its routing data, we add a routing entry to the QSYSWRK subsystem that will match this value and, as a result, issue the TIVADMSTR program. This is done using the Add Routing Entry command, ADDRTGE:

```
ADDRTGE SBSD(QSYSWRK) SEQNBR(7777) CMPVAL(QCMDB)
PGM(LPVLIB/TIVADMSTR) CLS(QSYSCLS50)
```

9. Finally, on the AS/400, we configure and start an AS/400 message adapter to monitor this message queue.

- a. First, we make a copy of the supplied configuration file:

```
CPYF(LPVLIB/TIVADMCFG) FROMMBR(MSGCFG) TOFILE(LPVLIB/TIVADMCFG)
FROMMBR(MSGCFG) TOMBR(MSGCFG) MBROPT(*REPLACE) CRTFILE(*YES)
```

- b. Edit this new TIVADMCFG file to specify the LPVLIB/TIVADM message queue shown in Figure 226 on page 378.

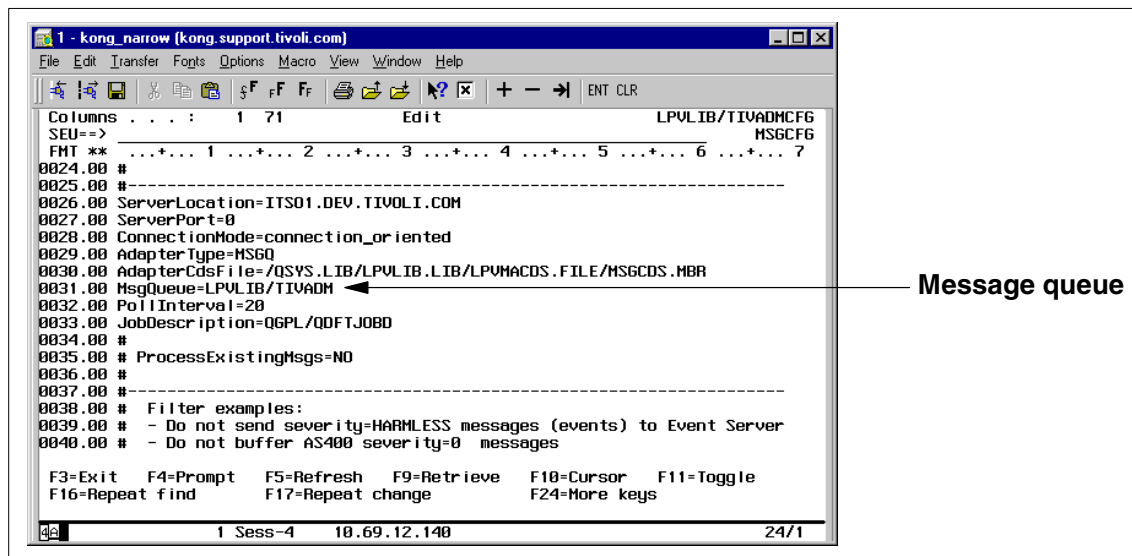


Figure 226. Editing the TIVADMCFG file

c. Start this AS/400 message adapter as follows.

```
STRTECADP EVTADP (TIVADMADP)
CFGFILE ('/qsys.lib/lpvlib.lib/tivadmcfg.file/msgcfg.mbr')
```

10. At this point, we distribute a User Administration profile and issue the Display Message command, `DSPMSG`, to verify that our program is running and that our TIV0001 message is sent to the TIVADM message queue:

```
DSPMSG MSGQ (LPVLIB/TIVADM)
```

See Figure 227 on page 379.

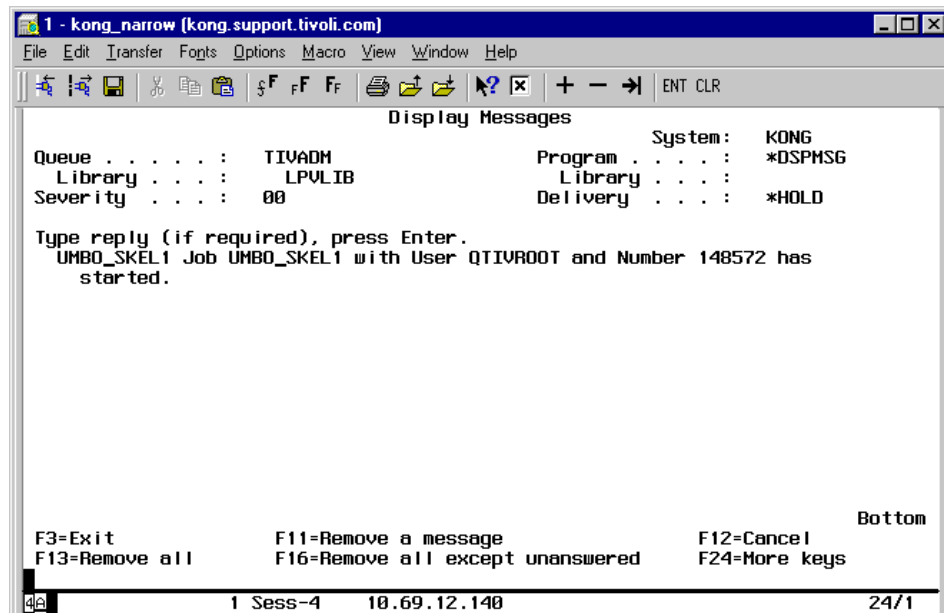


Figure 227. Displaying the TIVADM message queue

Take note of the User Administration fully-qualified name as follows:

Job UMBO_SKEL1
User QTIVROOT
Number 148572

11.From the TEC console, we receive the following event.

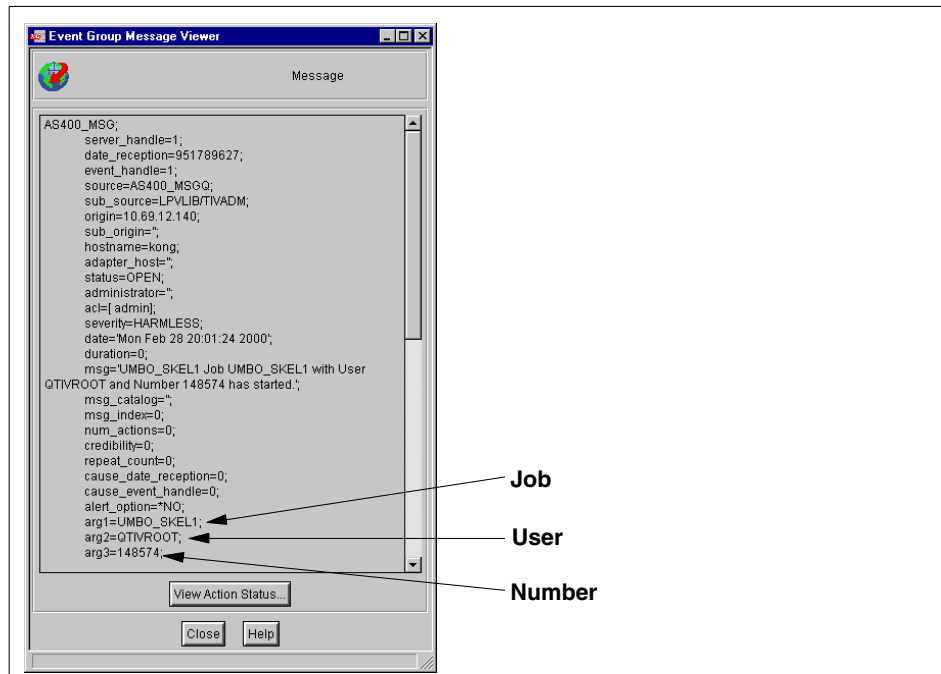


Figure 228. Browsing TEC event

Take note again of the UMBO_SKEL1 fully-qualified job name in the argument status.

12.The last step is to produce the task to perform the desired ftp function.

a. We show our task source here:

```
# cat LPV_Get_UA_Joblog.tll
Task LPV_Get_UA_Joblog {
    Description = ("", "LPV_Get_UA_Joblog",1);
    HelpMessage = ("", "The LPV_Get_UA_Joblog2 task sends retrieves an User
Admin joblog from the AS/400 and places it in the specified File using FTP.",1);
    Roles = ("!", "admin",1);
    Argument ("", "File to receive joblog",1) {
        DefaultValue = ("", "/tmp/UA_joblog",1);
        Layout = ("", "Simple_Text",1);
    };
    Implementation ("aix4-r1")
        .#!/bin/sh
        .export HOME=/
        .putfile="$1"
        .(echo quote R CMD "CRTPF FILE(QGPL/COPY) RCDLEN(140)";echo quote R CMD
"CPYSPLF FILE(QRJOBLOG) TOPFILE(QGPL/COPY) JOB($arg3/$arg2/$arg1) SPINBR(*LAST)";
echo get QGPL/COPY $putfile;echo quote R CMD "DLTF FILE(QGPL/COPY)";echo quit)|ft
p $hostname
        .cat $putfile
    };
};
```

Note

The task shown here is intended as a simple sample. It is supplied *as is* and is only intended as a sample.

Notice that our implementation is actually a script running `wruntask` and `wsetmsg`. Therefore, the script itself does not run on the AS/400 endpoint. Instead, as shown, we specify that it be run locally.

- b. Create the `LPV_Get_UA_Joblog` task in the existing AS/400 tasks library by issuing the following command:

```
wtl1 -P /bin/cat -i -l "AS/400 Tasks" LPV_Get_UA_Joblog.tll
```

- c. Since we are using the `.netrc` file, we must ensure the System Privileges are set as shown in Figure 229.

The screenshot shows the 'Edit Task' window for the task 'LPV_Get_UA_Joblog'. The window is divided into several sections:

- Task Properties:**
 - Platforms Supported:** A list of platforms with checkboxes. 'Generic' is selected. Other options include SPARC / SunOS, SPARC / Solaris, PA-RISC / HPUX 9, PA-RISC / HPUX 10, IBM RS/6000 / AIX 3, and IBM RS/6000 / AIX 4 (which is also checked).
 - Roles Required to Execute Task:** A list box containing 'install_client', 'security_admin', 'security_auditor', and 'security_operator'.
 - Execution Privileges:** Two text fields: 'User Name' set to 'root' and 'Group Name' set to 'system'.
- Task History and Comments:** A text area showing task details:
 - Task Name : AS400 Tasks/LPV_Get_UA_Joblog
 - Task Created : Tue Feb 29 00:32:14 2000
 - Task Created By : root@itso1.dev.tivoli.com
 - Task Files : aix4-r1 itso1 /tmp/taskbwAtRlb
 - Distribution Mode : ALI
 - Task Comments :
- New Comments:** An empty text area for adding new comments.

Figure 229. Setting task execution privileges

We use our task in exactly the same manner as previously shown. That is, we highlight the inquiry message to which we wish to respond and select **Execute on Selected Events**.

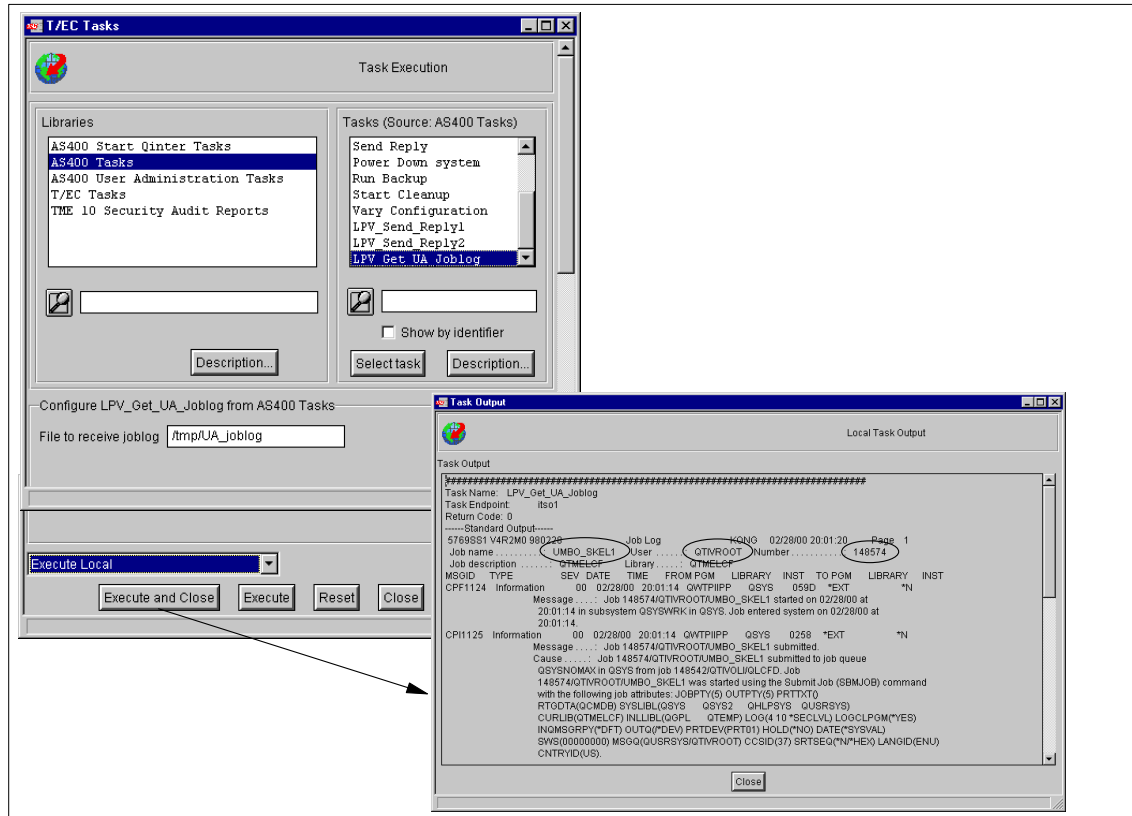


Figure 230. Executing the LPV_Get_UA_Joblog task

7.3.5.6 Creating an AS/400 Tivoli Inventory menu

The TEC AS/400 Adapters send information from the AS/400 to the TEC event console, thus, providing a means by which AS/400 users can communicate with Tivoli administrators. Furthermore, through the use of rules or automated tasks, AS/400 users could actually perform functions that would normally be initiated by a Tivoli administrator. In this scenario, we will illustrate a very simple example in which an AS/400 user can select an option from a Tivoli Inventory menu called TIVINV to request that an Inventory scan be performed. An option like this might be taken, for example, by an AS/400 system administrator after applying the latest cumulative PTF package.

Similar to the previous scenario, we begin by creating a new message file called TIVINVMSGF. We then add the three messages shown in Figure 231.

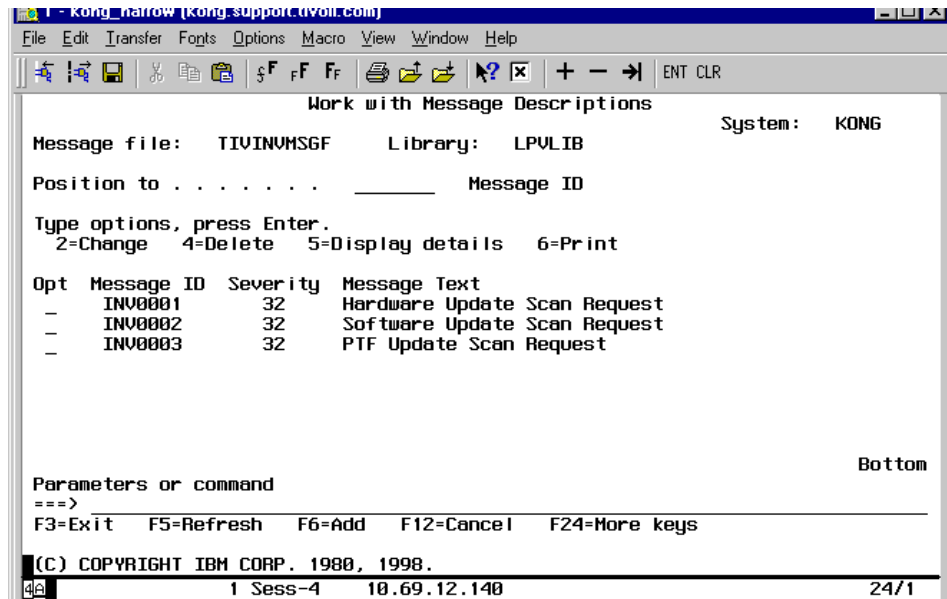


Figure 231. WRKMSGD MSGID(*FIRST) MSGF(LPVLIB/TIVINVMSGF)

These are simple text messages containing no field data. See the previous example in Section 7.3.5.5, "Using TEC to display User Admin UMBO_SKEL1 joblog" on page 371, for complete steps for creating a message file and adding message descriptions.

Also, as in the previous example, we create a message queue to be used strictly for monitoring our inventory request messages. We call it TIVINV and place it in the LPVLIB library.

Now, we write three simple CL programs, each issuing a Send User Message command, SNDUSRMSG, to send one of these three messages to our TIVINV message queue. Our programs are:

- SNDINV0001** Sends INV0001 to the TIVINV message queue
- SNDINV0002** Sends INV0002 to the TIVINV message queue
- SNDINV0003** Sends INV0003 to the TIVINV message queue

The source for the SNDINV0003 program is shown in Figure 232 on page 384.

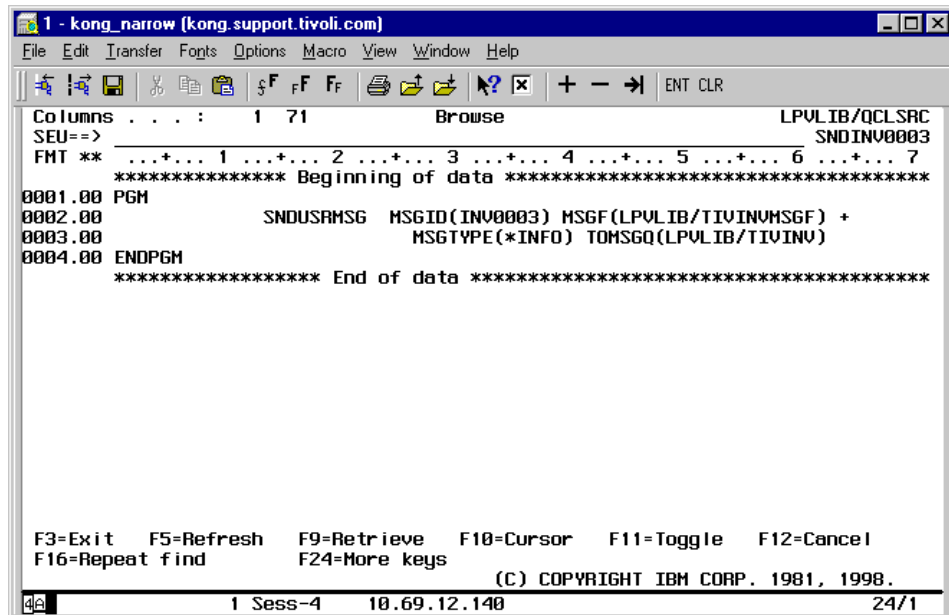


Figure 232. Source of SNDINV0003 program

Now, we have programs, each of which sends the appropriate message to the TIVINV message queue. We are ready to build our Tivoli Inventory menu. Building a menu on the AS/400 is quite easy using the Screen Design Aid (SDA), which is part of the Application Development Toolset. While we will illustrate the creation of this menu, we will not discuss the details of SDA beyond the illustrations. Refer to the *IBM AS/400 Screen Design Aid User's Guide and Reference*, SC09-1171, for details.

To begin, we must have a source physical file in which to store our menu source. We create this using the following Create Source Physical File command:

```
CRTSRCPF FILE(LPVLIB/MENUSRC)
```

We then issue the Start SDA command, `STRSDA`, resulting in the screen shown in Figure 233 on page 385.

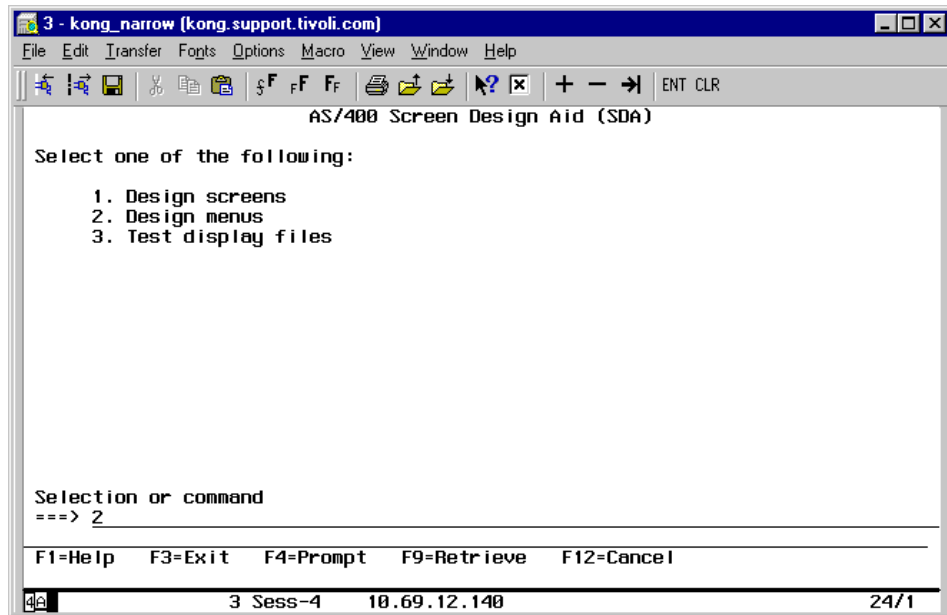


Figure 233. The AS/400 Screen Design Aid (SDA)

In this panel, select option **2 Design menus**. From the Design Menus panel, shown in Figure 234 on page 386, supply the source file that we created earlier and the name of the menu we wish to create.

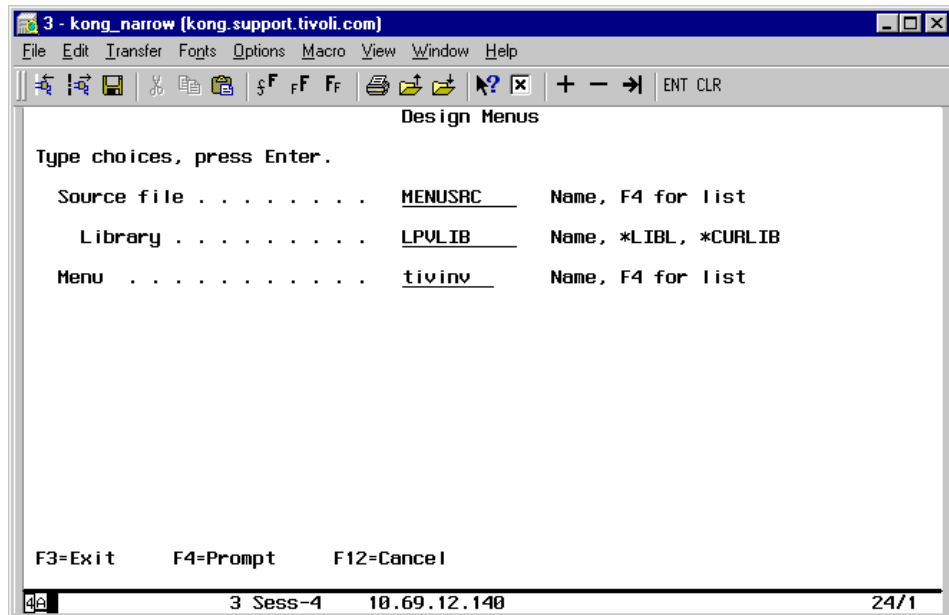


Figure 234. The design menus panel

Press **Enter**. We call our menu TIVINV. Select **Y** for the Work with menu image and commands prompt as shown in Figure 235 on page 387.

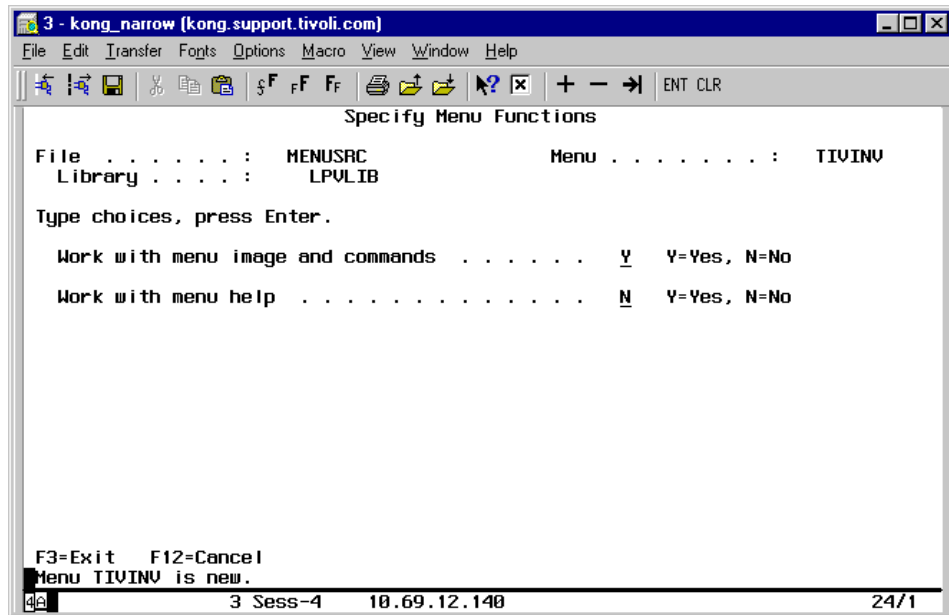


Figure 235. The specific menu functions panel

In this panel, press **Enter**. On the TIVINV menu, shown in Figure 236 on page 388, type in the menu option text as you would like it to appear.

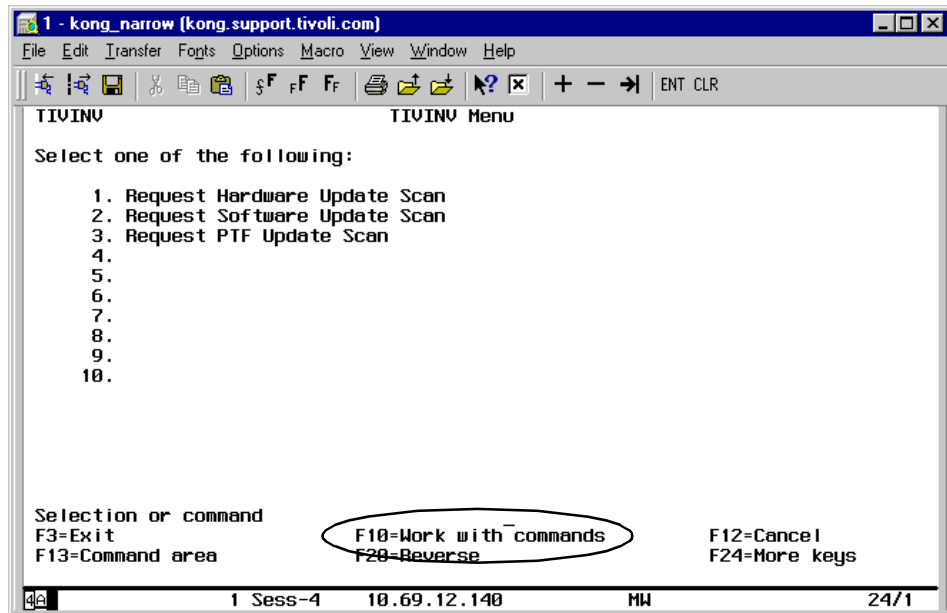


Figure 236. The TIVINV menu

Press **F10** (Work with commands). On the Define menu commands panel, shown in Figure 237 on page 389, supply the program CALL corresponding to the menu option.

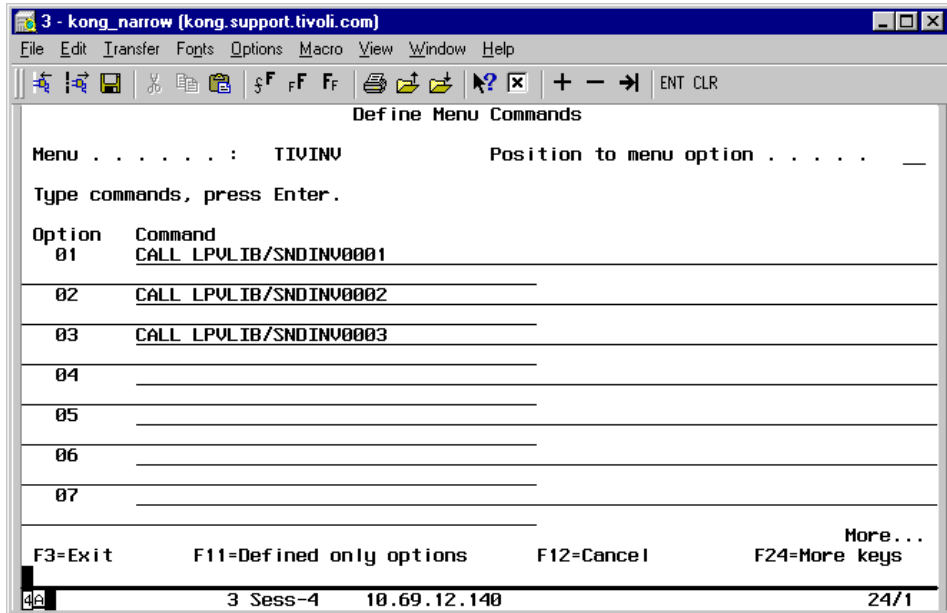


Figure 237. The define menu commands panel

For example, when selected, TIVINV menu option 3, Request PTF Update Scan, will issue:

CALL LPVLIB/SNDINV0003

Recall that this program will send the INV0003 message, PTF Update Scan Request, to the TIVINV message queue.

Press **Enter** to return to the TIVINV menu. Then, press **Enter** and **F3** (Exit). Now, on the Specify Menu Functions screen, shown in Figure 238 on page 390, specify **N** for both parameters.

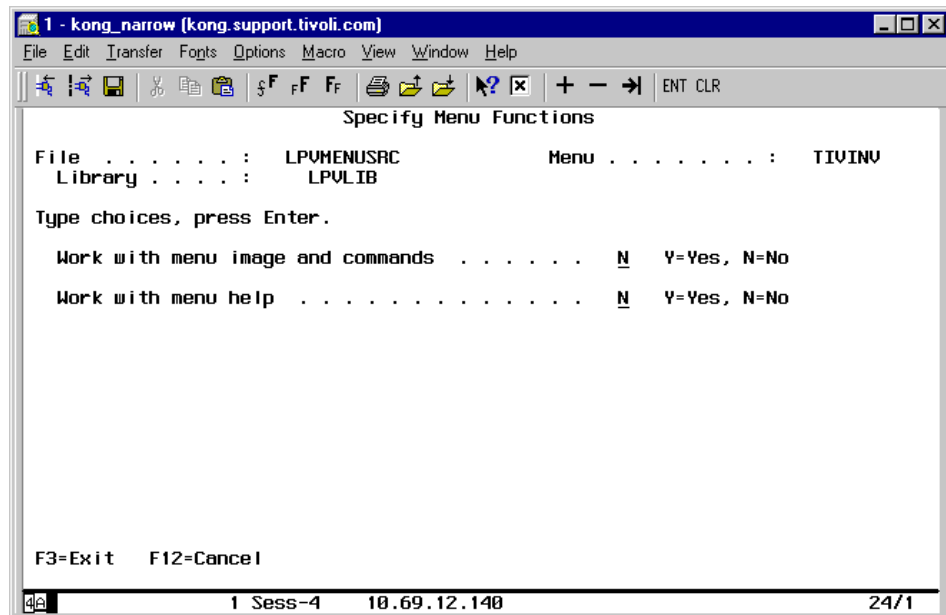


Figure 238. The specify menu functions panel

Press **Enter**. On the Exit SDA Menus panel, shown in Figure 239 on page 391, we choose to save the menu source and create the TIVINV menu.

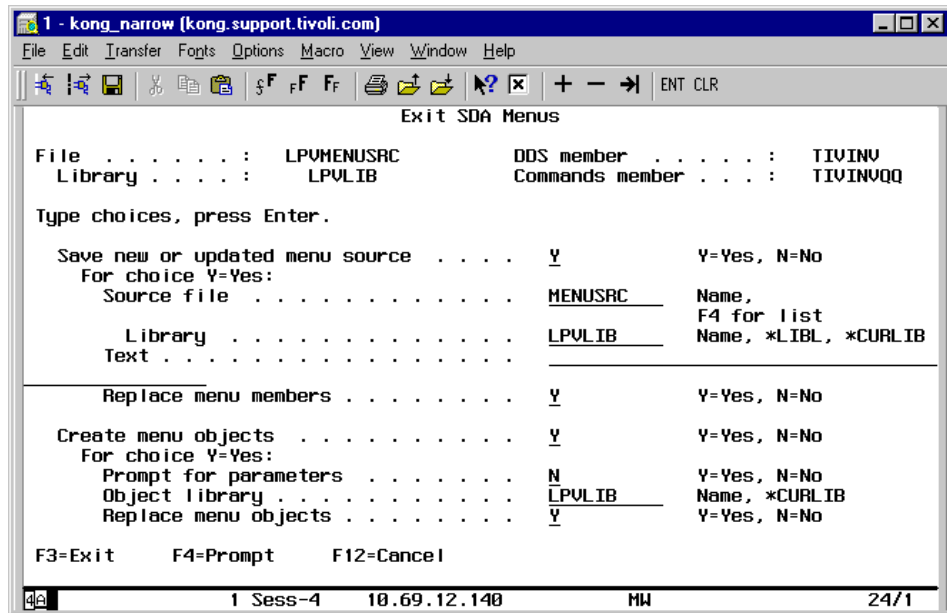


Figure 239. The exit SDA menus panel

Press **Enter**. Now that the menu has been compiled, we can access the menu using the Go to Menu command, **GO**:

GO LPVLIB/TIVINV

The menu will be displayed as shown in Figure 240 on page 392.

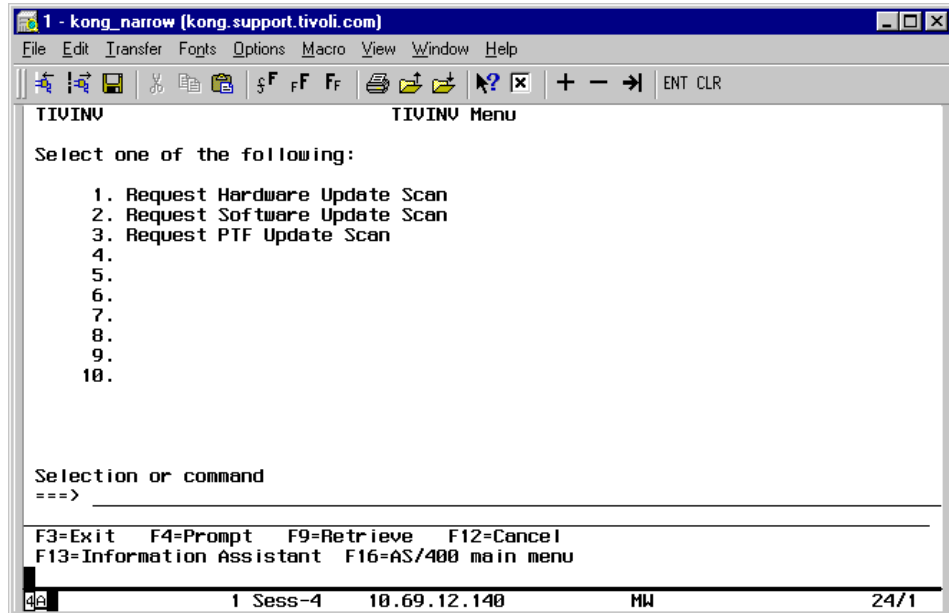


Figure 240. The TIVINV menu after compile

For testing, we select options **1**, **2**, and **3** from the menu, and then we issue the `DSPMSG LPVLIB/TIVINV` command to display the TIVINV message queue as shown in Figure 241 on page 393.

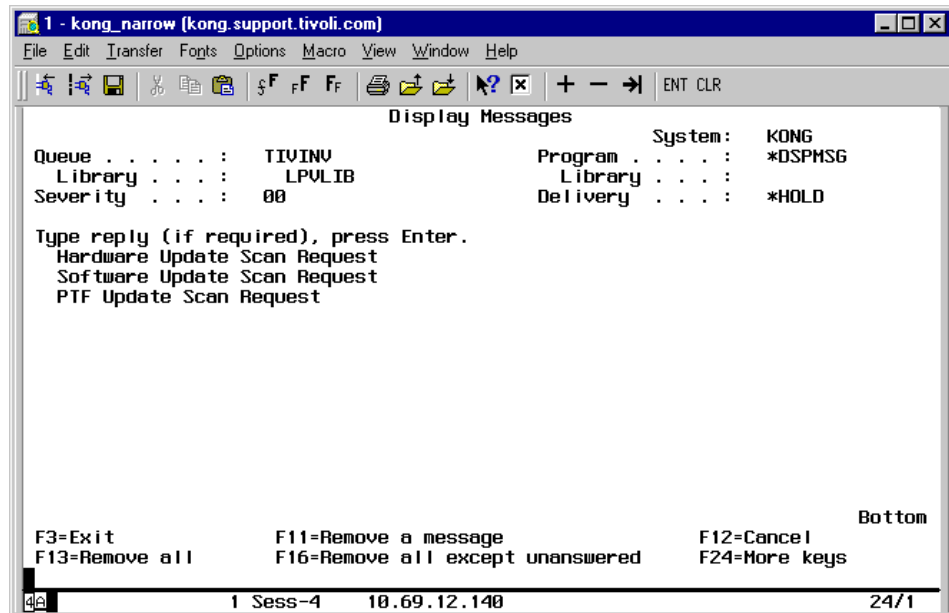


Figure 241. Displaying the TIVINV message queue

Now, we configure a TEC message adapter to monitor this message queue:

1. First, we make a copy of the supplied message adapter configuration file.

```
CPYF FROMFILE(QUSRSYS/QAYMACFG) TOFILE(LPVLIB/TIVINVCFG)
FROMMBR(MSGCFG) TOMBR(MSGCFG) MBROPT(*REPLACE) CRTFILE(*YES)
```

2. Edit this new TIVINVCFG file to specify the LPVLIB/TIVINV message queue as shown in Figure 242 on page 394.

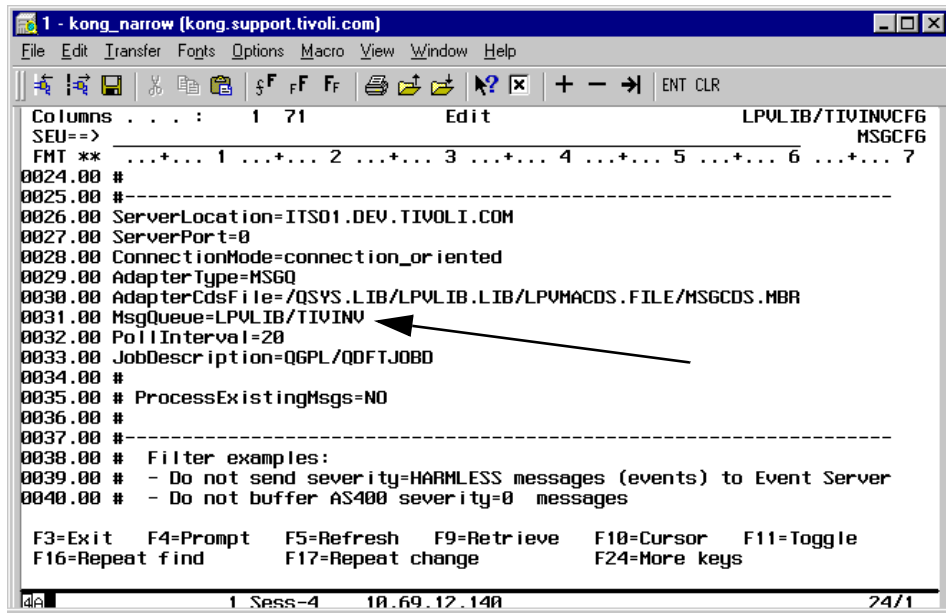


Figure 242. Editing the TIVINVCFG configuration file

3. Start this AS/400 message adapter.

```
STRTECADP EVTADP(TIVINVADP)
CFGFILE('/qsys.lib/lpvlib.lib/tivinvcfg.file/msgcfg.mbr')
```

Next, create three tasks to perform the `wdistrib` command of the Inventory profile corresponding to each of the requested scans. For example, the following screen shows the `LPV_INV_PTF_Scan.tll` defining task to be performed as a result of the `INV0003` "Request PTF Scan" message.

```
# cat LPV_INV_PTF_Scan.tll
Task LPV_INV_PTF_Scan {
  Description = (" ", "LPV_INV_PTF_Scan",1);
  HelpMessage = (" ", "The LPV_INV_PTF_Scan task distributes the PTF_AS400
Inventory Profile to the Endpoint.",1);
  Roles = ("! ", "admin",1);
  Implementation ("aix4-r1")
    .#!/bin/sh
    .wdistrib @InventoryProfile:PTF_AS400 @Endpoint:$hostname
};
```

Note

The task shown here is intended as a simple sample. It is supplied *as is* and is intended only as a sample.

Notice that, since our implementation is actually a script running the `wruntask` and `wsetmsg` commands. Therefore, this script itself does not run on the AS/400 Endpoint. Instead, as shown, we specify that it be run locally.

We create the LPV_INV_PTF_Scan task in the existing AS400 Task library by issuing the following command:

```
wtll -P /bin/cat -i -l "AS400 Tasks" LPV_INV_PTF_Scan.tll
```

Since no information is required from a Tivoli administrator, this task is best implemented as an automated task or a rule. For our example we will use an automated task for this. Refer to the *TME 10 Enterprise Console Rule Builder's Guide*, SC31-8508, for details of rule writing.

Create the Automated Task as shown in the following panels in Figure 243 on page 396.

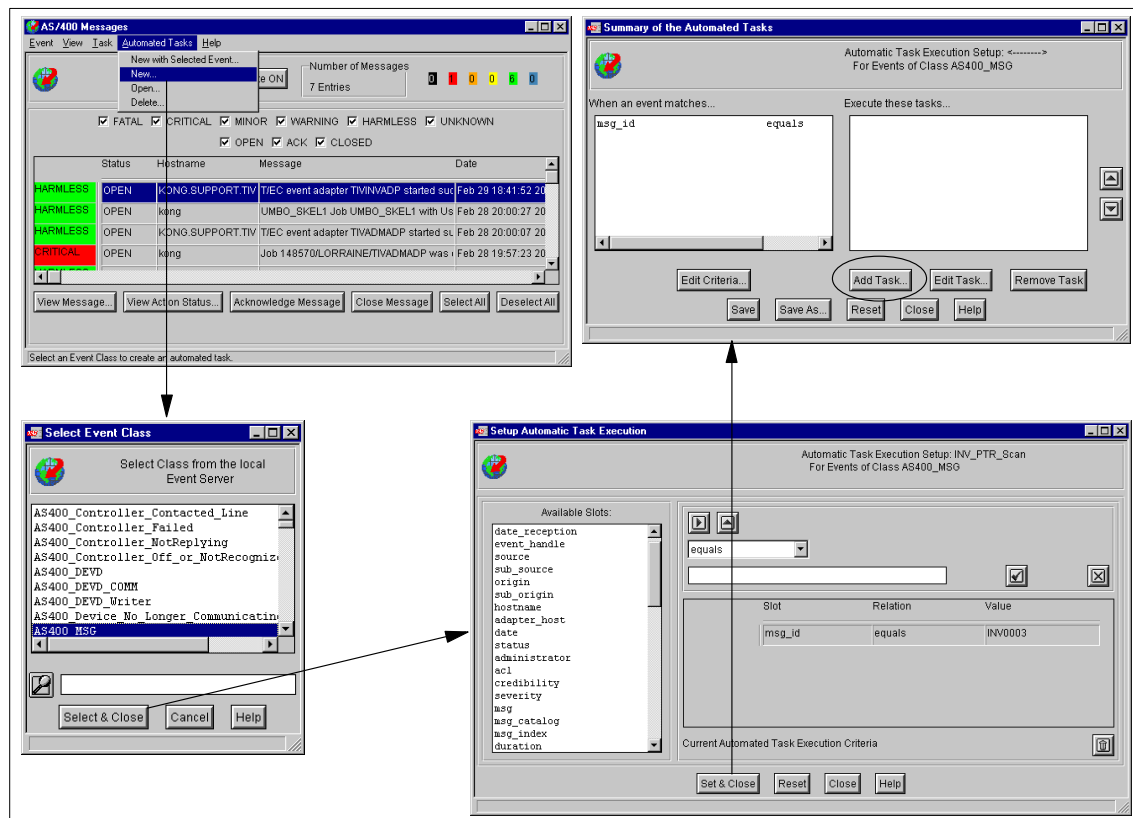


Figure 243. Creating automated task

Select **Automated Tasks** -> **New** on the AS/400 Messages panel, and then select **AS/400 MSG** as the event class on the Select Event Class panel. Press **Select & Close**.

Select the following values on the Setup Automatic Task Execution panel:

Slot msg_id
Relation equals
Value INV0003

Then, press **Set & Close** and **Add Task** on the Summary of the Automated Tasks panel shown in Figure 244 on page 397.

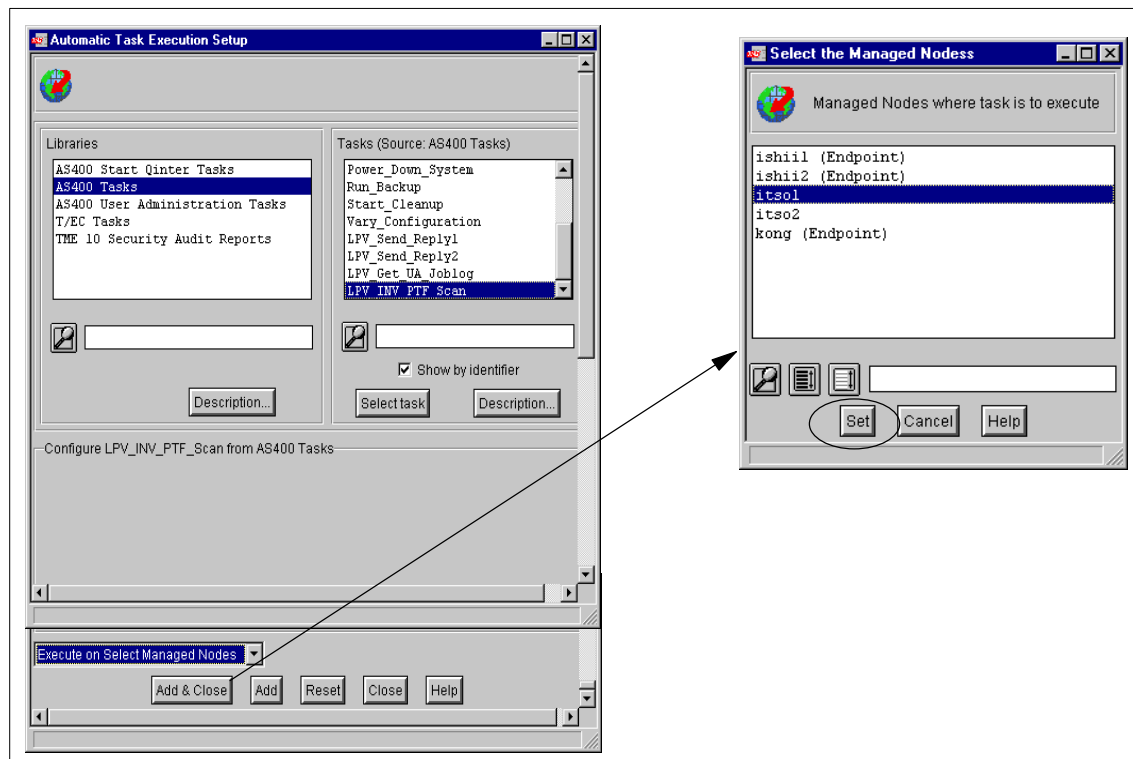


Figure 244. Setting up the automatic task execution

Then, on the Automatic Task Execution Setup panel, select the **LPV_INV_PTF_Scan** task. Specify **Execute on Select Managed Nodes** and press **Add & Close**.

Select the Managed Node on which you want the task, that is, the `wddistrib` command, to run. Then, press the **Set** button. It returns you to the Summary of the Automated Tasks panel shown in Figure 245 on page 398. Then, press the **Save** button.

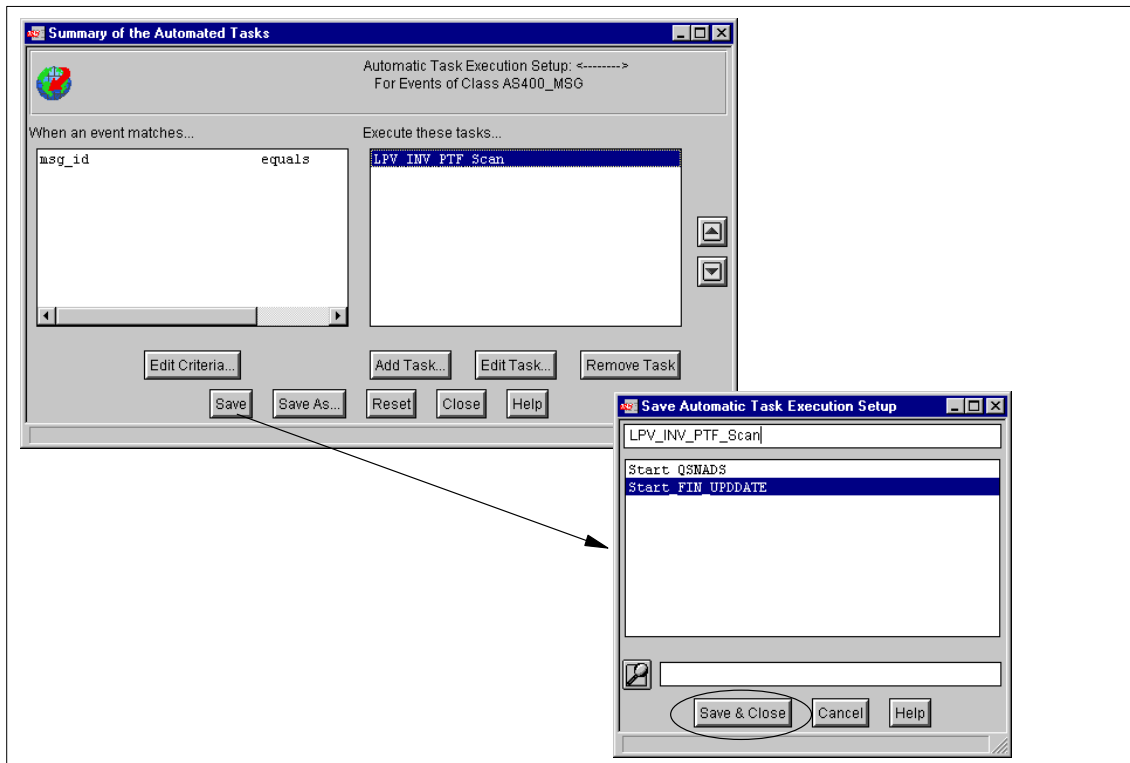


Figure 245. Saving Automatic task execution setup

On the Save Automatic Task Execution Setup panel, press **Save & Close**.

Now, let us see the task in action. Suppose we are an AS/400 system administrator who has just applied a PTF cumulative tape and now wants to run a PTF update scan. We enter the following command at the AS/400 command line:

```
GO MENU (LPVLIB/TIVINV)
```

On the TIVINV menu, shown in Figure 246 on page 399, we select option **3**.

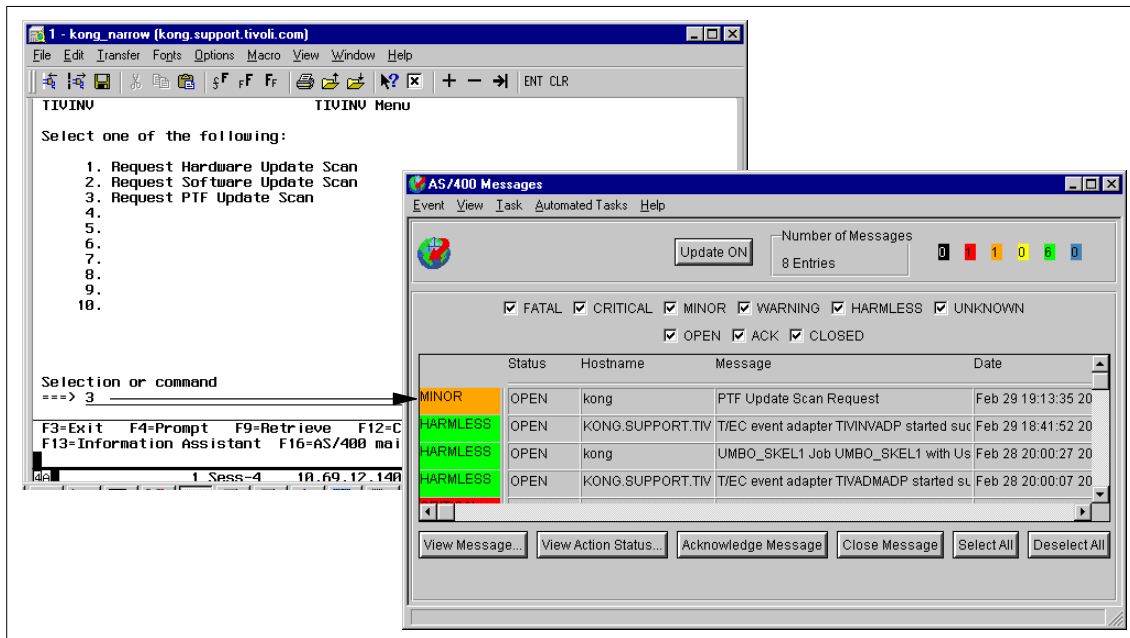


Figure 246. Requesting PTF update scan

This sends the INV0003 message to the TIVINV message queue when the TIVINV adapter extracts it and forwards the event to TEC. On the TEC console, we verify that the event is received as shown in Figure 246.

Since the PTF scan is an automated task, we do not need to do anything to start. We run the Work with Active Jobs command, `WRKACTJOB`, to see if we can identify the INVENTORY job running as shown in Figure 247 on page 400.

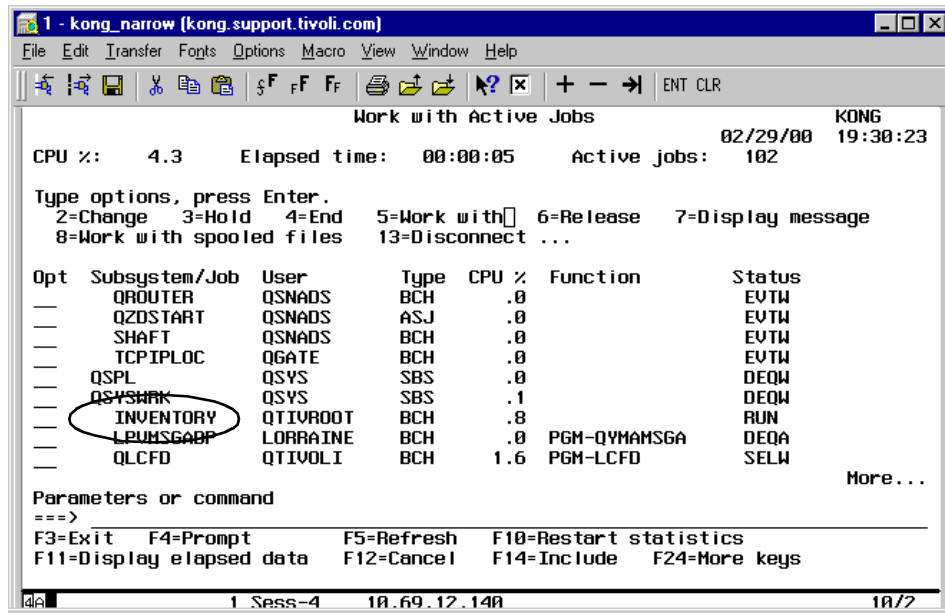


Figure 247. Inventory scanning request job

Chapter 8. AS/400 deployment management

Deployment addresses every phase in an application's life cycle from initial deployment planning through the final configuration and fine-tuning of the application on each system. In a deployment, you need to manage configurations, assets, and distribution.

Deploying a standard desktop application, such as a spreadsheet, is not difficult, but, when you try to distribute a multi-component application that has a database, a server component, and a desktop component, and they span different platforms types, then, the job is more complex. You have to ensure that all the pieces are in place before you enable the new application, and you have to have a way to roll back a deployment if something fails.

8.1 Overview of Tivoli deployment management applications

With Version 3.6.x, Tivoli has extended its management ability to the AS/400 platform. The Tivoli deployment management solution covers AS/400 systems as well as other platforms, and it enables you to perform seamless operations in deployment management.

The Tivoli deployment management solution consists of the following Tivoli applications:

- Tivoli Software Distribution
- Tivoli Inventory

These two applications rely on the Tivoli Management Framework for the delivery services. The following sections introduce the overview of Tivoli deployment management applications.

8.1.1 Tivoli Software Distribution

Tivoli Software Distribution provides facilities for the distribution and installation of software to managed systems including AS/400 Endpoints in a Tivoli environment.

Tivoli Software Distribution uses the facilities provided by the Tivoli Management Framework to distribute file packages in an efficient manner. Administrators use the profile paradigm used by most other Tivoli applications to define file packages to be distributed. These file packages can include any files (executable programs, data files, and so on) and scripts that will be executed before and after the distribution for a proper installation of the files on the target system.

The actual distribution process can use the Multiplexed Distribution (MDist) facility of the framework to optimize the use of the network. MDist is used to define nodes as repeaters so that they become fan-out points for the distribution. By defining an appropriate repeater hierarchy for your network environment, large file packages will only be moved once across slower links but will still reach multiple target systems. In Tivoli 3.6.x environments, the Endpoint Gateways are configured as an MDist repeater automatically.

8.1.2 Tivoli Inventory

One of the challenges in a network computing environment is keeping track of the hardware and software installed on each machine. Tivoli Inventory addresses this problem by providing the means to gather hardware and software information related to each system and then storing that information in a relational database. Queries and reports can be run to display the information in this database. Tivoli Inventory has the following three major advantages:

- It is based on the Tivoli Management Framework and, therefore, can be tightly and automatically integrated with other Tivoli applications.
- It stores inventory information in a Relational Database Management System (RDBMS) and, therefore, allows any non-Tivoli applications that can access SQL data to share the inventory information. Moreover, it benefits from the advanced features of an RDBMS system, such as scalability and performance.
- Tivoli Inventory has close links with other applications, such as Tivoli Software distribution, Tivoli Service Desk, and so on.

Tivoli Inventory is highly-configurable and integrated with Tivoli Enterprise applications. It enables you to perform seamless management operations.

8.2 Tivoli Software Distribution for AS/400

Tivoli Software Distribution for AS/400 allows you to distribute not only traditional AS/400 objects, such as DB2/400 files, RPG, or Cobol programs, but also objects to be used by other platforms, such as UNIX systems or Windows NT. In that case, you make use of the unique IFS architecture of the AS/400.

At this time, the AS/400 is only supported as an Endpoint; so, there is no possibility of using the AS/400 as a staging area for storing software or data that has to be distributed to client machines in the TMR. However, you can use the possibilities of an Integrated PC File Server.

Files that are distributed through Software Distribution need to be located in a source host prior to the distribution process. The source host has to be a Managed Node (including the TMR server), and the AS/400 is not supported as a Managed Node. This means that the software to be distributed will have to be sent either to a UNIX or Windows NT system before you send it to one or more AS/400 Endpoints.

A common method of sending files from an AS/400 to the source host is to use ftp, Client Access for AS/400, or MS Network Neighborhood. Obviously, this has to be done outside the Tivoli management environment.

There are various options for running configuration programs, such as pre-install and post-install exit programs, in Software Distribution configuration. You may want to check certain prerequisites before your installation or rename some files as a back-out option. After the installation, there can be a need for clean-up or further distribution of certain files to another file system within the AS/400 in case you are, for example, distributing client/server applications. You can define the following configuration programs:

- Before or after a distribution process
- During a commit operation
- Before and after removing file packages
- If an error occurs on a target that stops distribution or removal

Each configuration program (before, after, after removal, commit, and on error) can run on the targets. Every additional AS/400 object (QSYS.LIB) will be distributed to the AS/400 as a save file (*SAVF) and restored on the AS/400 prior to using it. If the file is already on the AS/400, it must be restored or installed as a program in an AS/400 library.

The exit programs can be any type of executable that can run on the targets, such as C program, CL program, REXX exec or batch job script. Normally, the C or CL programs are distributed as *SAVFs, and the REXX execs are distributed as ASCII text files.

Tivoli Software Distribution allows you to use the following containers to distribute objects:

File Package

This container is ideal for distributing applications that include unattended redirected installation programs. These programs are provided by the original software manufacturer and can be used to complete the installation automatically from a

remote or local drive. The program can auto detect a target's configuration and install any device-dependent drivers that are required. This is the appropriate way if you do not wish to handle so many files or if you would like to always distribute the most recent versions of files that are changing frequently.

File Package Block

This container enables you to take a snapshot of several file packages that contain files or data and create a single entry in packaged form.

Autopack

This container is an excellent replication tool. The replication process can be described in three steps:

1. Install an application the normal way, preferably on a pristine machine. A pristine machine does not have any users or other applications running on it.
2. Reverse engineer the installation process to find out about the files that were copied and the configuration files that were modified.
3. A customized procedure properly reinstalls the application without the original installation program.

Autopack is used in an environment where you have standard groups or classes of machines. You can use a reference PC to build and test Autopacks. This is similar to the NetView DM product Diskcamera that can be found in Netfinity for AS/400.

There is an additional tool provided on the Tivoli AS/400 CD-ROM called the TME 10 Object Packaging Utility for AS/400. The purpose of this tool is to provide an easy-to-use Windows interface to select files that are to be packaged and sent to the targets on later distribution using Tivoli. The Object Packaging Utility is useful in the AS/400 environment with its various file systems (the IFS structure).

The Tivoli Object Packaging Utility for AS/400 makes use of Tivoli's file package block. A file package block contains the file package definition, file package attributes, configuration programs, input files, and the data in the files. Figure 248 on page 405 shows the structure of file the package block.

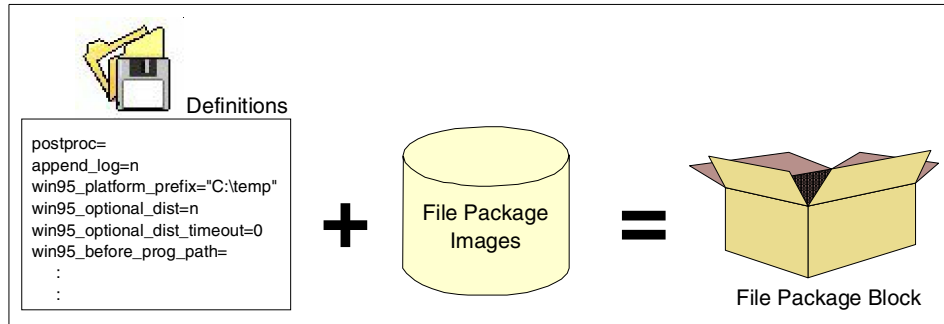


Figure 248. File package block components

You do not have to use this tool in all cases. If you just want to distribute a few files, it can be done without any tool. This utility has a different purpose, as described earlier, and cannot be used in an AS/400 environment. If you want to use the Tivoli Object Packaging Utility for AS/400, you need Client Access for Windows 95 or NT to have access to the IFS file system and to make a connection between Windows PC and the AS/400 that you will be using to build AS/400 objects package.

The Tivoli Object Packaging Utility for AS/400 can be installed on Windows 95 or Windows NT 3.5.1 or 4.0.

8.2.1 Installing Tivoli Software Distribution for AS/400

Tivoli Software Distribution for AS/400 consists of optionally installable patches that allow Version 3.6.1 of the Tivoli Software Distribution product to support AS/400 Endpoints. Therefore, before installing these patches, both the Tivoli Software Distribution Version 3.6.1 base product and the AS/400 Endpoint must be installed. For complete details on installing Version 3.6.1 of Tivoli Software Distribution, refer to the *TME 10 Software Distribution 3.6 User's Guide*, GC31-8330. Refer to Chapter 3, "AS/400 Endpoint installation and deployment" on page 37 of this redbook, for AS/400 Endpoint installation details including:

- Hardware requirements
- Software requirements
- Installation methods
- A complete AS/400 Endpoint Enablement Patch listing

Additionally, be sure to completely review the *TME 10 AS/400 3.6 Release Notes*, GI10-4775, since this document provides the most current information for the product including:

- New features
- Installation notes
- Documentation changes
- Defects, workarounds and limitations

3.6.1-COU-0001

Tivoli Software Distribution Gateway for AS/400, Version 3.6.1. is the enablement patch necessary for Tivoli Software Distribution for AS/400. It contains the Software Distribution methods to be installed on the Software Distribution Gateway to allow it to support AS/400 Endpoints. This patch needs to be installed on any Software Distribution Gateway that will support AS/400 Endpoints.

Note

Just as when managing TMA Endpoints, it is a good rule of thumb to install the Software Distribution Gateway on all of the Endpoint Gateways. It is also recommended that you install this Tivoli Software Distribution Gateway for AS/400 Endpoints patch on all your Software Distribution Gateways. By doing this, you are assured that the AS/400 Endpoints will remain supported should you reconfigure your Endpoint Gateways or migrate the AS/400 Endpoints.

To install sapack on the AS/400 Endpoint, do not forget to run the following script:

```
packinst.sh ep_label
```

To verify that the package is installed, they can check for the existence of the sapack in the QTMELCF library. This can be done using the display library command, `DSPLIB`, and verifying that the SAPACK program is found as illustrated in Figure 249 on page 407.

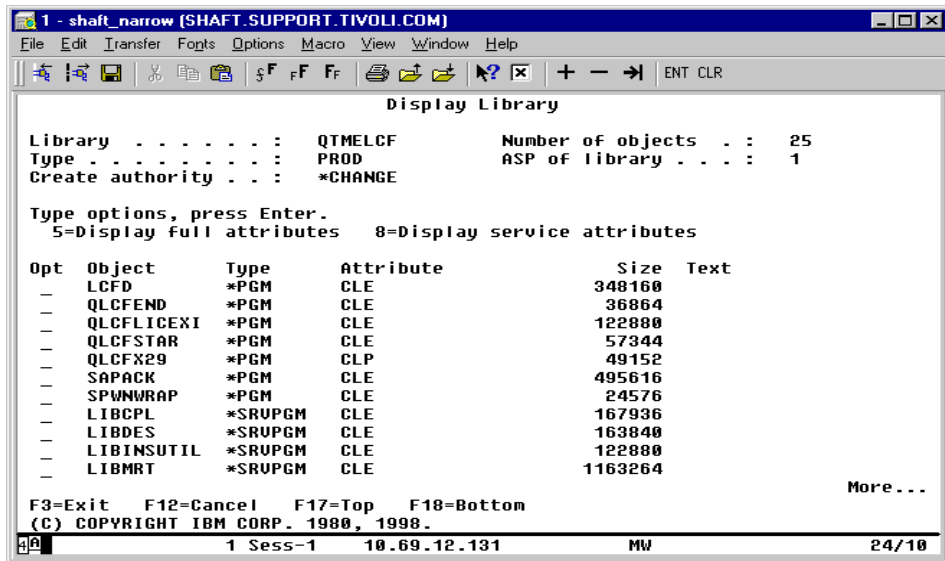


Figure 249. Display library of QTMELCF

8.2.2 Using Tivoli Software Distribution for AS/400

Tivoli Software Distribution provides three ways of performing software distribution, and Tivoli Software Distribution for AS/400 supports two of those three ways. First, we introduce the use of the file package option.

8.2.2.1 File packages

This section explains how to configure and use file packages in your management environment:

1. Place the objects that will be distributed in a directory on the source host using ftp, Client Access (in case of Windows NT), or another method.
2. Create a Software Distribution profile manager with dataless mode. In this profile manager, you can create various profiles for the distribution. In this example, we are going to distribute the Tivoli Enterprise Console Message Adapter for AS/400.

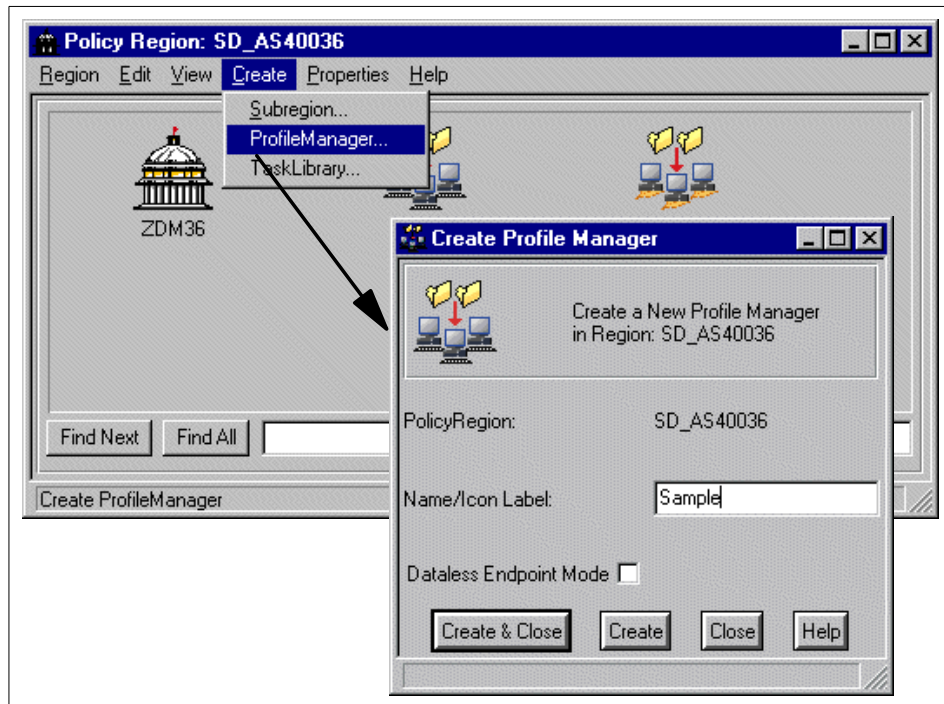


Figure 250. Creating a profile manager for Tivoli Software Distribution

Note

When you create a file package, we strongly recommend that you create the file package in a database profile manager. Dataless profile manager should be used as subscribers to database profile managers for distributing file packages to the Endpoint targets.

3. Double click on the profile manager and create a new profile. Select **File Package** as the profile type, and put the name into the label field as shown in Figure 251 on page 409. As we described, the Autopack option is not supported by the AS/400 Endpoint.

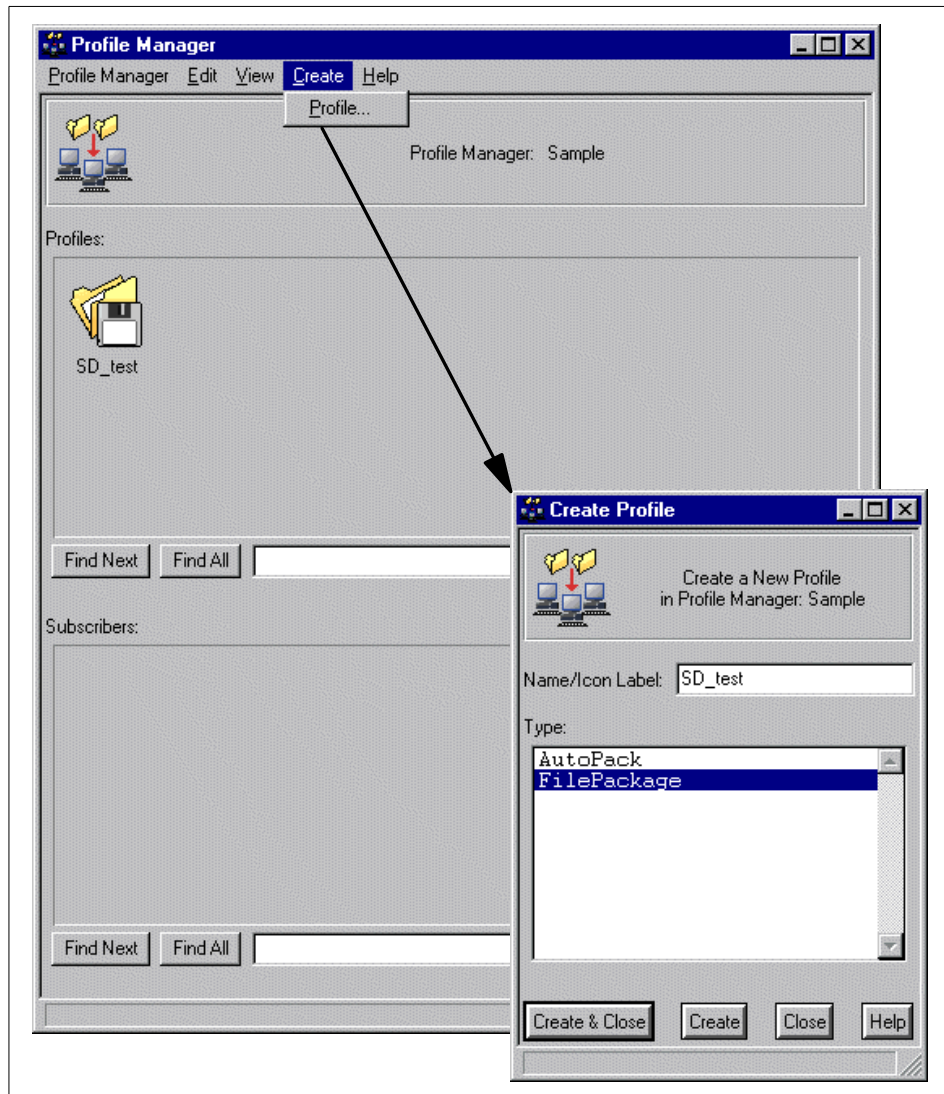


Figure 251. Creating a file package in database profile

4. We have created a profile for the distribution of the AS/400 BARC (configuration) scripts. We need to configure the specific options for this profile. The configurations can be done with two screens as shown in Figure 252 on page 410.

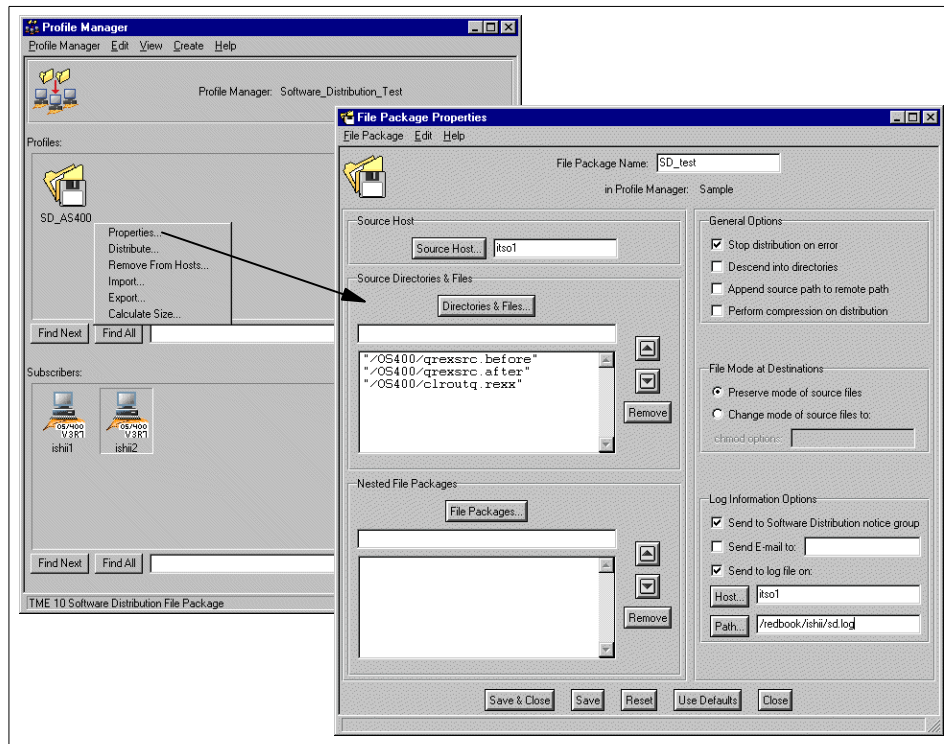


Figure 252. Configuring file package properties

Use the Destination Directory Path field to specify the destination path to AS/400 subscribers. In the Destination Directory Path field, type the full path to which you would like to distribute the file package on AS/400 subscribers. This can be an IFS or QSYS path name on the AS/400, but the path name must always be specified in IFS format (for example, /qsys.lib/qgpl.lib/dist.file/).

1. If you distribute a save file, specify xxx.file, not SAVF.MBR.
2. Only AS/400 *SAVF can be distributed to a QSYS destination. All types of objects can be contained in the *SAVF object.
3. Tivoli will create the *SAVF on the AS/400; there is no need to create an empty *SAVF beforehand.
4. Press the **Translate to EBCDIC** radio button to indicate that the contents of the distributed files should be converted to EBCDIC on the AS/400 subscriber. You should specify this option if you are distributing text files to an IFS destination path. Save files distributed to QSYS should not be translated.

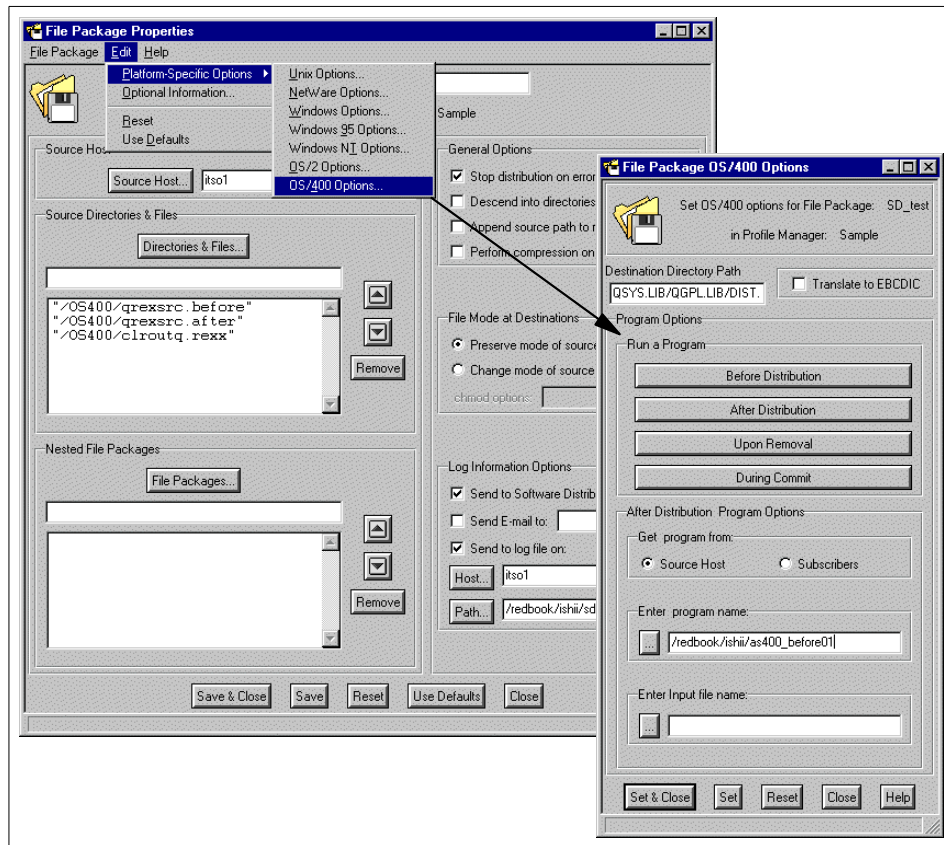


Figure 253. Configuring file package options for an OS/400 platform

Set configuration program options by pressing the **Run a Program** button. These buttons display options that enable you to run configuration programs. To run the programs on an AS/400 subscriber before or after a file package distribution, complete the following steps:

1. Press the program button that corresponds to the configuration program you wish to run.
2. Specify the location of the configuration program and input files using the Get program from radio buttons. If you press the Source host radio button, Software Distribution copies each program from the source host to a temporary file on each subscriber. Then, the programs run on each subscriber before or after the file package distribution and remove each temporary file from each subscriber upon completion. If you press the Subscriber radio button, Software Distribution runs each program directly on the subscriber.

3. In the Enter Program Name field, enter the full path of the program to run. If you specify that the program is located on the AS/400 subscriber, the path must be a QSYS path specified in IFS format. If you do not know the full path of the program and you specified that the program is located on the source host, press the ... button to display a browser dialog that enables you to traverse the directory structure of the machine on which it resides. You must enter the name of the program that exists at the time of distribution. The program must be located on the system (source or subscriber) specified by the Get program by radio buttons. If you specify multiple programs, you must separate them with commas.

Note

Software Distribution does not check the existence of configuration programs on either source host or subscribers until the time of the file package distribution. If the program is not found at the distribution, Software Distribution logs an error.

4. If you want to specify an input file with the configuration program, enter the full path in the Enter Input file name field. If you specify that the configuration program is located on the AS/400 subscriber, the path must be a QSYS path specified in IFS format. If you do not know the full path of the program and you specified that the program is located on the source host, press the ... button to display a browser dialog that enables you to traverse the directory structure of the machine on which it resides. This input file must be located on the same system (source host or subscriber) as the program indicated in the Enter program name field. The input file path is passed as the second argument to the configuration program.
5. If you specify multiple programs of the same type, you can also specify multiple input files separated by commas. If the program does not require an input file but subsequent programs do, you must specify the input files in the correct order and preceded by the same number of commas as programs. Do not follow the commas with spaces. For example, if you specify the path in the Enter program name field as follows:

```
/qsys.lib/prog1.pgm/qsys.lib/prog2/pgm,/qsys.lib/usr.lib/prog3.pgm
```

The prog2 does not require an input file; you must specify the following in the Enter Input File name field:

```
/qsys.lib/usr.lib/input1.file,, qsys.lib/usr.lib/input3.file
```

If you want to specify an input file that resides in a different location than the specified configuration program, refer to the `os400_xxx_input_from_src`

keyword (where xxx is the program type) in the *Tivoli Software Distribution Reference Manual*, SC31-8331, for more detailed information.

6. If you are running a before program, select the **Skip distribution to a host on a non-zero exit code of file** check box to skip distributing the file package to a subscriber if the program fails and returns a non-zero exit code. This option allows you to implement dependency checking so that files are only distributed to subscribers if dependencies are met. If you would like to set a time-out value for configuration programs, you can do it using the `wsetfpopts` command. For more detailed information, refer to the *TME 10 Software Distribution 3.6 Reference Manual*, SC31-8331, and the *Tivoli Enterprise Performance Tuning Guide*, SG24-5392.
7. Press the **Set & Close** button to save all distribution options of the file package.
8. Complete the File Package Properties window with the name of the source host and the specification of the files to be distributed. Make sure that you also fill in the name and the path of a file that can receive the log information. Then, press the **Set & Close** button.
9. Distribute the file package to the appropriate subscribers.

Note

The Object Packager utility could have been used to create and distribute file package block that consists of file packages. However, file package does not support file members with different names than the files themselves. For example, if we distribute the following:

```
QGPL/FIN_APP (MEMBER_1)
```

where MEMBER_1 is a member of the file, it will be restored as

```
QGPL/FIN_APP (FIN_APP)
```

losing the member identifier of the file.

8.2.2.2 File package block and object packaging utility for AS/400

As we mentioned, Software Distribution provides two different types of distribution method to AS/400 Endpoints: File package and file package block.

This section explains how to configure and use a file package block in your management environment.

A special graphical tool, called the Tivoli Object Packaging Utility for AS/400, is available on the Tivoli AS/400 CD-ROM to make object packaging process easier. Figure 254 on page 415 shows the GUI of the object packaging tool. The Object Packaging Utility for AS/400 provides a Windows interface to AS/400 systems in the network. Because an AS/400 Endpoint cannot currently serve as the source host of a distribution, the Object Packaging Utility enables you to create a file package block of files and directories that contain AS/400-specific files to be packaged, set distribution options, and specify any configuration programs that will run on the target during a distribution. The Object Packaging Utility creates a file package block of the selected data, which you can distribute using Tivoli Software Distribution commands.

Note

To use the Tivoli Object Packaging Utility for AS/400, the Object Packaging Utility must be installed on a Windows PC, and the PC must be connected to the AS/400 source host using Client Access/400. The Object Packaging Utility is installed from the /obj_pack directory of the Tivoli for AS/400 CD-ROM. For more information on installing or using Client Access/400, refer to the product manual that accompanies the Client Access/400 product.

This section explains how to create file a package block and how to distribute it to the targets in your management environment.

1. Start an IBM AS/400 Client Access connection to the AS/400 to be used as the source host. Make sure that you have made a connection to the AS/400 that you want to use with IBM AS/400 Client Access.
2. Then, go to the Start menu, and select the **Programs -> Tivoli -> Object Packaging Utility for AS/400** menu, or run the Object Packaging Utility by double-clicking on the **as4pack.exe** icon from the ObjPackAS400 window. See Figure 254 on page 415.

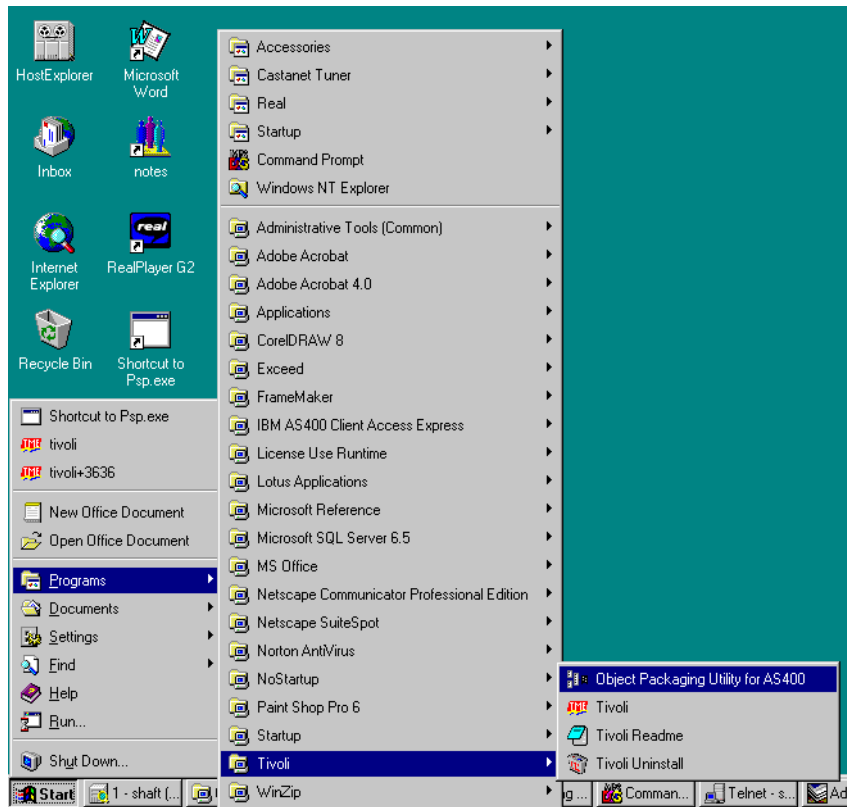


Figure 254. Starting Tivoli Object Packaging Utility for AS/400

3. The Connect to AS/400 dialog is displayed. Enter the name of the AS/400 to be used as the source host as shown in Figure 255 on page 416.

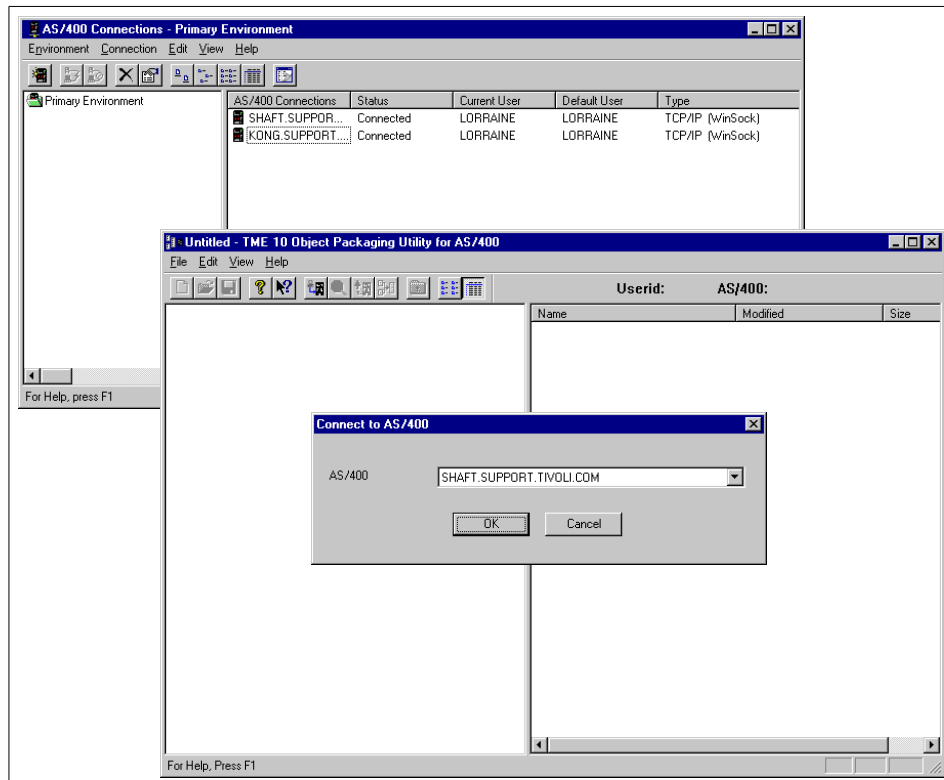


Figure 255. AS/400 Client Access and Tivoli Object Packaging utility for AS/400

The Tivoli Object Packaging Utility for AS/400 displays the first level of the directory tree for the source AS/400. The GUI is similar to Windows Explorer. The directory tree displays both available and unavailable directories and files. The box that is located on the left of each tree entry is used to expand and hide a particular sub-tree. By selecting the plus sign (+) in front of a directory icon, you can expand the list to view directory contents.

Note

Although you can view the entire directory structure of the AS/400, you can only select directories and files that appear next to a square check box.

The state box that is located on left of the folder icon is used to select a directory for distribution. Clicking on the box toggles it between the deselect and select states. As directories or objects are selected, the levels in the tree above the selected item become shaded. As you can see in Figure 256, it is

easy to tell which tree item contains selected objects even when they are not exposed.

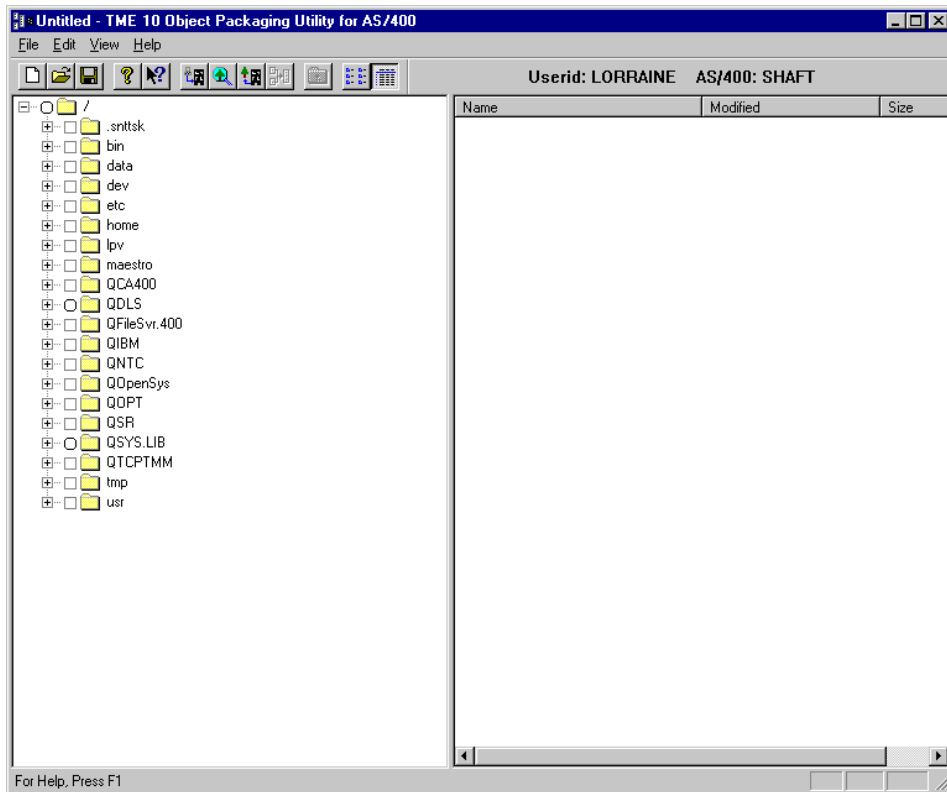


Figure 256. Tivoli Object Packaging Utility for AS/400

4. Select the files to be packaged. In the example in Figure 256 on page 417, we selected four library files. Note that, with the Object Packaging Utility, we do not first have to save the data to save files.

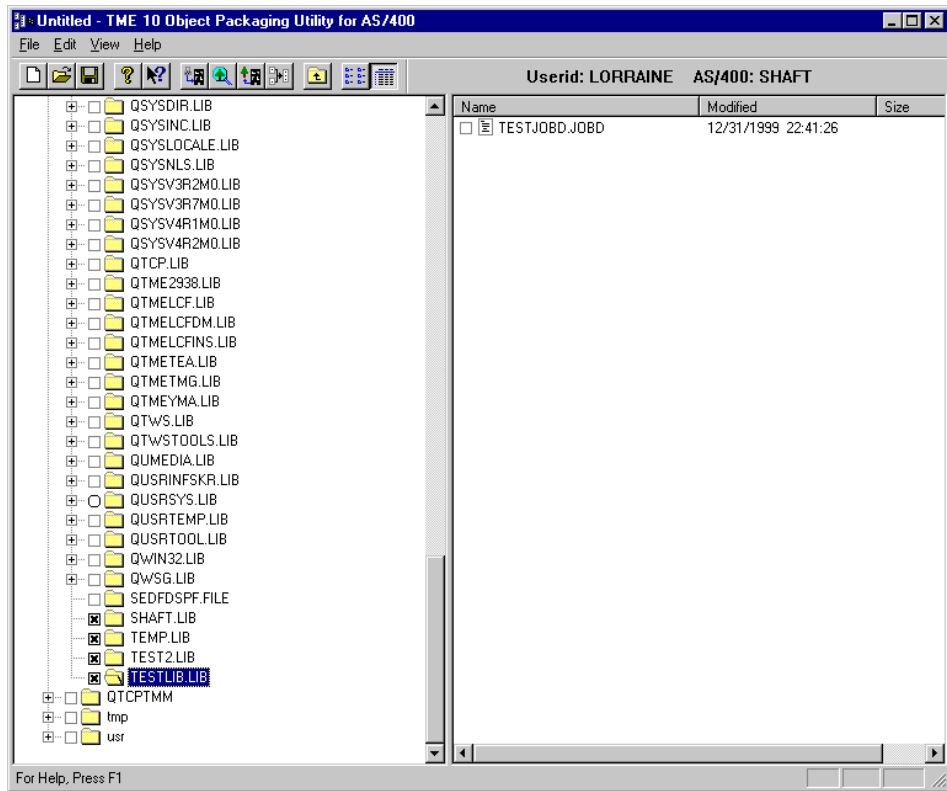


Figure 257. Selecting the files to be packaged

5. After you select the files that you want to create file package block with, select the **File -> Create File Pack** menu from the Object Packaging Utility menu bar. The **Create Tivoli Filepack** dialog box that shows the files you have selected will be displayed as shown in Figure 258 on page 419.

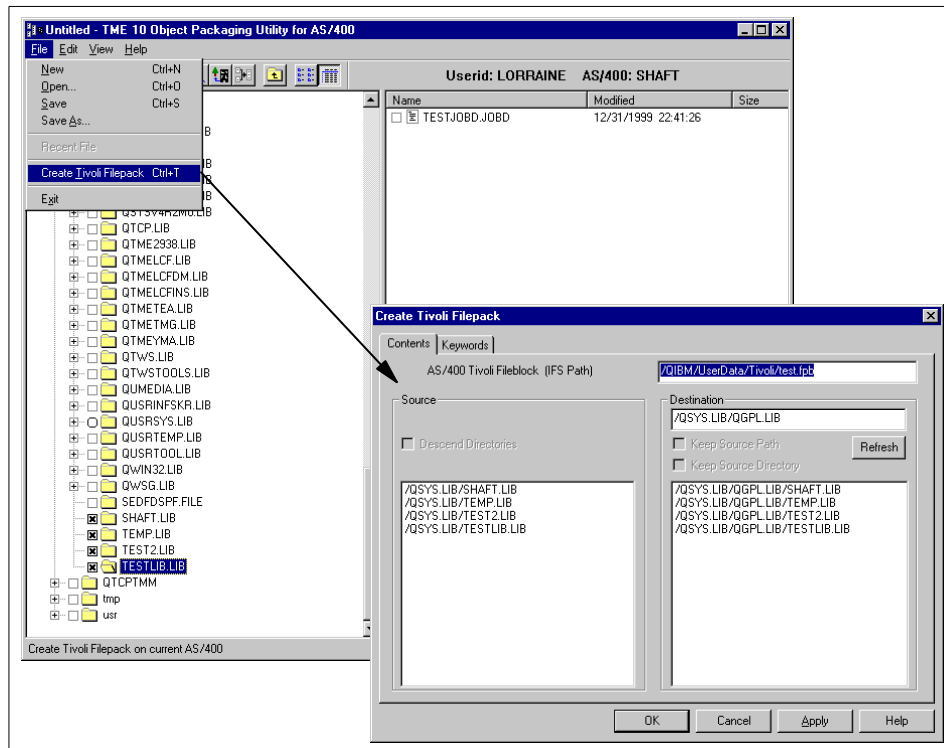


Figure 258. Create Tivoli Filepack dialog

6. In the Create Tivoli Filepack dialog box, specify the file package block Destination library (or IFS folder) and the IFS (AS/400 Integrated File System) path for the file package block. We selected QGPL library for the File Package destination and specified that the test.fpb File Package Block should be created in the /QIBM/UserData/Tivoli/ directory on the AS/400 IFS.
7. Select **Keywords** to specify the log file name and path (if you need to), and press the **Apply** button on the Create Tivoli Filepack dialog box, shown in Figure 259 on page 420, to apply the destination path.

Figure 259. Keywords setting dialog box

8. Press the **OK** button in the Create Tivoli Filepack dialog box.
9. Use binary FTP to transfer the created file package block to the TMR server.
10. Use the `wdistfpblock` command at the TMR server to distribute the file package block. For example, to distribute the file package block `test.fpb` that has been transferred from the `/dist` directory on the TMR server to AS/400 `shaft.support.tivoli.com`, use the following command:

```
wdistfpblock /dist/test.fpb @shaft.support.tivoli.com
```

Note

- When using the Object Packaging Utility, it is not necessary to restore the files at the target; this is done automatically.
- When distributing a file package block, you must use the `wdistfpblock` command. There is no GUI operation supported to perform a file package block distribution.

8.3 Tivoli Inventory for AS/400

Tivoli Inventory is a hardware and software inventory gathering application designed to help system administrators and accounting personnel manage the complexity of a distributed client/server enterprise.

Tivoli Inventory for AS/400 enables you to scan inventory information on AS/400 machines and to use this data to manage your AS/400 systems. All information is stored in the configuration repository (Inventory database), which is a relational database management system (RDBMS) that holds information about system configurations in your management environment. Tivoli Inventory provides the scripts necessary to create the configuration repository as well as to install the configuration repository schema.

Operations of Tivoli Inventory for AS/400 (for example, scanning targets or query Inventory data) are the same as other platforms, such as UNIX. Tivoli Inventory uses RIM object and RDBMS; however, we do not describe Inventory setup information in this redbook. Refer to the *Tivoli Inventory User's Guide*, SC31-8381 for more information.

Tivoli Inventory for AS/400 collects the following information about AS/400 hardware:

- Processor installed
- System memory installed
- Disk drives installed
- CD-ROM drives installed
- Printers installed
- Tape drives installed
- Diskette drives installed
- Network interface cards installed

Tivoli Inventory for AS/400 also collects information about installed software and installed PTFs. The following sections discuss how to use and configure Tivoli Inventory for AS/400 in your environment.

8.3.1 Installing Tivoli Inventory for AS/400

Tivoli Inventory for AS/400 consists of optionally-installable patches that allow Version 3.6.1 of the Tivoli Inventory product to support AS/400 Endpoints. Therefore, before installing these patches, both Tivoli Inventory, Version 3.6.1 base product, and the AS/400 Endpoint must be installed. For complete

details on installing Version 3.6.1 of Tivoli Inventory, refer to the *Tivoli Inventory User's Guide*, GC31-8381. Refer to Chapter 3, "AS/400 Endpoint installation and deployment" on page 37 of this redbook, for AS/400 Endpoint installation details including:

- Hardware requirements
- Software requirements
- Installation methods
- A complete AS/400 Endpoint Enablement Patch listing

Additionally, be sure to completely review the *TME 10 AS/400 3.6 Release Notes*, GI10-4775, since this document provides the most current information for the product including:

- New features
- Installation notes
- Documentation changes
- Defects, workarounds and limitations

The enablement patches necessary for Tivoli Inventory for AS/400 are:

- **3.6.1-INV-0001** - Tivoli Inventory for AS/400 Endpoint, Version 3.6.1 contains the server and database schema scripts to be installed on the TMR server for customizing your Inventory repository for AS/400 PTF information.

Note

It is very important that this patch only be installed on the TMR server. Installing this patch on a Managed Node will corrupt the installation.

- **3.6.1-INV-0002** - Tivoli Inventory Gateway for AS/400 Endpoint, Version 3.6.1 contains the Inventory methods which the Inventory Gateway will need to support AS/400 Endpoints. This patch needs to be installed on any Inventory Gateway that will support AS/400 Endpoints.

Note

Just as when managing TMA Endpoints, it is a good rule of thumb to install the Inventory Gateway on all of the Endpoint Gateways; it is also recommended that you install this Tivoli Inventory Gateway for AS/400 Endpoints patch on all of your Inventory Gateways. By doing this, you are assured that the AS/400 Endpoints will remain supported should you reconfigure your Endpoint Gateways or migrate the AS/400 Endpoints.

8.3.2 Using Tivoli Inventory for AS/400

As we mentioned, there is no difference between using Tivoli Inventory for AS/400 and other platforms. Tivoli Inventory has two different types of operations:

- Scan inventory data
- Query inventory data

Figure 260 on page 424 shows the Inventory process flow of each operation and the interactions between each of the components.

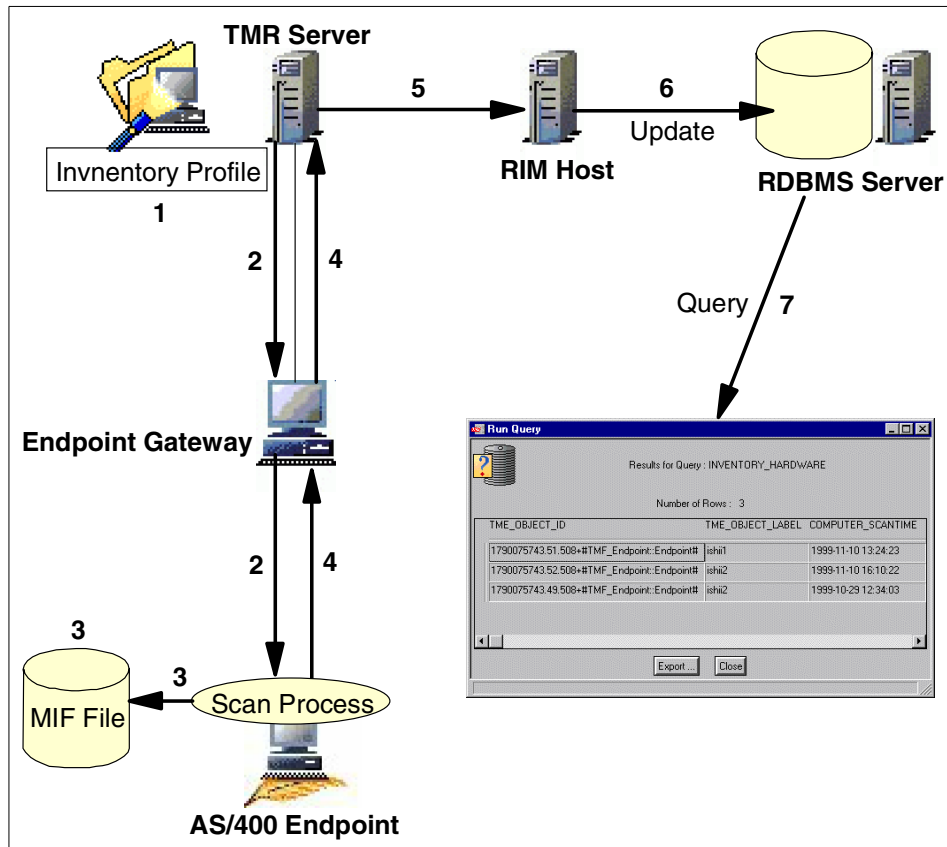


Figure 260. Tivoli Inventory operations

As you can see, the scanning operation also includes two types of processes: Scanning target and updating Inventory database. Both processes can be defined in one Inventory profile. Once the scanning operation has been completed, the inventory data is stored in an Inventory database, and you can query the data using the Tivoli Desktop.

Note

As we described, scanning target and updating Inventory database processes can be defined in a single Inventory profile. However, we recommend that you divide these processes into two different Inventory profiles, especially in a large-scale environment. For example, one Inventory profile performs scanning process and another profile performs updating database. Since Inventory processes are very heavy, dividing processes improves the performance of Inventory operations.

For more detailed information, please refer to the redbook, *Tivoli Enterprise Performance Tuning Guide*, SG24-5392.

8.3.2.1 Creating and configuring Inventory profile

Once you have installed and configured Tivoli Inventory software, RIM object, and RDBMS, you can create an Inventory profile to scan targets. There is no difference between creating an Inventory profile for AS/400 targets and other platform targets. Figure 261 shows the procedure for creating an Inventory profile.

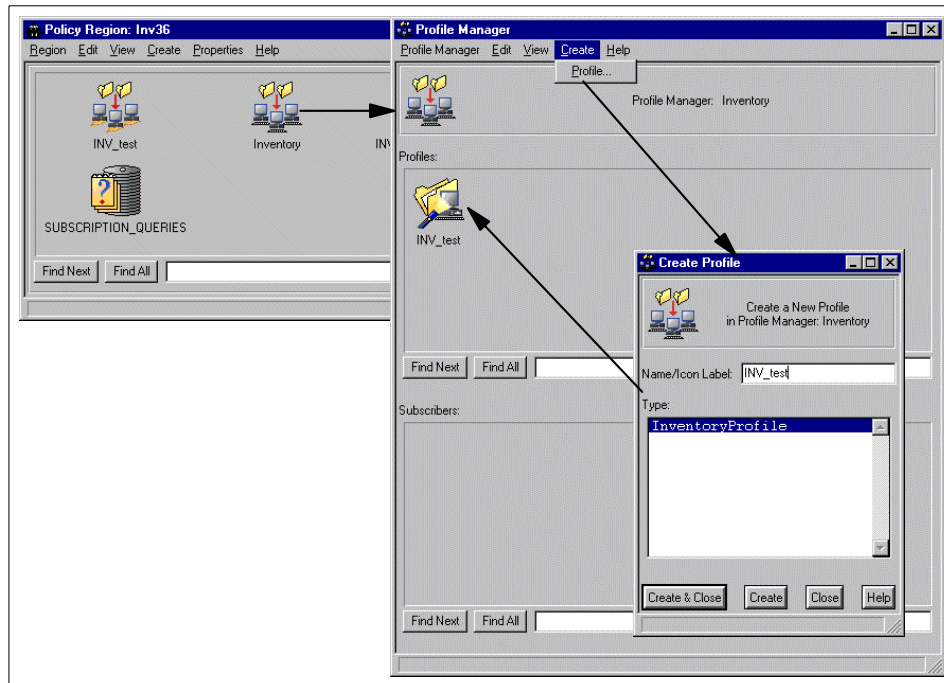


Figure 261. Creating Inventory profile

Then, you need to define scanning targets as subscribers. The add subscribers operation is exactly the same as other Tivoli core applications, such as Software Distribution.

Note

When you create an Inventory profile, we strongly recommend that you create the Inventory profile in a database profile manager. Dataless profile manager should be used as subscribers to database profile managers for distributing an Inventory profile to the Endpoint targets.

8.3.2.2 Configuring Inventory profile

Figure 262 shows an example of Inventory profile configuration. There is also no difference between configuring an Inventory profile for AS/400 and other platforms.

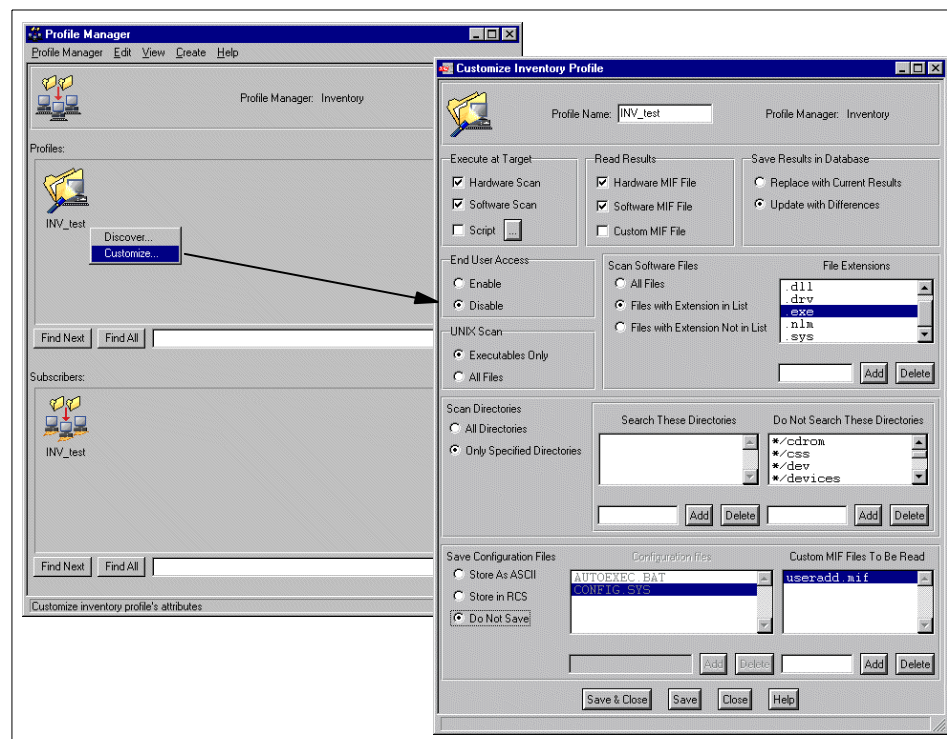


Figure 262. Configuring Inventory profile

As you can see, you can define the following different processes in a single Inventory profile:

- Hardware scanning
- Software scanning
- Collecting hardware MIF files
- Collecting software MIF files

These processes (hardware and software scans) are carried out by default, but it is not recommended from an Inventory performance point of view, especially in a large-scale environment. To improve Inventory performance and throughput, you can divide each process into different Inventory profiles. Refer to the redbook *Tivoli Enterprise Performance Tuning Guide*, SG24-5392, for more detailed information about Inventory profile configurations.

Once you have completed Inventory profile configuration, you can distribute the Inventory profile to scanning targets. Figure 263 on page 428 shows the procedure for Inventory profile distribution.

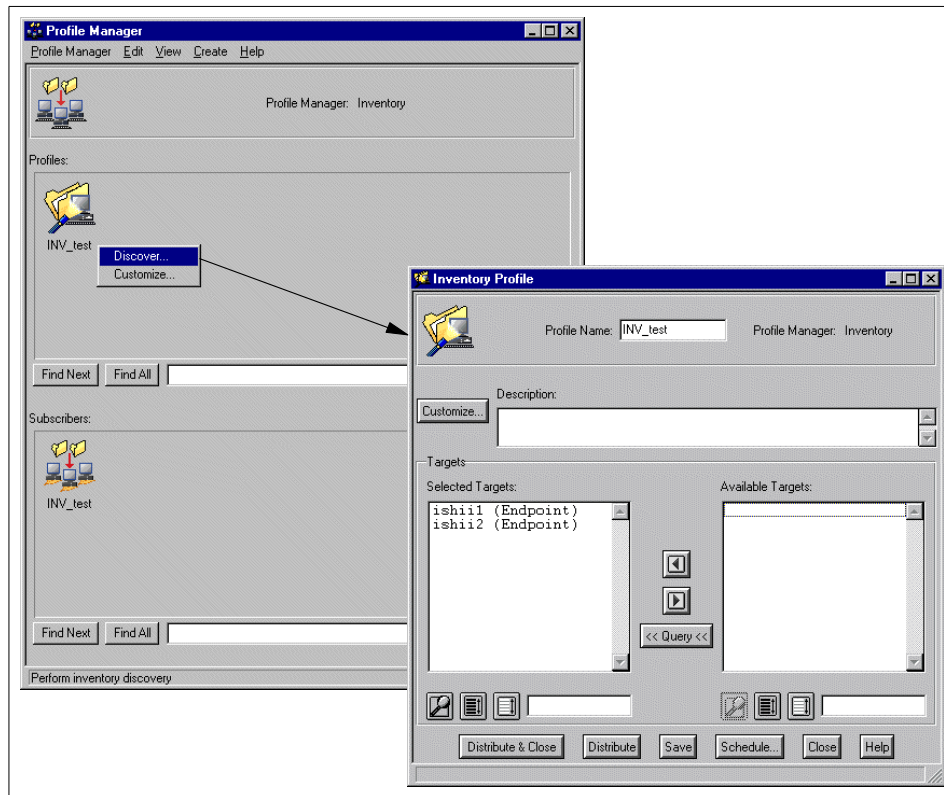


Figure 263. Distributing Inventory profile

If the distribution is completed without problems, the software and hardware scanning information (MIF file) has been stored in the Inventory database (RDBMS). After distribution completion, you can perform a hardware inventory query or software inventory query on the Tivoli Desktop.

8.3.2.3 Querying inventory

In the Tivoli management environment, a query library facility enables you to create, manage, and logically-organize Tivoli queries. For example, you can create a query library to hold queries that selectively retrieve information about AS/400 software inventory information. Tivoli provides the customized INVENTORY_OS400_SOFTWARE query and the INVENTORY_OS400_PTF query to retrieve information about AS/400 inventory. It also provides the PTF_VIEW table in which updated PTF information is stored.

The INVENTORY_OS400_SOFTWARE query enables you to retrieve information about software that is installed on AS/400 Endpoints. To install

this query, run the `inventory_queries.sh` script that is provided with Tivoli Inventory:

```
inventory_queries.sh region
```

This script automatically installs a query library and all predefined queries. See the *Tivoli Inventory User's Guide*, GC31-8381, for information about installing queries with the `inventory_queries.sh` script.

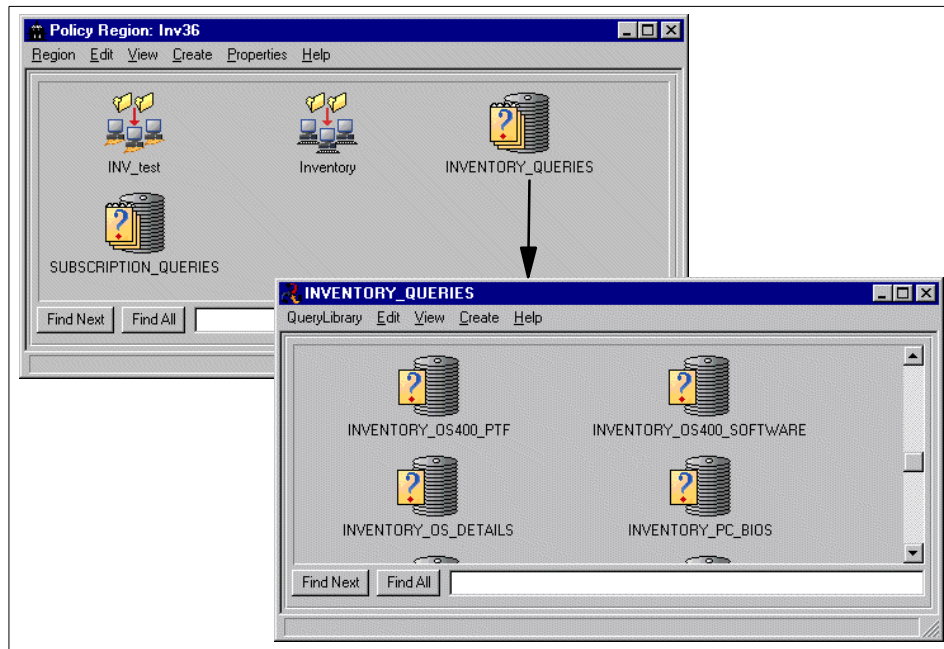


Figure 264. Query libraries for OS/400 software information

To query the OS/400 software information stored in the Inventory database information, perform the following steps:

1. Select **INVENTORY_QUERIES** from the policy region shown in Figure 265 on page 430.

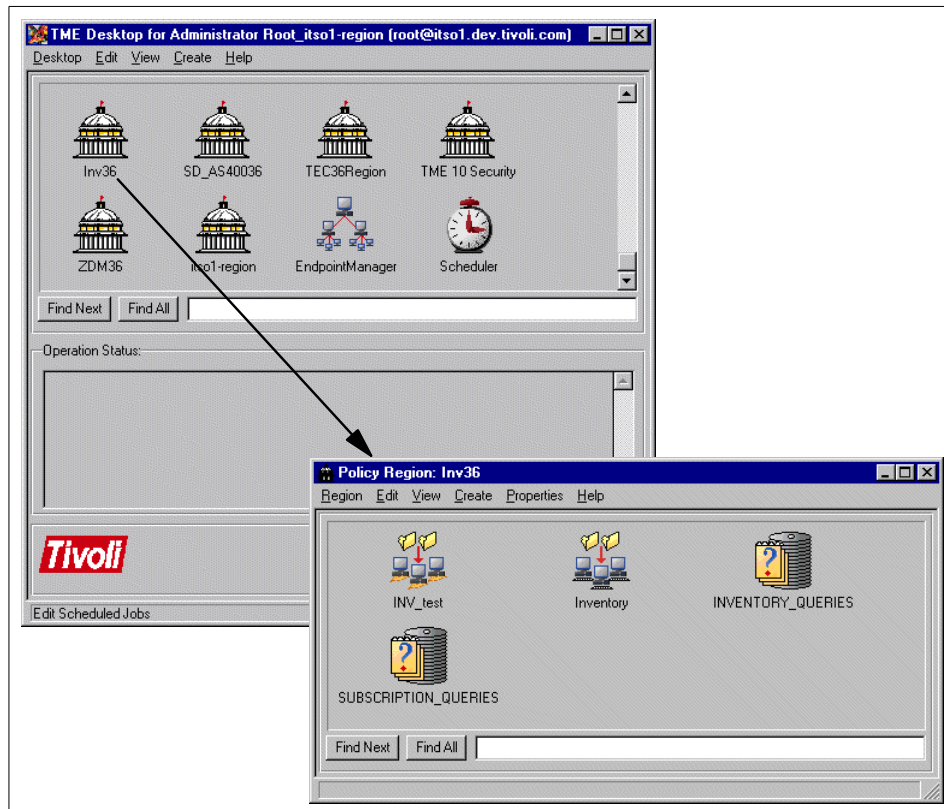


Figure 265. Using Tivoli Inventory on Tivoli Desktop

2. Select **INVENTORY_OS400_SOFTWARE** from **INVENTORY_QUERIES**.
3. From the INVENTORY_OS400_SOFTWARE dialog box, select **Run Query** as shown in Figure 266 on page 431.

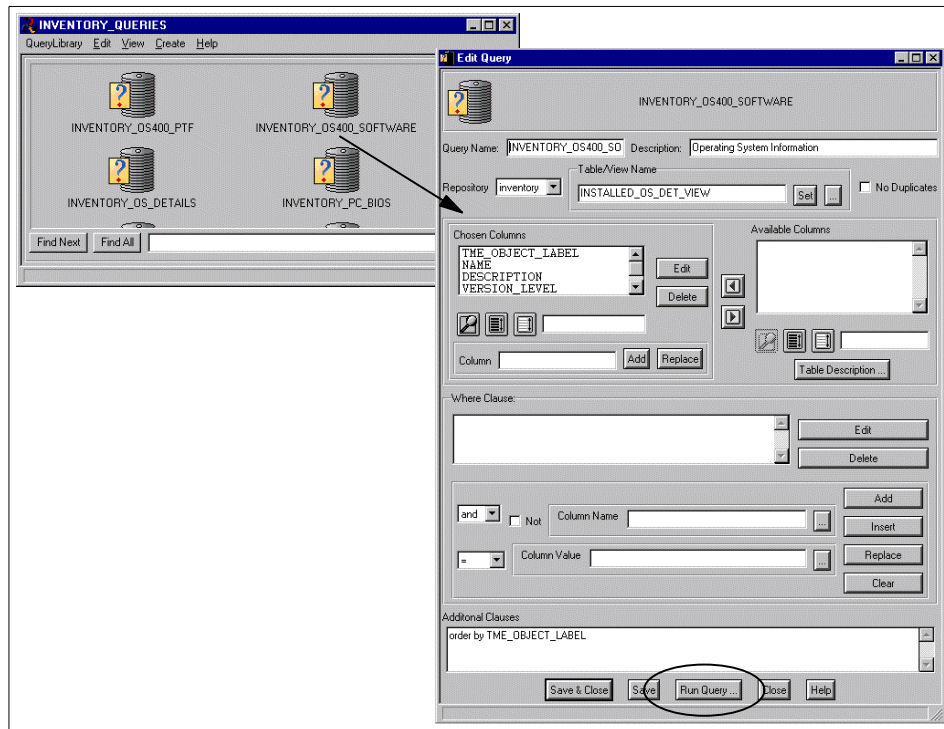
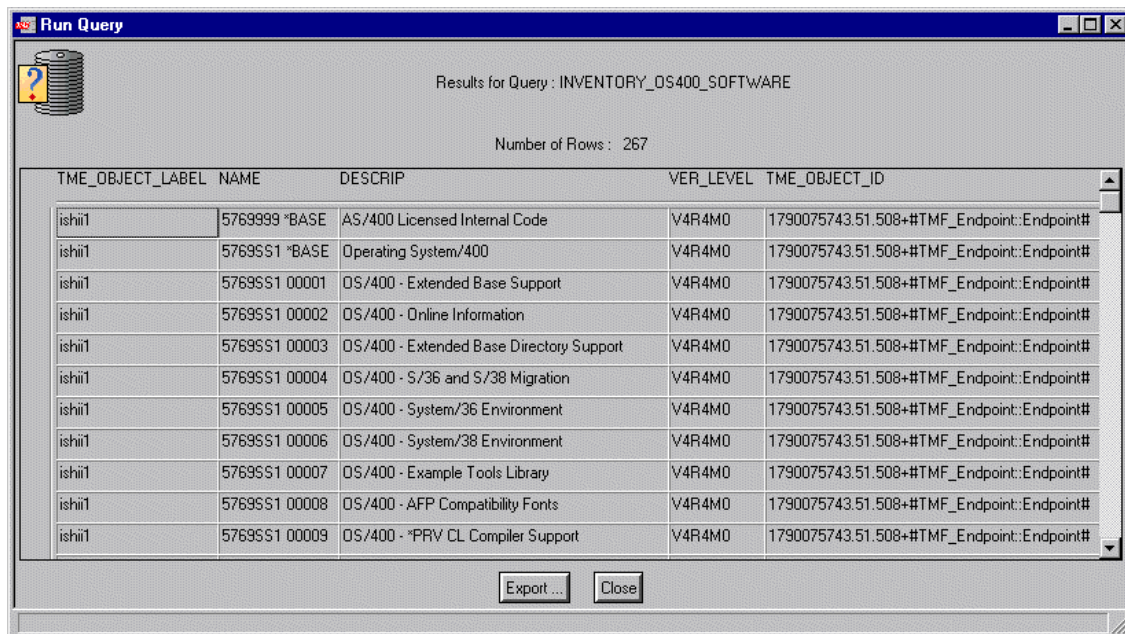


Figure 266. OS/400 software inventory query

4. You will see the OS/400 software query results shown in Figure 267 on page 432.



Run Query

Results for Query : INVENTORY_OS400_SOFTWARE

Number of Rows : 267

TME_OBJECT_LABEL	NAME	DESCRIP	VER_LEVEL	TME_OBJECT_ID
ishii1	5769999 *BASE	AS/400 Licensed Internal Code	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 *BASE	Operating System/400	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00001	OS/400 - Extended Base Support	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00002	OS/400 - Online Information	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00003	OS/400 - Extended Base Directory Support	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00004	OS/400 - S/36 and S/38 Migration	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00005	OS/400 - System/36 Environment	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00006	OS/400 - System/38 Environment	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00007	OS/400 - Example Tools Library	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00008	OS/400 - AFP Compatibility Fonts	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#
ishii1	5769SS1 00009	OS/400 - *PRV CL Compiler Support	V4R4M0	1790075743.51.508+TMF_Endpoint:Endpoint#

Export ... Close

Figure 267. OS/400 software query results

8.3.2.4 Scanning for AS/400 PTFs information

AS/400 Program Temporary Fixes (PTFs) are supported through a user-defined MIF file in Tivoli Inventory. To scan for AS/400 PTF Inventory information, you must first extend your configuration repository by running the supplied scripts. You must also create and customize an Inventory profile that will run a program from a REXX script or a batch job and read the results of the custom MIF file named ptf.mif.

To customize an Inventory profile for AS/400 PTFs, complete the following steps from the Tivoli Desktop:

1. Extend your configuration repository by running the `tivoli_dbtype_schema_as400_ptf.sql` script located in the `$BINDIR/TME/INVENTORY/SCRIPTS/RDBMS` directory on the TMR server.
2. Right click on the created Inventory profile, and select **Customize**. Add a script to the profile as follows:
 1. Under the Execute at Target field, shown in Figure 268 on page 433, select the **Script** option.
 2. Press the ... button to open the Edit Target Script dialog box.

3. Enter the following script:

```
"CALL PGM(QLCF CACHE/OS4_PTFDIS) PARM (<prodid1 prodid2 ...prodidn>)"
```

where: `prodid1 prodid2 ...prodidn` specifies the AS/400 product identifier numbers, such as 5763CM1 or 5763DB1.

You can display your AS/400 product identifier numbers via the GO LICPGM menu on the AS/400 machine. If the PARM argument is omitted (as shown in the example), all installed software products will be scanned for PTFs.

Note

When you specify the script, you must include double quotes ("xxx") around a CALL command as shown in our example.

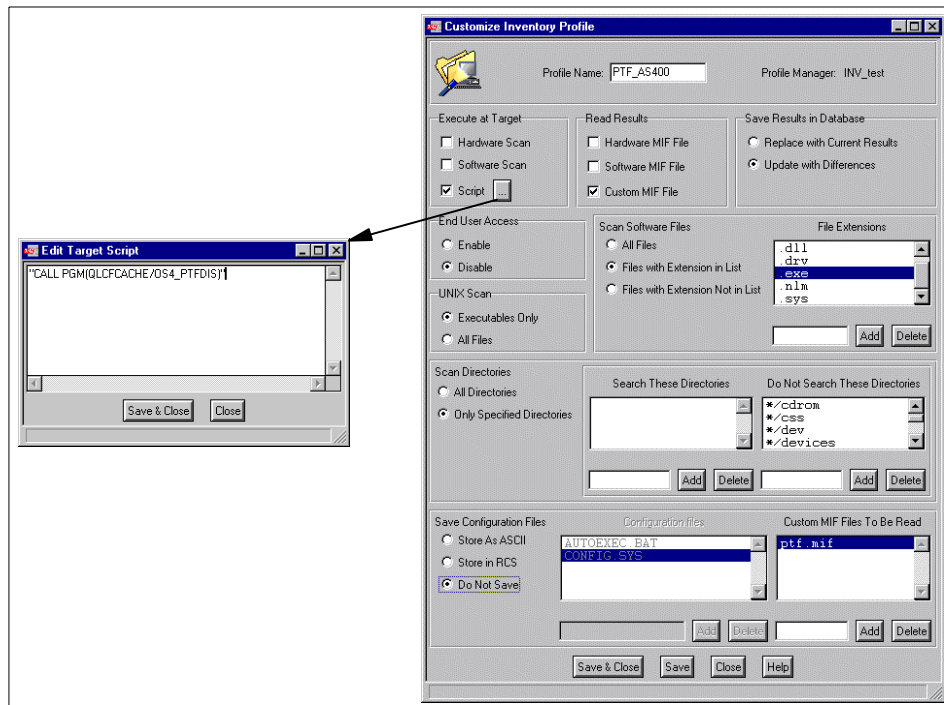


Figure 268. Inventory profile for scanning AS/400 PTFs

3. To read the results of the script, add the `ptf.mif` file to the Custom MIF Files to Be Read panel.
4. Perform the preceding steps to Add Subscribers and a Distribute Profile to run the PTF scan on the selected AS/400 machines.

Note

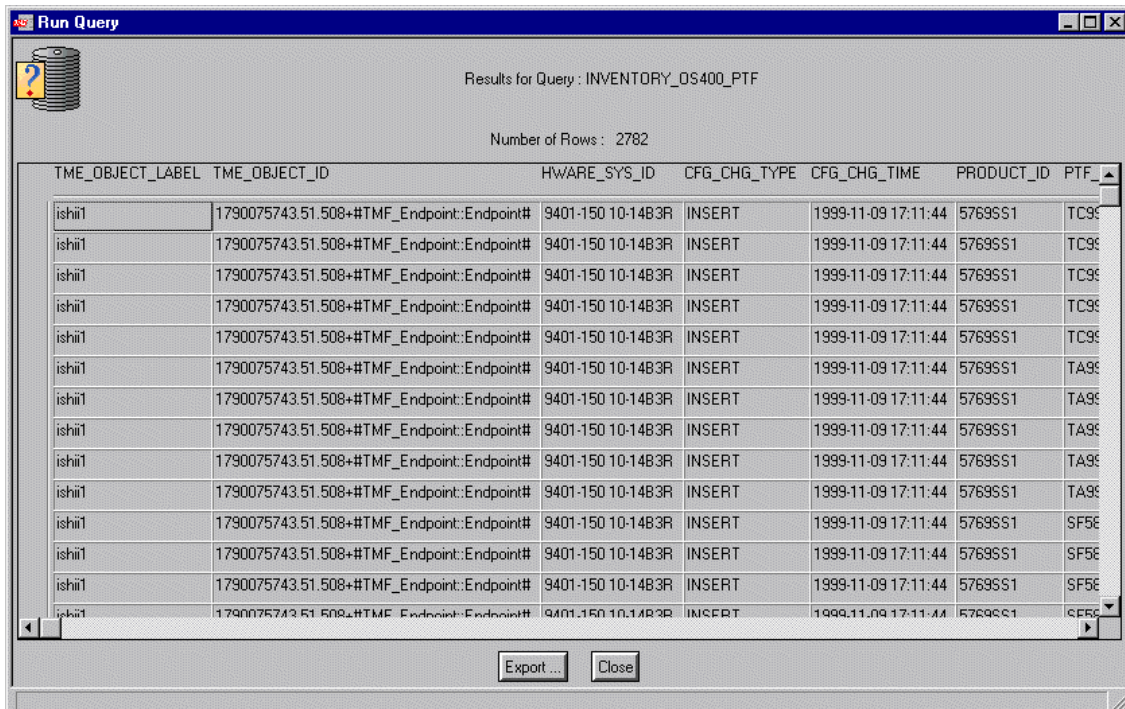
In this example, we deselected the default hardware and software scan run. You do not have to do this, but this is recommended.

8.3.2.5 Run PTF query

Tivoli provides the INVENTORY_OS400_PTF query to retrieve information from PTF_VIEW. To set up this query, you must install the inventory_as400_queries.sh script.

To query the OS/400 PTF information stored in an Inventory database, perform the following steps:

1. Select **INVENTORY_QUERIES** from the policy region.
2. Select **INVENTORY_OS400_PTF** from INVENTORY_QUERIES.
3. From the INVENTORY_OS400_PTF dialog box, select **Run Query**. You will see the OS/400 PTFs query results listed in Figure 269.



Run Query

Results for Query : INVENTORY_OS400_PTF

Number of Rows : 2782

TME_OBJECT_LABEL	TME_OBJECT_ID	HWARE_SYS_ID	CFG_CHG_TYPE	CFG_CHG_TIME	PRODUCT_ID	PTF
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TC9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TC9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TC9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TC9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TA9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TA9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TA9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TA9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	TA9S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	SF5S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	SF5S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	SF5S
ishii1	1790075743.51.508+#TMF_Endpoint::Endpoint#	9401-150 10-14B3R	INSERT	1999-11-09 17:11:44	5769SS1	SF5S

Export ... Close

Figure 269. OS/400 PTFs query results

8.3.2.6 Other queries for AS/400 Endpoints

You can query other inventory information about AS/400 hardware by performing the same steps. By default, the Inventory query libraries, listed in Figure 270, are available, and you can select one of them and run query as needed.

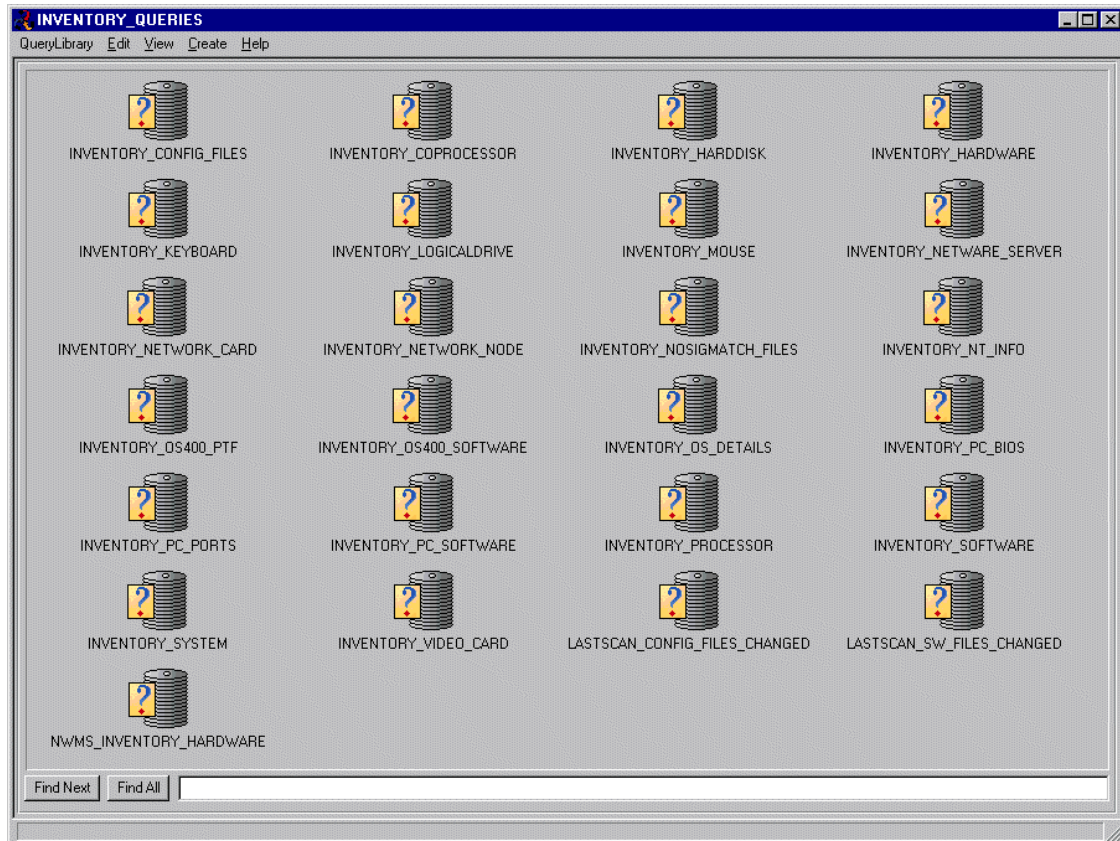


Figure 270. Tivoli Inventory queries

Chapter 9. AS/400 management scenarios

This chapter will present AS/400 scenarios in which Tivoli for the AS/400 can be utilized. Each scenario will contain example configurations and will cover the Tivoli products already introduced in this book. The scenarios will cover some of the more important tasks concerning AS/400 systems management and should serve as a good starting point from which to get Tivoli working in your AS/400 environment, no matter how many AS/400s it may consist of.

9.1 AS/400 systems environment

Being such a versatile machine, it is not easy to generalize about the AS/400 from a systems environment perspective. The AS/400 has many uses, mainly in the financial and manufacturing sectors. Each environment will have specific management needs, and it depends mainly on the types of applications running on the AS/400.

Some AS/400 environments may have thousands of interactive users maintaining and querying vast amounts of data. This is typical of a financial organization running one of the many third-party financial packages available for the AS/400. Performance, availability, security, and backup and recovery are all critical to the daily operations of such a setup. Continuous monitoring of system resources, background jobs, communications lines, user access, and disk storage are all part of keeping the system operating efficiently and minimizing the chance of downtime due to hardware or software failure.

In fact, the same basic principles of AS/400 management apply to most if not all AS/400 environments. Whether you operate in a financial, manufacturing, or any other environment, your AS/400 is likely to be a critical part of your business operations and must be available as required.

The scenarios in this chapter will be based on two AS/400s in an organization. Both AS/400s support the financial operations of the company using standard third-party applications. Through examples, we will demonstrate how Tivoli for the AS/400 can be configured to manage these AS/400s. These examples can be extended to suit any AS/400 configuration and will act as a basis on which to incorporate your AS/400 network into the Tivoli managed environment.

9.2 Management scenario

The examples involving the scenarios presented below will be discussed under the following topics using the products listed in Table 26.

Table 26. Tivoli products in our scenarios

Management examples	Tivoli products
Availability Management	Tivoli Distributed Monitoring
	Tivoli Enterprise Console
Security Management	Tivoli User Administration
	Tivoli Security Management
Deployment Management	Tivoli Software Distribution
	Tivoli Inventory

To demonstrate the examples in this chapter, we will use two AS/400s acting as Endpoints in the TMR as shown in Table 27.

Table 27. AS/400 test environment

Machine name	Model	OS/400 version	Tivoli products installed
ishii1	150	V4R4	Tivoli Endpoints for AS/400 & TEC Adapters
ishii2	170	V4R4	Tivoli Endpoints for AS/400 & TEC Adapters

As shown in Figure 271 on page 439, the AS/400 Endpoints communicate with an Endpoint Gateway, itso2, which in turn is connected to the TMR server, itso1.

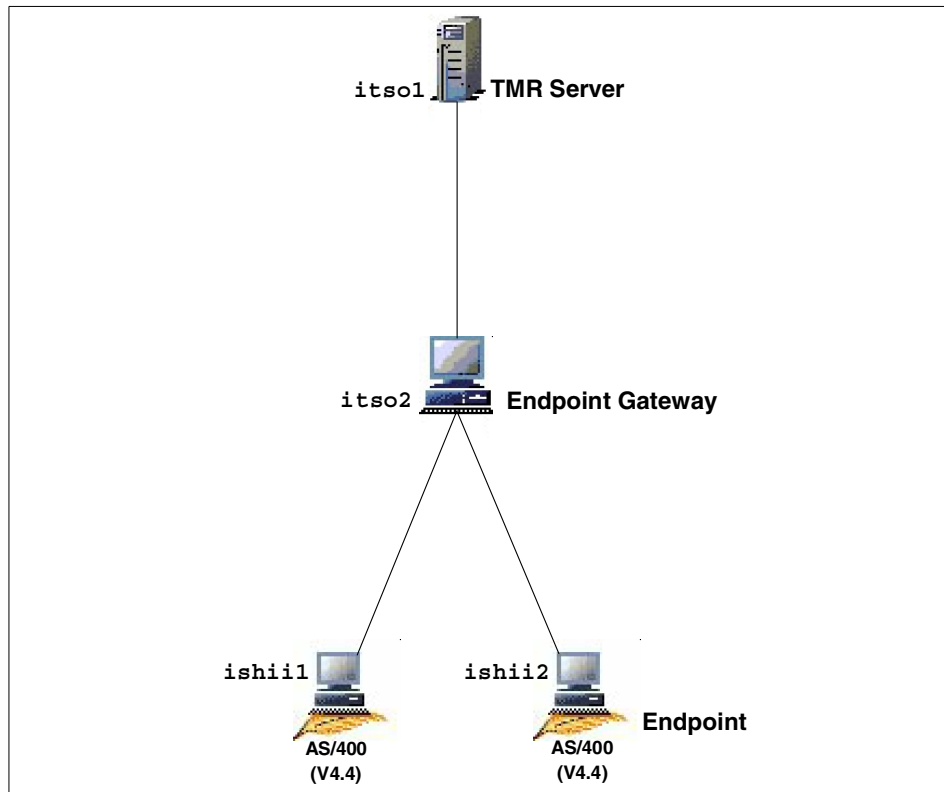


Figure 271. Our Tivoli management environment

Our environment only consists of two AS/400 Endpoints; however, Tivoli is capable of supporting all AS/400 Endpoints on your network regardless of size and complexity.

9.2.1 Financial environment scenario

In our scenario, we will be using the ishii1 and ishii2 AS/400s. They have a full suite of products installed to enable application development and maintenance. The following outlines some of the assumptions we make about this environment and the constraints that must be adhered to:

- The AS/400s must be available for use during normal business hours. They play a critical role in the day-to-day running of this organization. Poor performance and interruptions in availability must be avoided.
- The normal times of operation are 7:00 a.m. to 11:00 p.m. Monday through Saturday. Each weekday, from 11:00 p.m. to 7:00 a.m., backups and system maintenance are performed. On Sunday, a full system backup and

IPL take place. The machines are available for use again by 7:00 a.m. on Monday morning.

- The systems have many user profiles and contain a lot of confidential data; therefore, security and user administration are important roles in the system management of the AS/400s.
- The third-party financial software running on the AS/400s has a number of background jobs running in various subsystems. These jobs are a crucial part of the application and must be monitored for problems at all times.
- A large amount of data is input and is generated each day on this system. Data storage resources have to be monitored to ensure that existing data integrity is maintained and to allow for planning storage needs in the future.
- To support data exchange, there is an SNA link between the two AS/400s. This connection must be monitored on a regular basis to ensure that the link is functioning and stable.

In our examples, we will show how this environment can be managed using the Tivoli products. We will also see that many of the functions explored can be used across many AS/400s in an environment, thus, helping to avoid any repetition of effort when managing your environment.

Both ishii1 and ishii2 AS/400s will be used during the course of the examples. We will show how to configure aspects of Tivoli for AS/400 Endpoints and the TEC Adapters. The examples will detail management solutions using Tivoli based on the requirements and constraints presented by the scenarios.

9.3 AS/400 availability management

The examples in this section will use Tivoli Distributed Monitoring for the AS/400 and the TEC alert and message adapters for AS/400. For installation details, refer to Chapter 7, “AS/400 availability management” on page 217. We will explore Distributed Monitoring and TEC under the following topics:

- Subsystem management
- Alerts and messages
- Managing system resources

9.3.1 Subsystem management

Subsystems manage resources on your AS/400. They control the work flow and resources for all interactive and batch jobs on the machine. If a particular subsystem is not running, no jobs allocated to that subsystem can execute. If

a subsystem suddenly ends, all the jobs running in it also end. Therefore, subsystems are a vital component of AS/400 system availability.

One of the most important subsystems on the AS/400 is QINTER. QINTER runs all the users' interactive jobs. If you use telnet to access the AS/400, your job will run in QINTER. If QINTER becomes inactive, all interactive sessions end and, thus, nobody can use the AS/400. The following example will show how to monitor QINTER using Distributed Monitoring.

Note

If your AS/400 system is configured to use the QBASE subsystem, you may substitute this subsystem for QINTER.

If you create a Distributed Monitoring profile with too many monitors, it may affect the schedule of each monitor, and they may not run when expected. Refer to Chapter 7, "AS/400 availability management" on page 217, for more information.

9.3.1.1 Example 1: Monitoring the QINTER subsystem

In this example, we will use Distributed Monitoring to monitor the QINTER subsystem every two minutes. We will set the monitor to send an e-mail to the system operator and a critical event to TEC. The monitor will also attempt to restart the subsystem using the Run Program option in the profile definition. The steps for creating this monitor are outlined as follows:

1. Open your Distributed Monitoring profile manager, and create a new profile using the **Create** option on the **Profile** menu.
2. Give the profile a name and select its type as **SentryProfile** as shown in Figure 272 on page 442. Click **Create & Close** when you are finished.

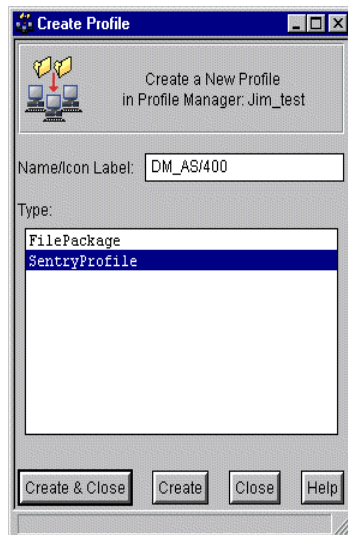


Figure 272. Creating a Distributed Monitoring profile

- Now, double click on the new profile icon. This will display the Distributed Monitoring Profile Properties screen. This screen lists all your monitor definitions for this particular profile as shown in Figure 273.

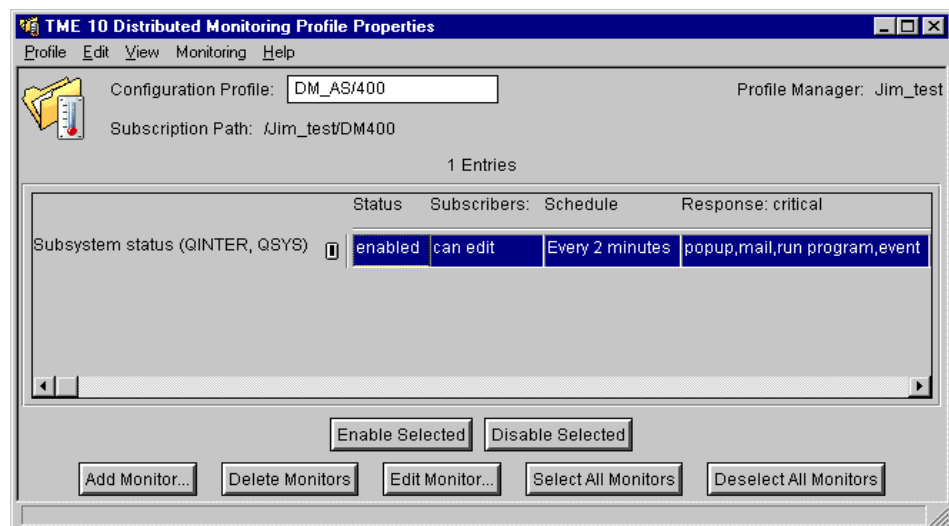


Figure 273. Distributed Monitoring Properties

4. Click on the **Add Monitor** button to add a new monitor. This option will open the Add Monitor screen shown in Figure 274.
5. Notice the list of Monitoring Collections on the screen. We want to monitor a subsystem; so, highlight the **OS/400 Subsystem** monitoring collection. In the *Monitoring Sources* section, highlight the **Subsystem Status** item. Now, in the *Monitoring Argument* section, enter the subsystem name and the library in which the subsystem description resides. In this case, we want to monitor the QINTER subsystem in the QSYS library; so, enter these values in the provided fields. Click the **Add Empty** button when you are finished.

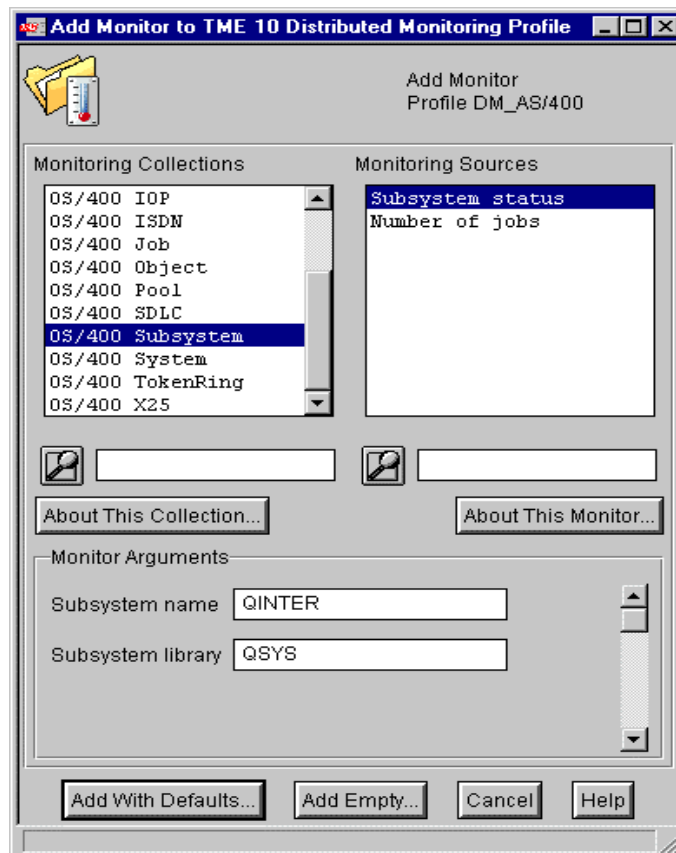


Figure 274. Choosing a monitor collection and source

Note

In this process, you must enter the subsystem name and library in uppercase letters.

- Next, the Edit Monitor screen, shown in Figure 275, is displayed. Here, we can tailor the monitor to suit our environment. First, we need to select a response level and a trigger.

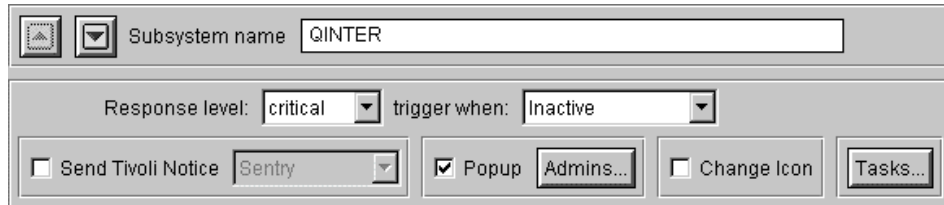


Figure 275. Selecting the monitor severity level

In the event of the QINTER subsystem ending, we want to produce a response level of critical. To do this, choose **Critical** from the *Response level* list and **Inactive** from the *trigger when* list.

- When this event occurs we want to send a Tivoli pop-up message to the system operators so select the **Popup** checkbox. Next, click the **Admins.** button and choose the recipients of the message as shown in Figure 276.

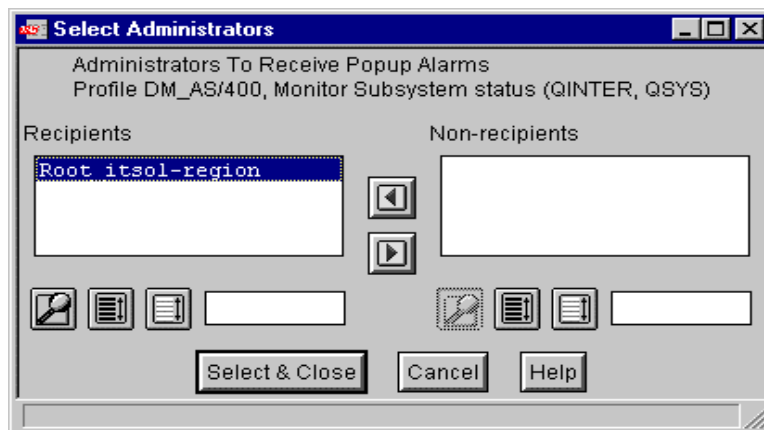


Figure 276. Selecting popup message recipients

Note

A popup message will only appear to those users who have the Tivoli desktop open at the time of the event.

8. We also want Tivoli to send an e-mail to the systems operator when QINTER ends. Select the **Send email to** checkbox, and enter the e-mail address in the field provided.
9. Next, we want configure the monitor to restart QINTER when it ends. To do this, we select the **Run Program** checkbox as shown in Figure 277.

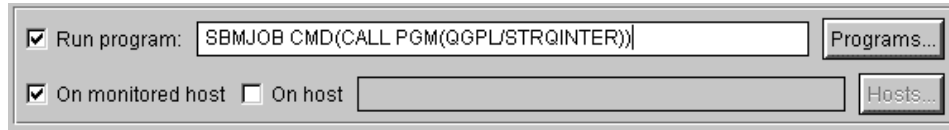


Figure 277. Using the Run Program monitor option

The program will be run on the AS/400; so, we must choose the **On monitored host** checkbox. In this example, we will call the STRQINTER program as follows:

```
SBMJOB CMD(CALL PGM(QGPL/STRQINTER) ) JOBQ(QBATCH)
```

Note

When using the Run Program option no syntax checking is performed by Distributed Monitoring. You must also use uppercase letters.

10. To send this event to our TEC server, click on the **Send Enterprise Console event** checkbox. Choose a severity level of **Fatal** from the list as shown in Figure 278.

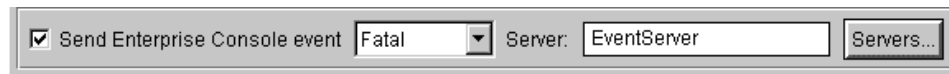


Figure 278. Selecting the TEC server

Next, click on the **Servers...** button to select the TEC server to which we want to send the event.

Note

TEC needs to be configured to receive events from Distributed Monitoring. See the *Tivoli Distributed Monitoring User's Guide*, page 3.7 for details.

11. When you have finished steps 8 through 13, the monitor should look like the screen in Figure 279.

The screenshot shows the 'Edit Monitor' dialog box with the following configuration:

- Profile: DM_AS/400
- Monitor: Subsystem status
- Subsystem name: QINTER
- Response level: critical
- trigger when: Inactive
- ☐ Send Tivoli Notice (Sentry)
- ☒ Popup (Admins...)
- ☐ Change Icon (Tasks...)
- ☒ Send E-mail to: colljim@je.ibm.com
- ☐ Log to file: (Files...)
- ☐ On monitored host
- ☐ On host: (Hosts...)
- ☒ Run program: SBMJOB CMD(CALL PGM(QGPL/STRQINTER)) (Programs...)
- ☒ On monitored host
- ☐ On host: (Hosts...)
- ☒ Send Enterprise Console event: Fatal
- Server: EventServer (Servers...)
- Buttons at the bottom: Set Message Styles..., Set Distribution Actions..., Set Monitoring Schedule..., Change & Close, Cancel, Help

Figure 279. Completed monitor definition

12. Next, we need to set the execution schedule for the monitor. Click on the **Set Monitoring Schedule** button to display the Monitoring Schedule window.

If required, you may change the initial monitor start time, but, in this example, we will use the default. On this screen, we will also set the monitor frequency to run every two minutes as shown in Figure 280.

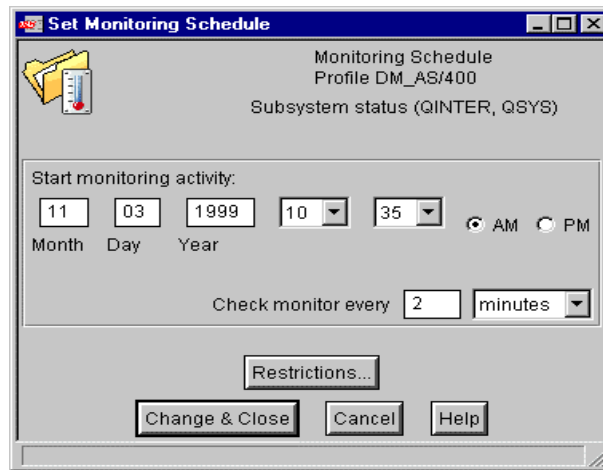


Figure 280. Setting monitoring schedule

13. We do not want our monitor to run during the daily or weekly backup jobs because QINTER will be ended; we don't want it restarting automatically. In order to achieve this, we must set monitor restrictions. Click on the **Restrictions...** button.
14. The Monitoring Schedule Restrictions screen allows us to choose the times during which we want the monitor to run. We want to set our own customized hours of operation from 7:00 a.m. to 11:00 p.m., Monday through Saturday. To do this, select the **Custom Hours** and **Custom days** checkbox, as shown in Figure 281.

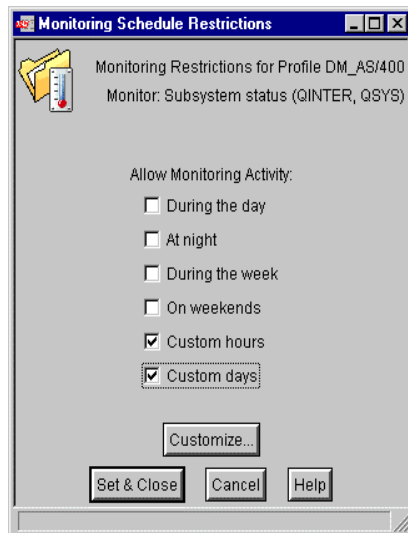


Figure 281. Choosing custom times and days

15. Click on the **Customize** button to open the Customize Monitoring Schedule Restrictions screen. Set the Custom Hours field to run the monitor from 7:00 a.m. through 11:00 p.m. and the Custom Days field from Monday through Saturday each week as shown in Figure 282 on page 449. The other default settings will be ignored. Click on **Set & Close** to save the schedule.

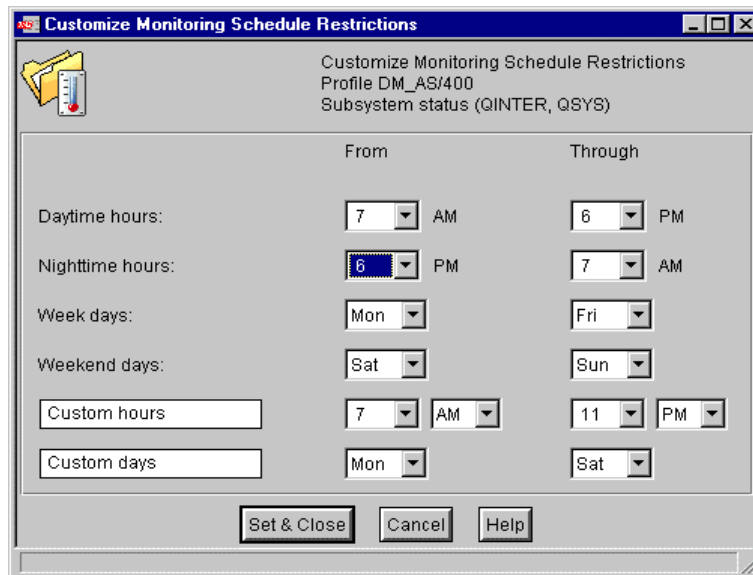


Figure 282. Customizing the QINTER monitor schedule

16. Back on the Monitoring Schedule Restrictions screen, click on the **Set & Close** button.
17. On the Set Monitoring Schedule screen, click the **Change & Close** button.
18. You should now be back in the Edit Monitor screen; click on the **Change & Close** button.
19. The monitor has now been created and will appear in the Monitoring Profiles Properties screen as shown in Figure 283 on page 450. To save the new monitor, select the **Save** option from the Profile menu.

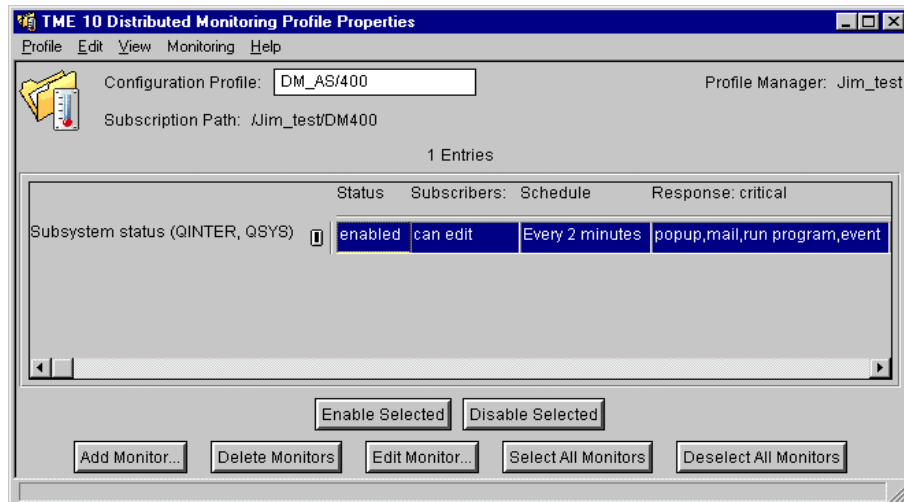


Figure 283. QINTER monitor in the DM_AS/400 profile properties

20. To distribute this new profile to both of our AS/400s, select the **Distribute** option from the Profile menu. The Distribute Profile screen will be displayed as shown in Figure 284.

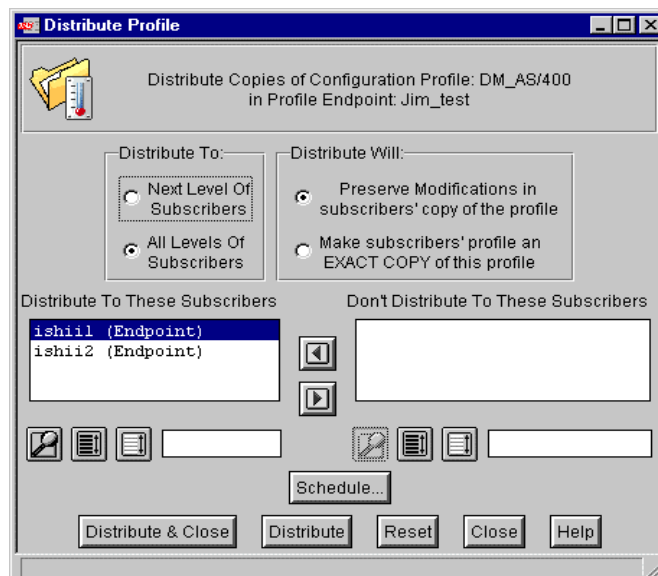


Figure 284. Distributing the DM_AS/400 profile

Select the subscribers to which you want to distribute, in this case, ishii1 and ishii2, and then click **Distribute & Close**.

Within a couple of minutes, the monitors should have distributed and be running on the AS4/00s. In the event of QINTER ending, a pop-up box will be displayed on your Tivoli desktop as shown in Figure 285.

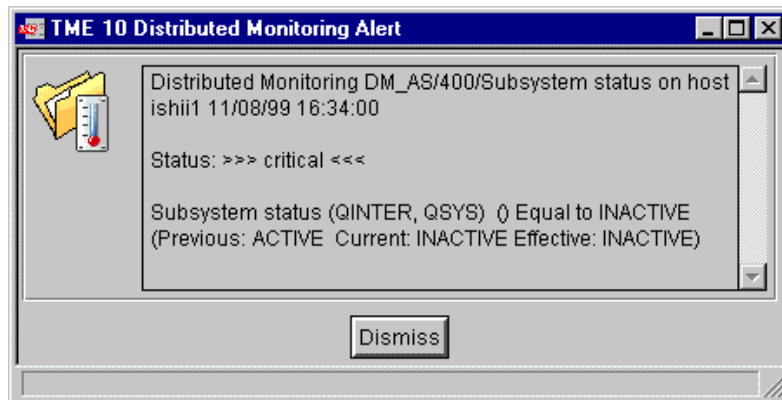


Figure 285. Tivoli Desktop popup message

An e-mail message will also be sent to the system operator displaying the same details as the popup message. TEC will also receive a Fatal event as shown in Figure 286.

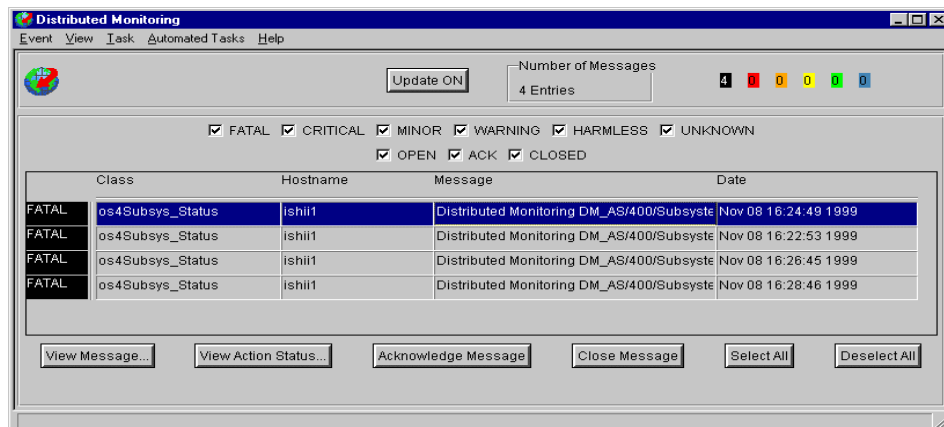


Figure 286. TEC events generated by DM_AS/400 QINTER subsystem monitor

Table 28 shows a list of other possible subsystem monitors and timing frequencies that may be useful.

Table 28. Suggested Subsystem Monitors and actions

AS/400 subsystem	Monitor frequency	Action
QBATCH	Every 5 minutes	Restart subsystem and any background jobs that use it
QSERVER	Every 10 minutes	Restart subsystem and run the STRHOSTSVR command
QSNADS	Every 15 minutes	Restart subsystem
QSPL	Every 20 minutes	Restart Subsystem and any print writers that ended

9.3.2 Alerts and messages

With Tivoli Enterprise Console (TEC), all your AS/400-generated alerts and messages from your network can be logged in one central console. TEC logs events from AS/400 alerts and message queues using the TEC Alert Adapters and the TEC Message Adapters.

By default, the message adapter uses the QSYSOPR message queue. However, TEC can be configured to log messages from any application that generates messages in a message queue. For both alert and message adapters, you can filter out unimportant events so that only the events that you want to see and act upon will appear.

The first of the next three examples will show how to create a message adapter to monitor a non-system or application message queue. The second example will show how to create an automated response for an event logged to the TEC server. In Section 9.3.2.3, “Example 4: filtering TEC alerts” on page 464, we will show how we can configure the adapter filters to filter out unwanted events.

9.3.2.1 Example 2: customizing TEC message adapters

On ishii1 and ishii2, there are a large number of interactive and batch programs running at any time. These programs generate messages detailing normal operating conditions and any errors that may occur. The main message queue for our application is FIN_MSGQ in the FIN_APPS library.

Note

TEC needs to be configured to receive events from Distributed Monitoring. See the *Tivoli Distributed Monitoring Use's Guide*, page 3.7 for details.

This example will monitor for messages in FIN_MSGQ that are generated by the program, FIN_UPDATE, which runs as a background job in QBATCH. Every 15 seconds, the program updates the various financial ledgers with new and modified data collected from the numerous financial modules. If the program encounters an error, it posts a message to the FIN_MSGQ message queue.

The first step in customizing the Tivoli Message Adapter is to configure the adapter to log events from this queue. Perform the following steps to create and start the adapter:

1. Create a library on ishii1 to store the configuration of the customized adapter. In this case, we will create a library called FIN_ADPTRS using the following command:

```
CRTLIB LIB (FIN_ADPTRS)
```

2. Copy the default message adapter configuration file, QAYMACFG, from the QUSRSYS library to our new FIN_ADPTRS library as follows:

```
CPYF FROMFILE (QUSRSYS/QAYMACFG) TOFILE (FIN_ADPTRS/QAYMACFG)
```

3. Edit the configuration file using the AS/400's Source Entry Utility (SEU):

```
STRSEU SRCFILE (FIN_ADPTRS/QAYMACFG) SRCMBR (MSGCFG)
```

Change the MsgQueue parameter to the message queue you wish to use; in this case, it is FIN_APPS/FIN_MSGQ. Figure 287 on page 454 shows the message adapter configuration file.

```

Columns . . . : 1 71          Edit          FIN_ADPTRS/QAYMACFG
SEU==>
FMT ** ..... 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6 ..... 7
0018.00 # 5733-TA1 (C) Copyright Tivoli CORP. 1997, 1997
0019.00 # All rights reserved.
0020.00 # US Government Users Restricted Rights -
0021.00 # Use, duplication or disclosure restricted
0022.00 # by GSA ADP Schedule Contract with Tivoli Corp.
0023.00 # Licensed Material - Property of Tivoli
0024.00 #
0025.00 #-----
0026.00 ServerLocation=itsol.dev.tivoli.com
0027.00 ServerPort=0
0028.00 ConnectionMode=connection_oriented
0029.00 AdapterType=MSGQ
0030.00 AdapterCdsFile=/QSYS.LIB/QUSRSYS.LIB/QAYMACDS.FILE/MSGCDS.MBR
0031.00 MsgQueue=FIN_APPS/FIN_MSGQ
0032.00 PollInterval=20
0033.00 JobDescription=QGPL/QDFTJOB0
0034.00 #

F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle
F16=Repeat find   F17=Repeat change   F24=More keys

MR a                                           02/009

```

Figure 287. Message Adapter configuration file - Defining a message queue

- Now that we have configured our custom message adapter, we need to start it using the following command, specifying the adapter job name and the configuration file parameters:

```

STRTECADP EVTADP (FINMSGADP)
CFGFILE ('/qsys.lib/fin_adptrs.lib/qaymacfg.file/msgcfg.mbr')

```

The adapter will run in the QSYSWRK subsystem as shown in Figure 288 on page 455.

Note

To save time and help prevent errors, it is recommended that you create a small CL program containing the `STRTECADP` command to start your TEC adapters.

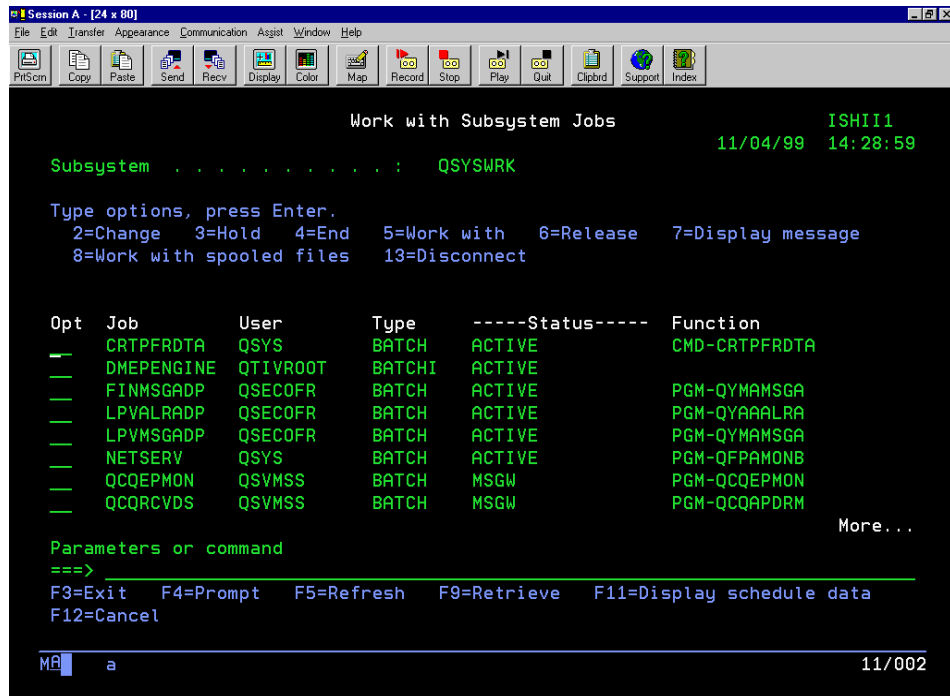


Figure 288. Message Adapter, FINMSGADP, running in QSYSWRK

Once the message adapter is started, it will send events to the TEC server when messages are generated by our applications. The events can be viewed from the TEC Console on your Tivoli Desktop. The events will appear in the AS/400 Messages Console as shown in Figure 289 on page 456.

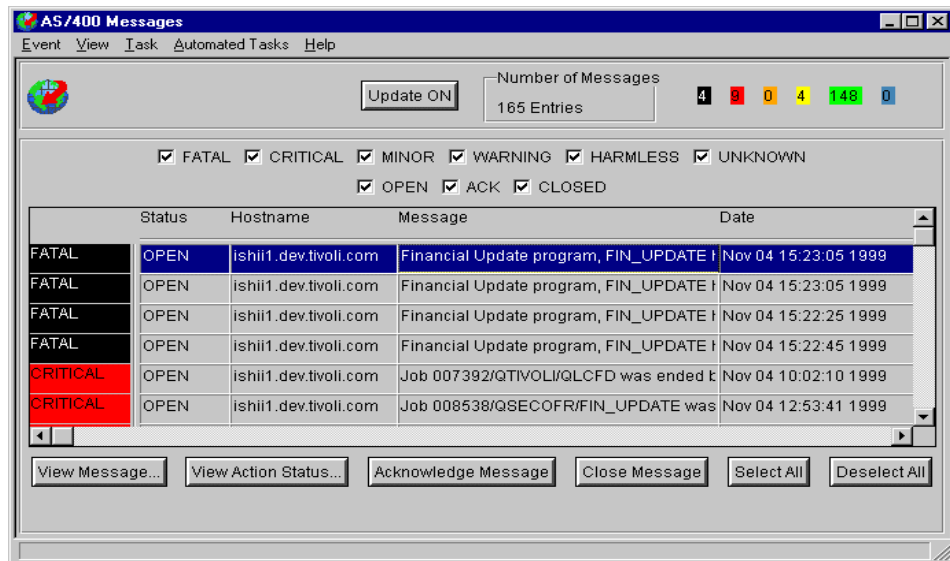


Figure 289. Viewing AS/400 message adapter events in TEC

9.3.2.2 Example 3: creating an automated response in TEC

Continuing with the last example, we will now create an automated response for the messages logged to TEC by our customized message adapter.

First, we need to define a response action using the AS/400 task library. Then, we associate the response with an event in TEC, namely, the FIN_UPDATE message, FIN0001, which tells us that the update program has ended.

The following list outlines the steps involved in setting up the automatic response:

1. First, we need to create a task to run when the event is detected by the TEC Console. Open the AS/400 Task Library shown in Figure 290 on page 457.

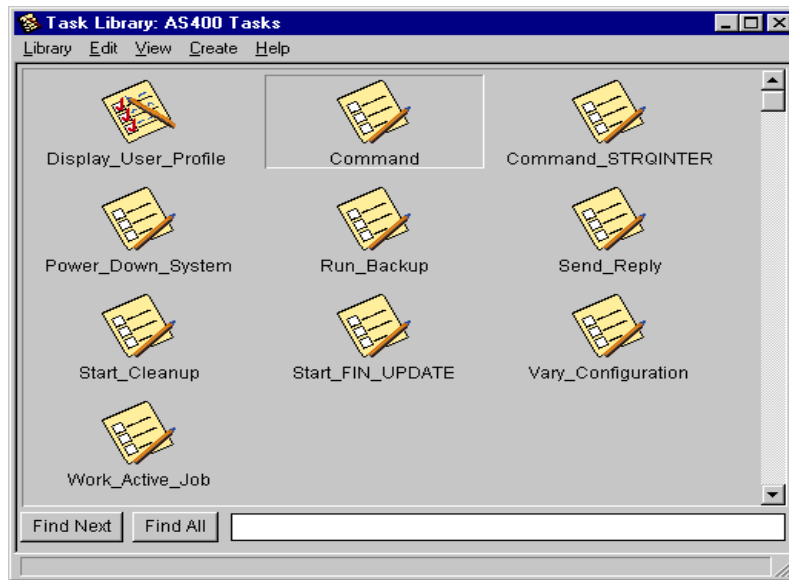


Figure 290. AS/400 Task Library

Double click on the **Command** task to open it.

2. On the Execute Task screen shown in Figure 291 on page 458, select the endpoint on which the task will be run. In this case, we will set the task to run on ishii1; however, this can be set up to run on all AS/400s shown in the Endpoints list.

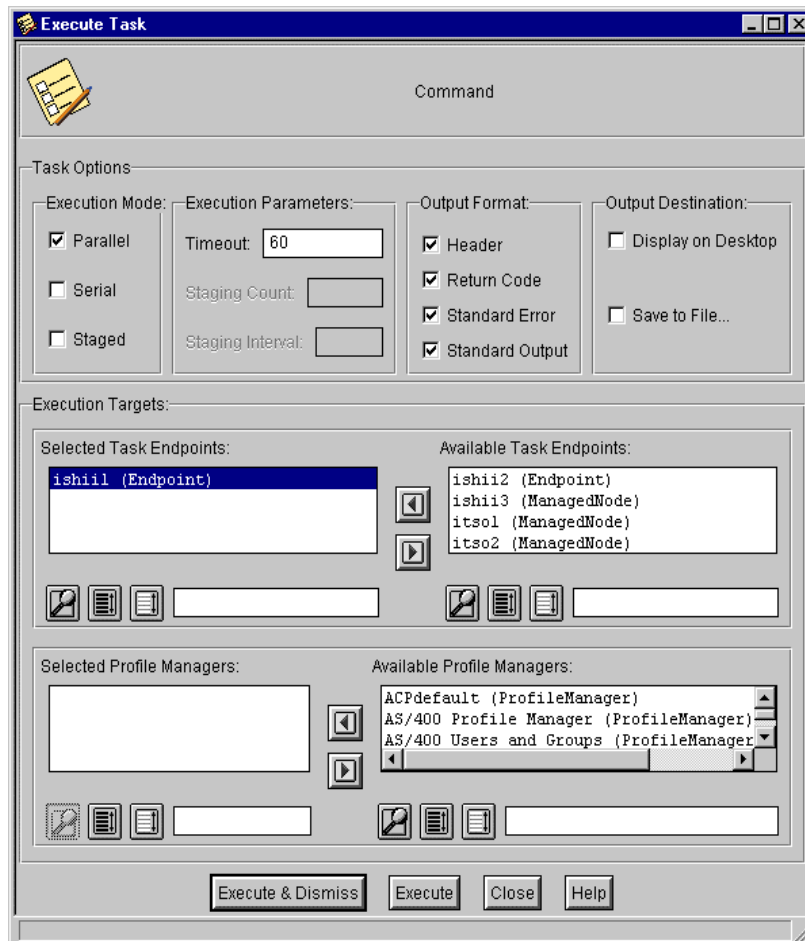


Figure 291. Selecting endpoint on the Execute Task screen

When completed, click on the **Execute & Dismiss** button.

3. The next screen will prompt you for an AS/400 command. For our example, we want to restart the FIN_UPDATE job in the QBATCH subsystem; so, enter the following command:

```
SBMJOB CMD(CALL PGM(FIN_UPDATE)) JOB(FIN_UPDATE)
```

When you have entered the command, click the **Save** button.

4. On the Save screen, we must enter a command name and an identifier that will be used when setting up our automated TEC response. We will name both the task and Identifier Start_FIN_UPDATE as shown in Figure 292 on page 459. You may also enter a description if necessary.

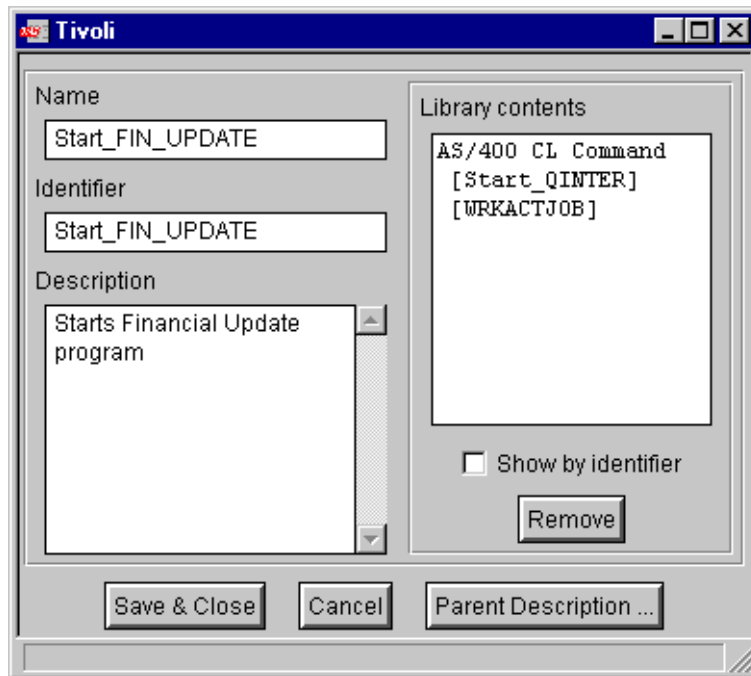


Figure 292. Saving a task in the AS/400 Task Library

Click on **Save & Close** to save the new task. Close all other Task Library screens, and return to the AS/400 Messages console in TEC.

5. In the AS/400 Messages console, select **New** from the Automated Tasks menu to display the Select Event Class screen shown in Figure 293 on page 460.

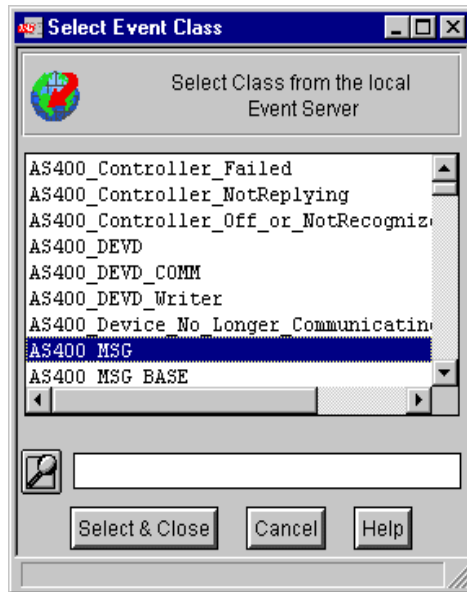


Figure 293. Select Event Class screen

Select the **AS400_MSG** class from the list, and click **Select & Close**.

6. On the Summary of Automated Tasks screen, we have to select the criteria that we will use to determine when our automated response will run. First, click on the **Edit Criteria** button to display the Setup Automatic Task Execution screen shown in Figure 294 on page 461.

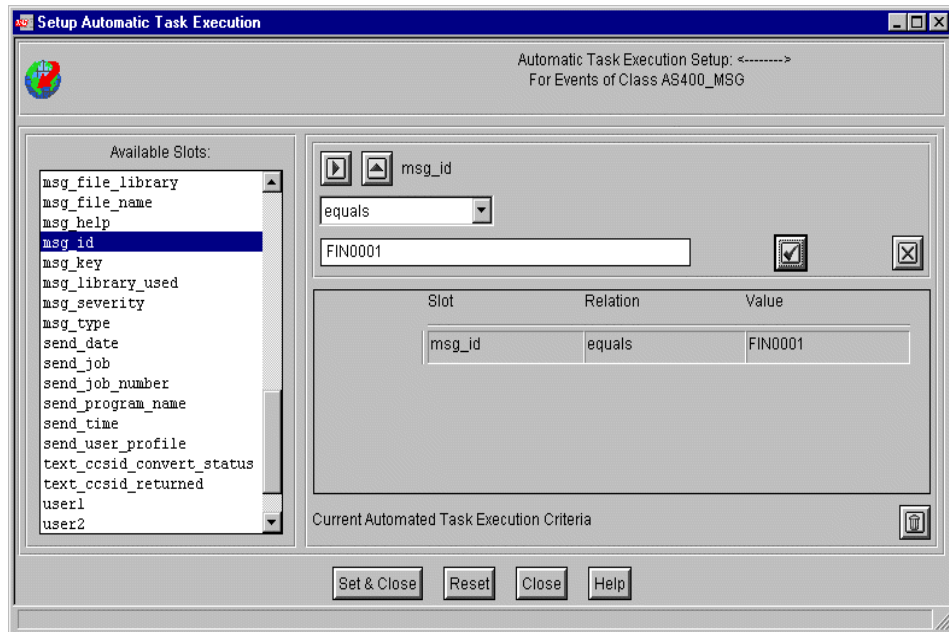


Figure 294. Editing Automatic Task execution criteria

Choose **msg_id** from the Available Slots list. This is the part of the TEC event that contains the AS/400 Message ID field. We want to trigger an automated response when the message ID equals FIN0001; so, enter this in the provided field. Click the **Select & Close** button when completed.

- Now, we have to associate our automated response with the event. Click on the **Add Task...** button to display the screen shown in Figure 295 on page 462.

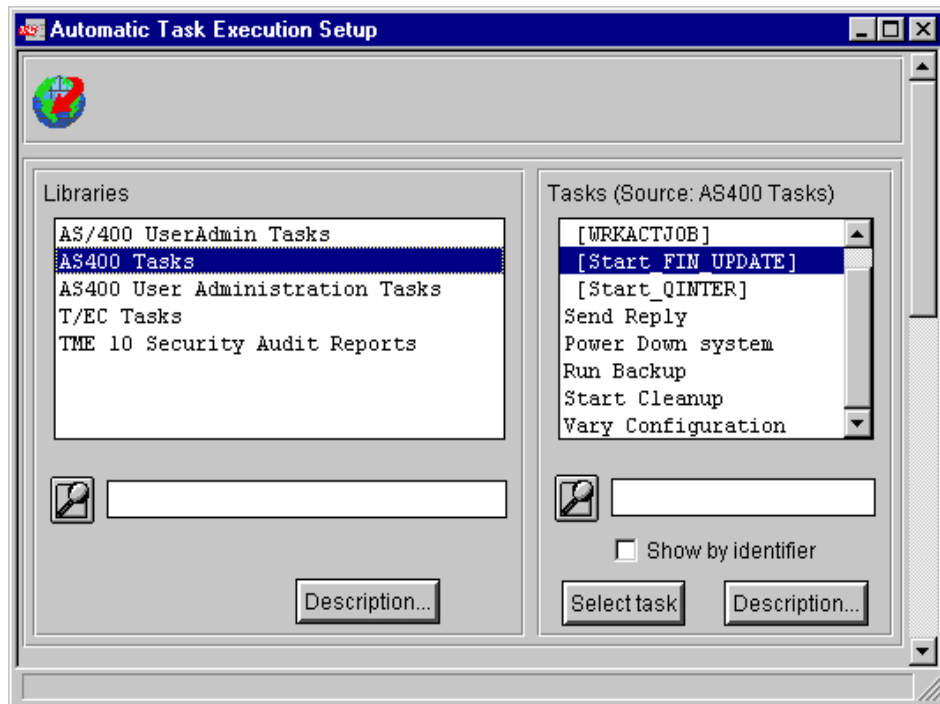


Figure 295. Select an automated task

Select **AS/400 Tasks** from the list of libraries and the defined task, **[Start_FIN_UPDATE]**, from the Tasks list. Click on the **Select Task** button when done.

8. Next, select **Execute on Select Managed Nodes** at the bottom of the same screen as shown in Figure 296.



Figure 296. Select Managed Nodes for automated task

Click on **Add & Close**. You will be shown a list of available managed nodes as shown in Figure 297 on page 463.

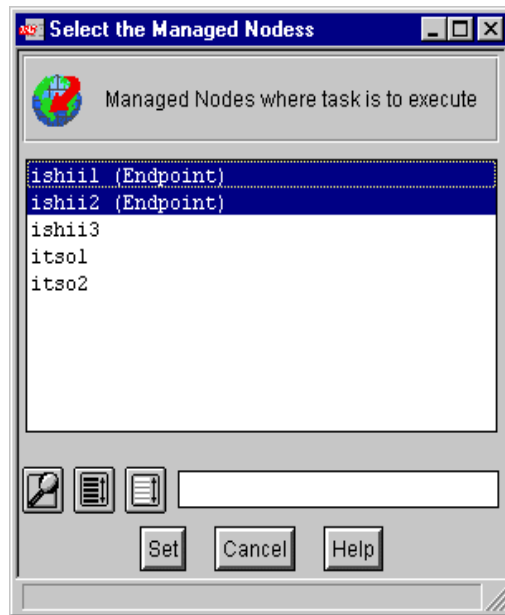


Figure 297. Select nodes for automated task execution

In this case, select **ishii1** and **ishii2** from the list, hold down the **Ctrl** key to highlight each Endpoint, and click **Set** when completed.

9. The Summary Of Automated Tasks screen should now look like that in Figure 298 on page 464.

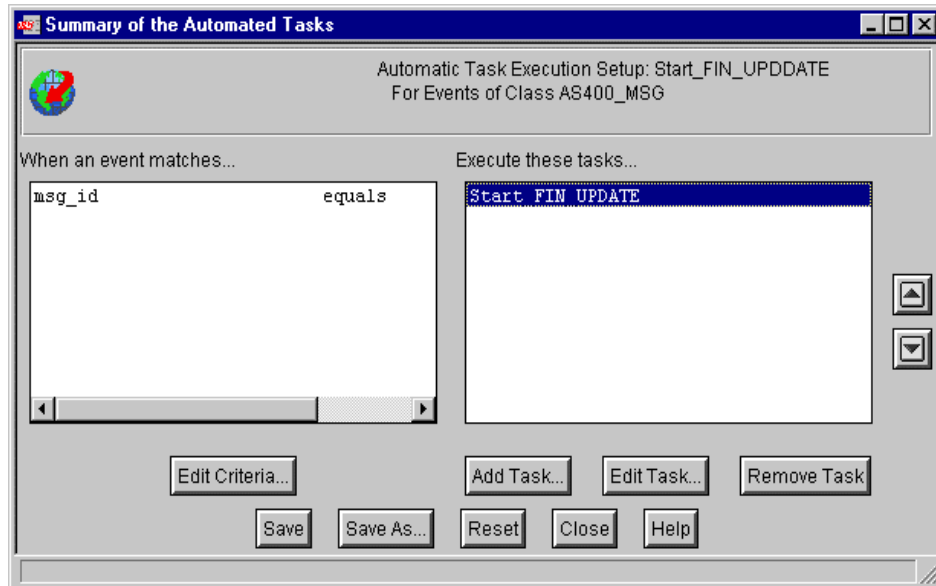


Figure 298. Completed automated task

Click on the **Save As...** button to save the automated response. When completed, click the **Close** button and return to the AS/400 Messages console.

Now that our Automated Response has been configured, TEC will perform the automated task, that is, it will restart the FIN_UPDATE job when it receives an event with an AS/400 Message ID of FIN0001. These automated tasks can be configured for any event on the AS/400 that posts a message to a Message Adapter or an Alert Adapter monitored by the TEC server.

9.3.2.3 Example 4: filtering TEC alerts

In this example, we will create a filter for the TEC Alert Adapters on ishii1 and ishii2. There is an AnyNet/400 link between ishii1 and ishii2 emulating an SNA link over TCP/IP between the two AS/400s. This can generate a large number of informational messages that we do not want logged in TEC. We only want SNA alerts with a severity of 50 or higher to arrive at the TEC Server Console.

Note

In order for alert events to get to TEC Alert Logging must be enabled on the AS/400. To do this run the following command:

```
CHGNETA ALRLOGSTS(*ALL)
```

The filter settings are stored in the QAYAACFG file in our alert adapter library, FIN_ADPTRS on ishii1.

To configure the filters, complete the following steps:

1. End any active Alert Adapters using the `ENDTECADP` command as follows:

```
ENDTECADP EVTADP(*ALL) OPTION(*IMMED)
```

2. Open the QAYAACFG file in the alert adapters library, FIN_ADPTRS, using the `STRSEU` command as follows:

```
STRSEU SRCFILE(FIN_ADPTRS/QAYAACFG) SRCMBR(ALRCFG)
```

3. The filter settings are stored at the end of the file. To create a new filter, use the following syntax:

```
Filter:Class=AS400_ALERT;severity=sev_level
```

where `sev_level` is the severity level that you want to exclude from TEC.

In this example, we want to exclude messages with severities of **0**, **10**, **20**, **30**, and **40**. To do this, add the following filter definitions to the end of the configuration file:

```
Filter:Class=AS400_ALERT;severity=0
Filter:Class=AS400_ALERT;severity=10
Filter:Class=AS400_ALERT;severity=20
Filter:Class=AS400_ALERT;severity=30
Filter:Class=AS400_ALERT;severity=40
```

The Alert Adapter configuration file is shown in Figure 299 on page 466.

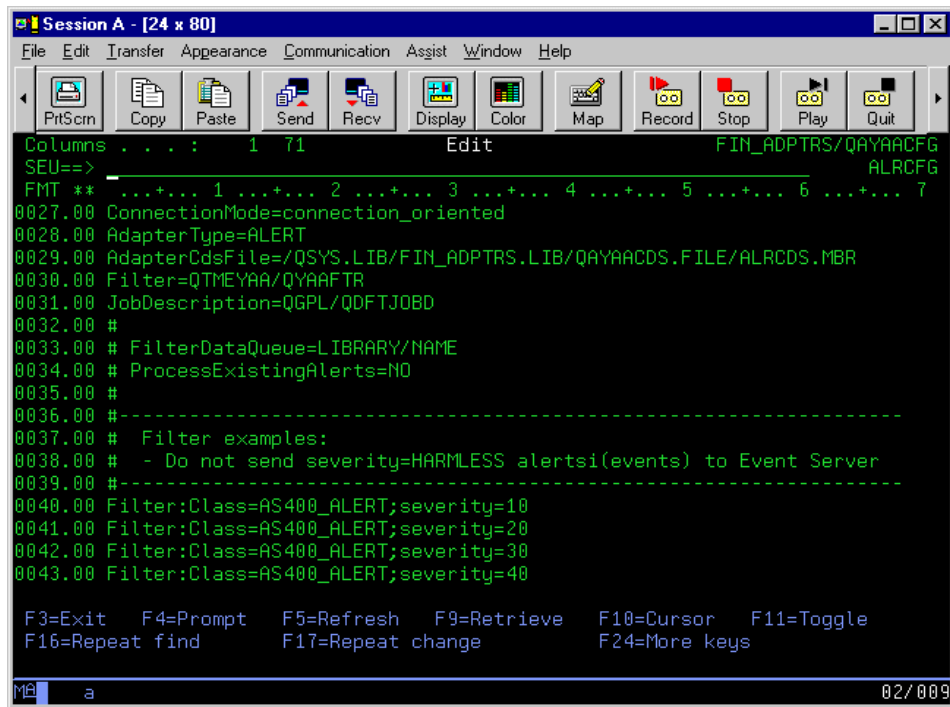


Figure 299. Configuring Alert Adapter filters

When you have finished entering the filters, press **F3** to exit and save the file.

4. Now, start the adapters again using the following STRTECADP command:

```
STRTECADP EVTADP(FINALRADP) +
CFGFILE('QSYS.LIB/FIN_ADPTRS.LIB/QAYAACFG.FILE/ALRCFG.MBR)
```

The AS/400 Alerts console should now receive alert events from ishii1 with a severity of 50 or over as shown in Figure 300 on page 467.

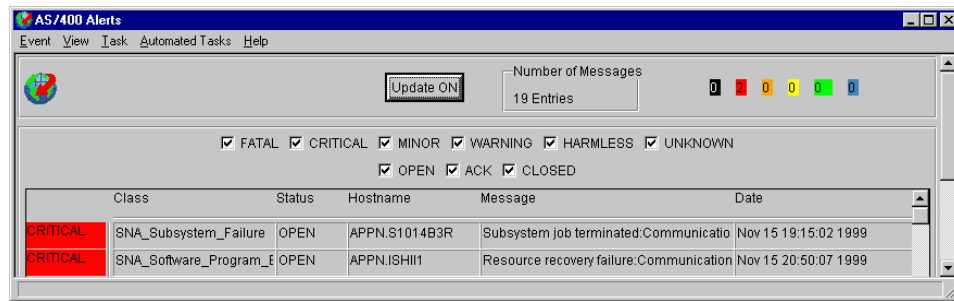


Figure 300. AS/400 Alerts console

The same procedure can be applied to the TEC Message Adapters. Add the filters to the end of the QAYMACFG file (member MSGCFG in your Message Adapter library on the AS/400).

9.3.3 Managing system resources

Tivoli Distributed Monitoring for AS/400 provides a number of monitors for various system resources. You can monitor CPU usage, communications lines, disk storage, jobs, and objects to name a few. Tivoli can monitor these resources constantly and alert you when a problem occurs. You can also specify a recovery program to attempt to resolve a problem when it occurs.

9.3.3.1 Example 5: monitoring disk storage used on AS/400

Our Financial environment requires that we monitor the status of our disk storage levels. We will use the **OS/400 Disk** monitoring collection to monitor two severities of events as follows:

- Warning** When the percentage of disk storage used exceeds 80 percent
- Critical** When the percentage of disk storage used exceeds 90 percent

When the AS/400 system storage used reaches 80 percent, an event will be sent to TEC, and an e-mail will be sent to the System Operator.

When the storage used reaches 90 percent, the monitor will send an e-mail to the system operator and a **Fatal** Event to TEC. A basic CL program will also be run to clear the QEZJOBLOG output queue to help reduce storage in the short term. This program can be distributed using Tivoli Software Distribution, see Section 9.5, “AS/400 inventory and deployment management” on page 483, for examples.

Complete the following steps to create this monitor:

1. Open a Distributed Monitoring Profile from the Profile Manager screen. We will use DM_AS400 extending our earlier QINTER monitor example in 440. Double click on the **Profile Icon** to edit it.
2. In the Profile Properties screen, select the **Add Monitor...** button. Select **OS/400 Disk** from the Monitoring Collection list and **Percent System ASP used** from the Monitoring Sources list as shown in Figure 301.

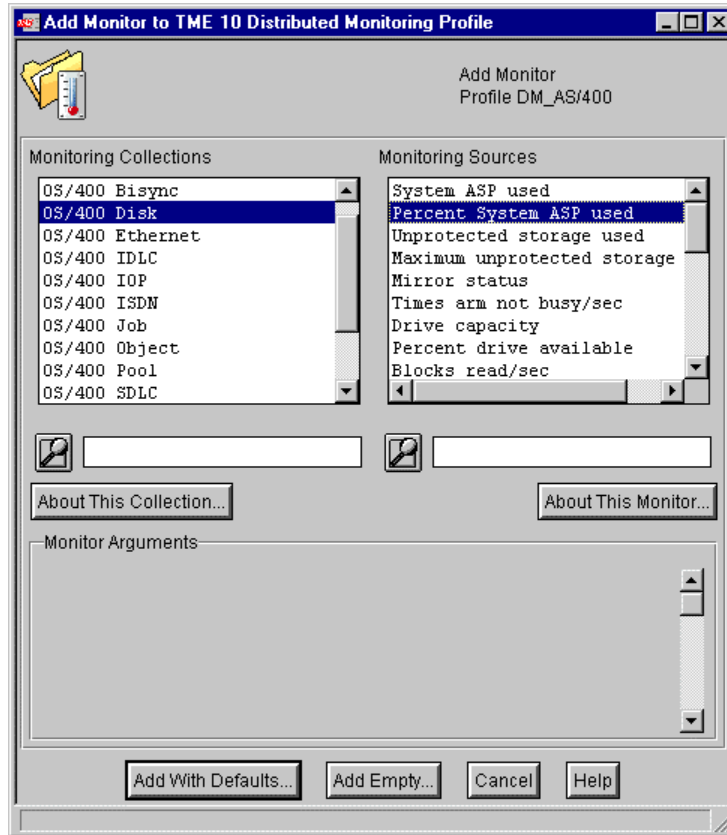


Figure 301. Select the OS/400 Disk monitor

Click the **Add Empty...** button when you have selected the monitor.

3. For our Warning monitor, select the response level from the field provided and set it to trigger when it is greater than 80.
4. Now, click the **Send Enterprise Console event** checkbox, and select **Warning** from the list. Enter the TEC Server name in the Server field (in this case, it is EventServer). The monitor should now look like that shown in Figure 302 on page 469.

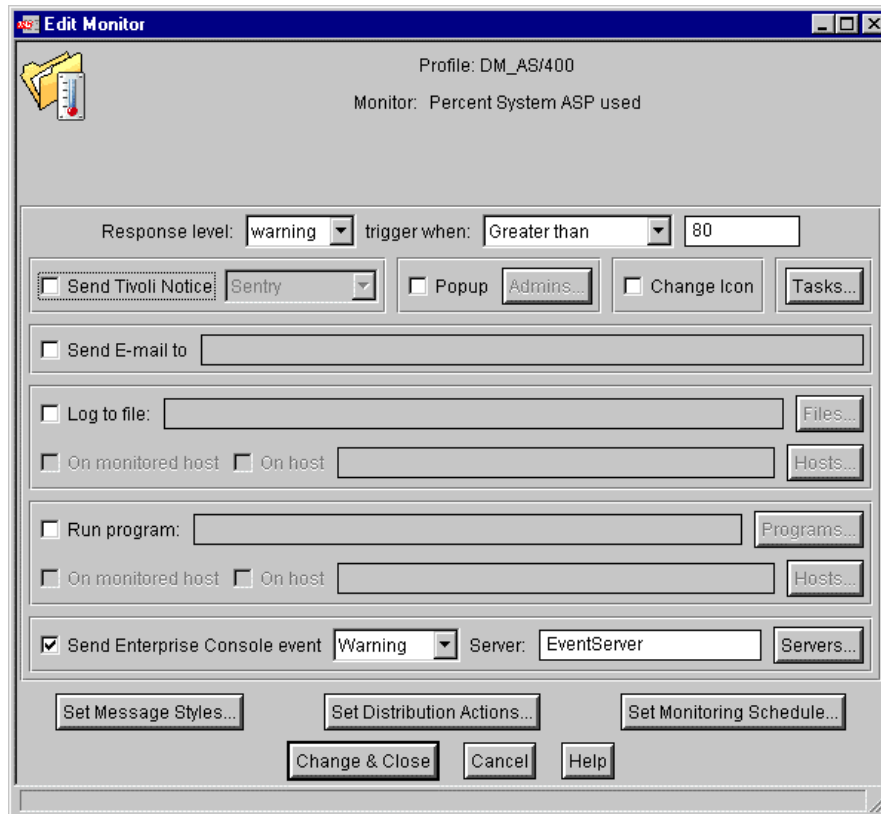


Figure 302. Configuring a Warning AS/400 Disk Monitor

5. Next, we need to configure our monitor to generate a Critical event and run the CLROUTQS program when storage reaches 90 percent. Select **Critical** as the response level, and set it to trigger when it is greater than 90.

6. Select the **Send E-mail to** checkbox and enter the e-mail address of the recipient. In the Run Program field, enter the following command:

```
SBMJOB CMD(CALL PGM(QGPL/CLROUTQS))
```

Now, click the **Send Enterprise Console event** checkbox and select **Critical** from the event list. The Critical Monitor should look like the screen in Figure 303 on page 470.

Profile: DM_AS/400
Monitor: Percent System ASP used

Response level: **critical** trigger when: **Greater than** **90**

☐ Send Tivoli Notice **Sentry** ☒ Popup **Admins...** ☐ Change Icon **Tasks...**

☒ Send E-mail to **colljim@je.ibm.com**

☐ Log to file: **Files...**

☐ On monitored host ☐ On host **Hosts...**

☒ Run program: **SBMJOB CMD(CALL PGM(QGPL/CLROUTQS))** **Programs...**

☒ On monitored host ☐ On host **Hosts...**

☒ Send Enterprise Console event **Fatal** Server: **EventServer** **Servers...**

Set Message Styles... **Set Distribution Actions...** **Set Monitoring Schedule...**

Change & Close **Cancel** **Help**

Figure 303. Configuring a Critical AS/400 Disk Monitor

When you are finished, click the **Change & Close** button to return to the Profile Properties screen shown in Figure 304 on page 471.

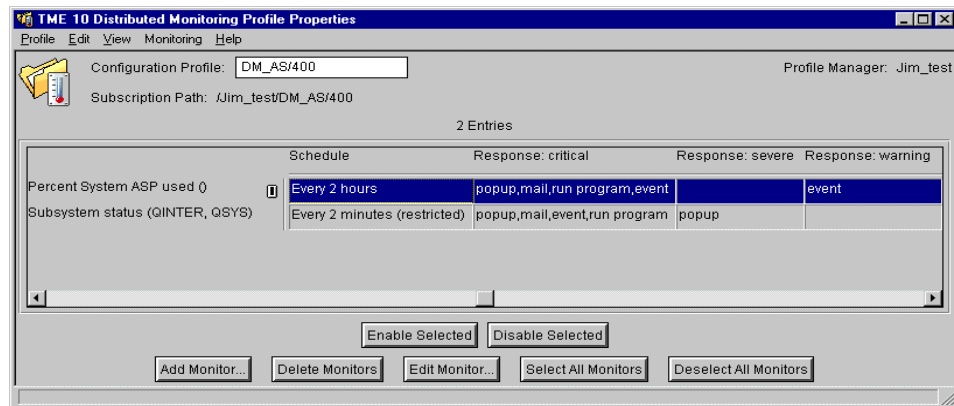


Figure 304. Monitor Profile Properties screen

7. Save the new monitor using the **Save** option in the Profile menu. Now, select **Distribute** from the Profile menu and select the AS/400 to distribute to as shown in Figure 305.

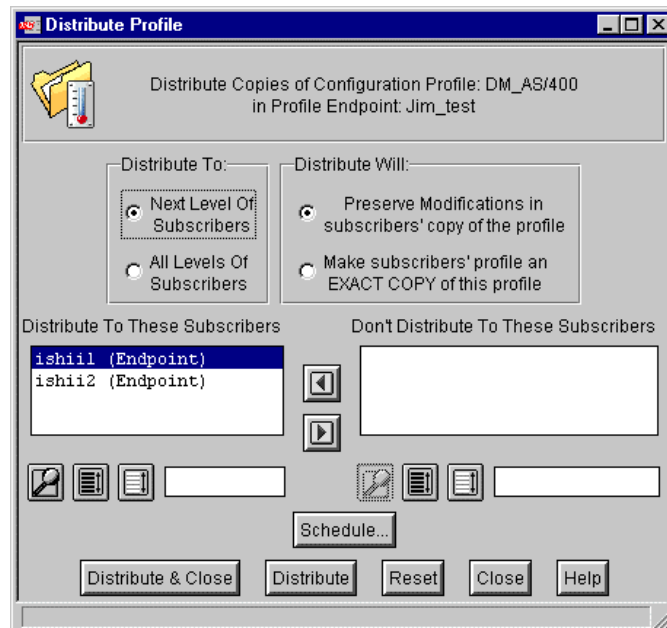


Figure 305. Distributing the AS/400 Disk Monitor

To distribute, click the **Distribute & Close** button.

The monitor will now trigger when the System Storage reaches 80 percent and send an event to TEC if the event occurs.

If the system storage reaches 90 percent, a Fatal event will be logged in TEC. The CLROUTQS program will also run to help reduce the amount of storage being used. While the CLROUTQS program may not free up much space on our AS/400s, it serves as a practical example of how customized programs can be used as actions to Distributed Monitoring events.

We have seen how Tivoli Distributed Monitoring and Tivoli Enterprise Console can be used to monitor our AS/400 resources and applications. Having Tivoli installed allows us to automate our availability and resource management.

9.4 AS/400 user administration and security management

Tivoli User Administration and Tivoli Security Management provide the means for managing all your AS/400 user administration and security from a single graphical interface.

We can use these products to centrally manage all our AS/400 user profiles and security policies. The following two examples show how Tivoli User Administration and Tivoli Security Management can help manage your AS/400 network with a minimum of effort.

9.4.1 Tivoli User Administration for AS/400

One useful feature of Tivoli User Administration is the ability to create a generic user profile template for new AS/400 user profiles. Our first example will show how to create a generic template for user profiles in our Financial Environment.

9.4.1.1 Example 6: Creating a user profile template

In this example, we will create a user profile template for the ishii1 and ishii2 AS/400s. In our environment, there are a number of situations in which a user will require a user profile on both machines; so, creating such a template will save time, be easier to administer, and minimize the chance of errors.

Our template will consist of the default user profile attributes listed in Table 29.

Table 29. User Profile Default attributes

User profile attribute	Tivoli user profile attribute	Default value
User Class	OS/400_Class	*USER
Attention Program	OS/400_Attention_PGM	QGPL/ATTNPGM
Initial Menu	OS/400_Initial_Menu	FIN_APPS/FIN_MENU
Initial Program	OS/400_Initial_PGM	FIN_APPS/FIN_PGM
Limit Capabilities	OS/400_Limit_Cap	*YES
Group Profile	OS/400_Prim_GRP	FIN_GROUP

To create the AS/400 user profile template, complete the following steps:

1. Create a new User Administration profile using the **Create** option in the Profile menu of the Profile Manager screen. Choose **UserProfile** as the profile type as shown in Figure 306.

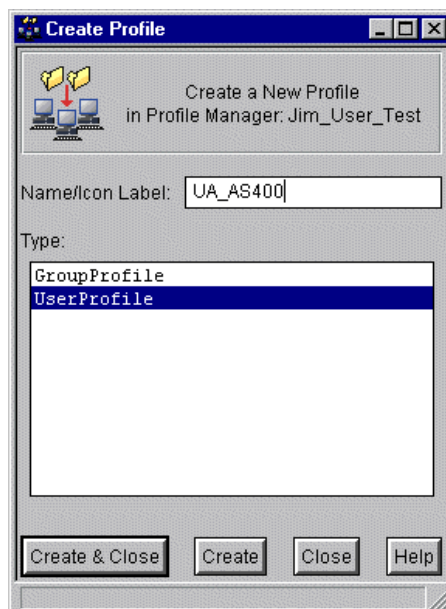


Figure 306. Creating a new User Administration profile

We will name the profile UA_AS400. Click **Create & Close** when finished.

2. Open the **User Profile Properties** screen by double clicking on the new profile icon, in this case, **UA_AS400**.
3. Before setting up a new template, we need to populate the User Administration database with our existing AS/400 user profiles. To do this, select the **Populate...** option from the Profile menu as shown in Figure 307.

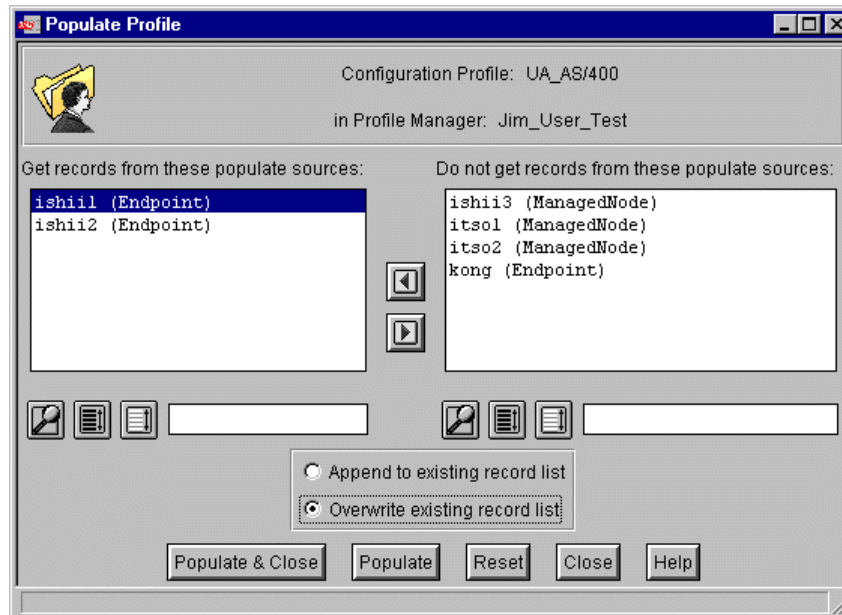


Figure 307. Populating the User Administration database

Select the AS/400 endpoints from which you want to retrieve profile records, and click the **Populate & Close** button.

4. On the User Profile Properties screen, select the **Default Policies** option from the Edit menu.
5. Use the scroll bar for the Attributes list to locate the OS/400 user profile attributes. We will select the **OS/400_Attention_PGM** attribute first, as shown in Figure 308 on page 475, to set the program that will run when an AS/400 user presses the attention key.

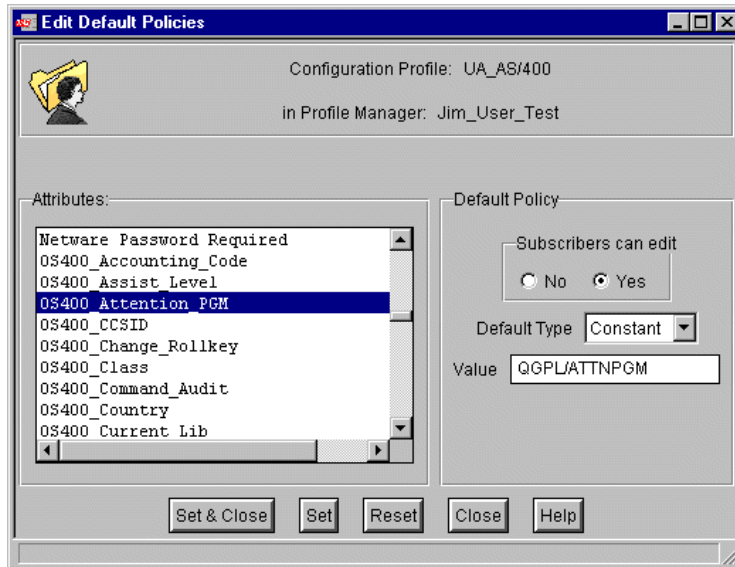


Figure 308. Select a default user profile attribute for OS/400

In the **Value** field, enter the name of the attention program and its library in standard OS/400 notation, for example, QGPL/ATTNPGM.

6. We will not show how to configure the rest of the default attributes listed in Table 29 on page 473 because the procedure is similar for each attribute. When you have set all the default values, click **Set & Close** to exit.
7. Now, we will set up a new profile using the default template that we just created. On the User Profiles Properties screen, click the **Add User** button.
8. On the User Properties screen, choose **OS/400** from the Category list. Next, enter the name of the profile in the field provided, and click the **Generate Defaults** button. This option will take the values from our template and populate the fields in the profile definition. Figure 309 on page 476 shows the OS/400_Session_Startup section of the User Properties screen.

The screenshot shows a 'Tivoli User Administration' window titled 'Add Record'. On the left is a tree view of categories under 'OS400', with 'OS/400 Session Startup' highlighted. The main area contains the following fields:

- User Name:** NEW_USER
- Common Login:** newuser
- Common Password:** *****
- OS/400 Session Startup** (Section Header)
 - Initial program:** FIN_APPS/FIN_PGM
 - Initial menu:** QGPL/FIN_MENU
 - Display sign-on information:** Use system value
 - Limit device sessions:** Use system value

At the top right, it says 'User Profile: UA_AS/400' and 'Profile Manager: Jim_User_Test'.

Figure 309. Generated defaults for AS/400 user profiles

Notice that the default values have been added in the Initial program and Initial menu fields.

Note

When creating a user profile with Tivoli User Administration, it will set the initial password the same as the User Name field. The password can be manually changed to comply with your environment's security policy.

- To save your new profile, click the **Add & Close** button. The new profile definition will appear in the User Profile Properties screen as shown in Figure 310 on page 477.

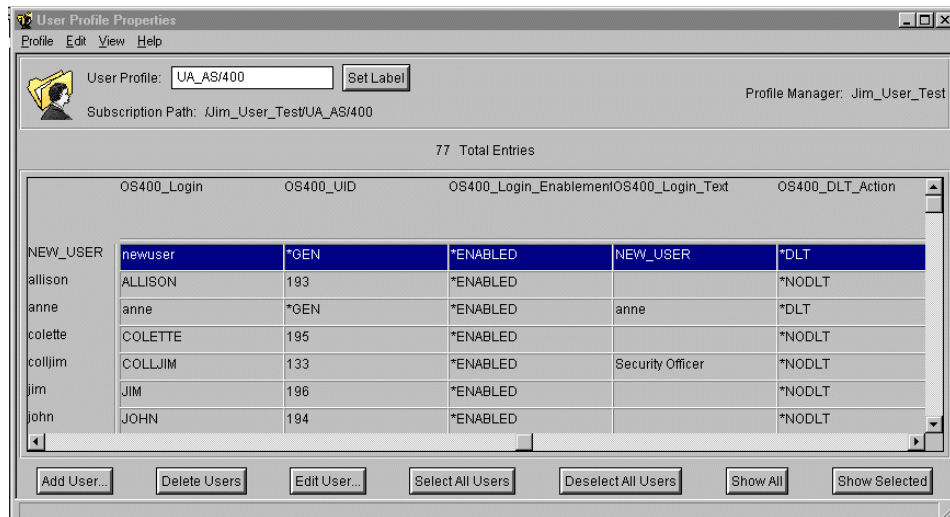


Figure 310. Viewing AS/400 user profiles

10. To create the profiles on our AS/400s, we have to distribute to both machines. Select the **Distribute** option in the Profile menu. Now, in the panel in Figure 311, choose the machines on which the profiles will be created, in this case, ishii1 and ishii2.

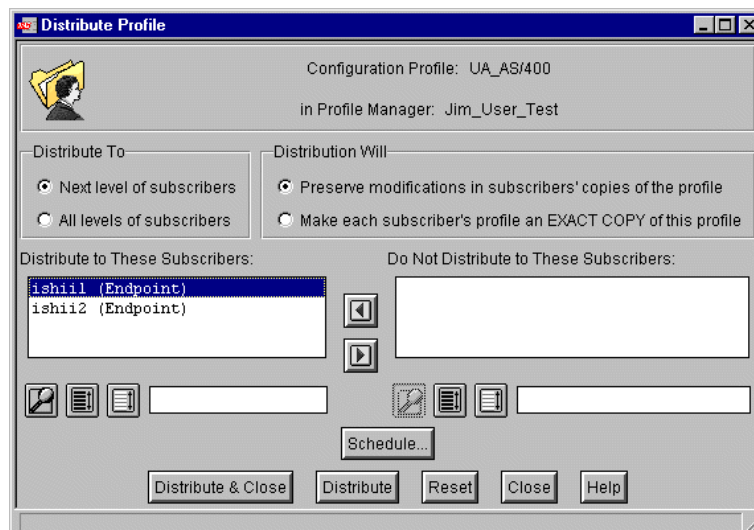


Figure 311. Distributing new AS/400 profiles

Select **Distribute and Close** when you have chosen the AS/400s to distribute to.

Creating a generic user profile template for the AS/400 can save a lot of time when creating user profiles. It also minimizes the chance of errors since we are using a pre-defined template. You can also eliminate duplication of work as now you only have to create the profile once and distribute it to as many AS/400's as required in your environment.

Tivoli User Administration has many other features than can help you manage your systems profiles across all platforms. Central management reduces the time needed to create user profiles and allows you to integrate this support with other operating systems in your network.

For more information about configuring and using Tivoli User Administration, refer to the *User Admin 3.6.2 User's Guide*, GC32-0291.

9.4.2 Tivoli Security Management for AS/400

Tivoli Security Management for AS/400 provides a centrally-managed security tool for your AS/400 network. It enables you to implement a consistent security policy for your AS/400 environment. Changes can be easily distributed to all you AS/400s with the click of a button.

The next example will show how we change the AS/400 login policy in for all the AS/400s in our Financial environment.

9.4.2.1 Example 7: Changing AS/400 system security values

Currently, in our AS/400 environment, we are using the default login policy security values shipped with our AS/400s. We want to change these values to those listed in Table 30.

Table 30. Default User Profile Attributes

Attribute	Value
Suspend Inactive Accounts Suspend if Inactive (after x days)	90
Lock Account upon Multiple Login Failures Max number of failed login attempts	5
Limit Concurrent Logins	1

To set these security values, complete the following steps:

1. Using the Profile Manager, create a new security profile. Choose a type of SecurityProfile in the Create Profile screen as shown in Figure 312.

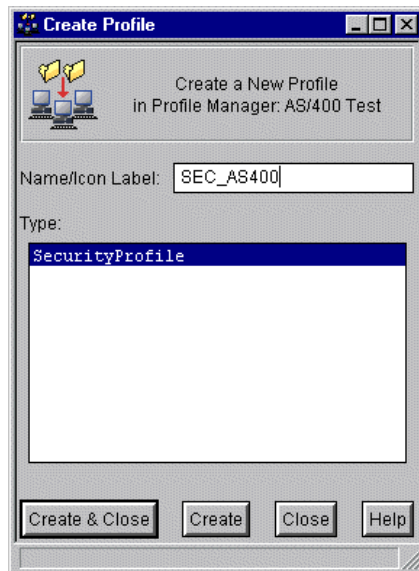


Figure 312. Creating a Security Management profile

We will call the profile SEC_AS400. Select **Create & Close** when finished.

2. Double click on the new profile icon to open the Security Profile screen shown in Figure 313.

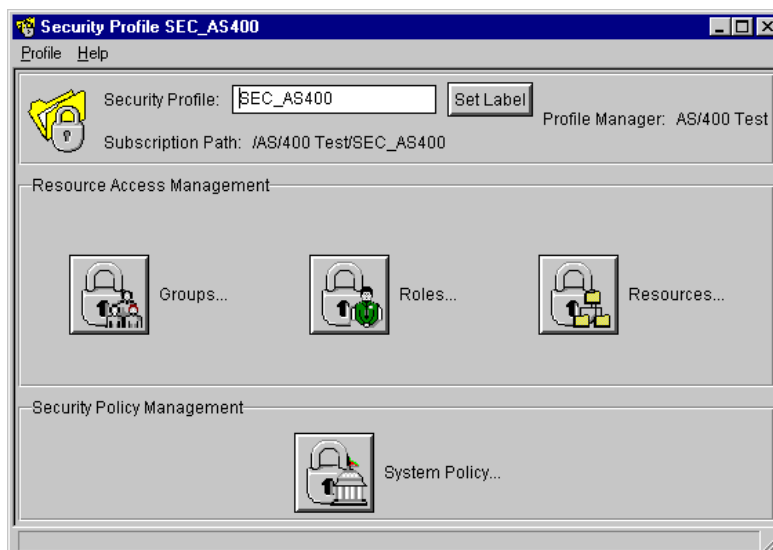


Figure 313. Security Profile screen

3. Before we change any of the security values on the AS/400, we need to populate the Security Management database with the existing Security values. To do this, select the **Populate** option from the Profile menu.
4. On the Populate Profile screen, select the AS/400s with which you want to populate the Security Management database. In this case, we will select ishii1 and ishii2 as shown in Figure 314.

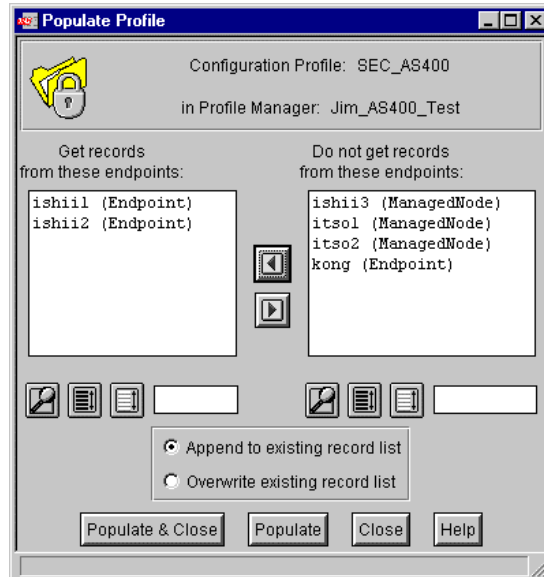


Figure 314. Populating the Security Management database

Click on **Populate & Close** when you are done.

5. When the populate function has finished, click on the **System Policy** icon in the Security Policy Management section of the Security Profile screen.
6. You will see that a new record has been created. This record contains the current login policy values from our AS/400 subscribers. See Figure 315 on page 481.

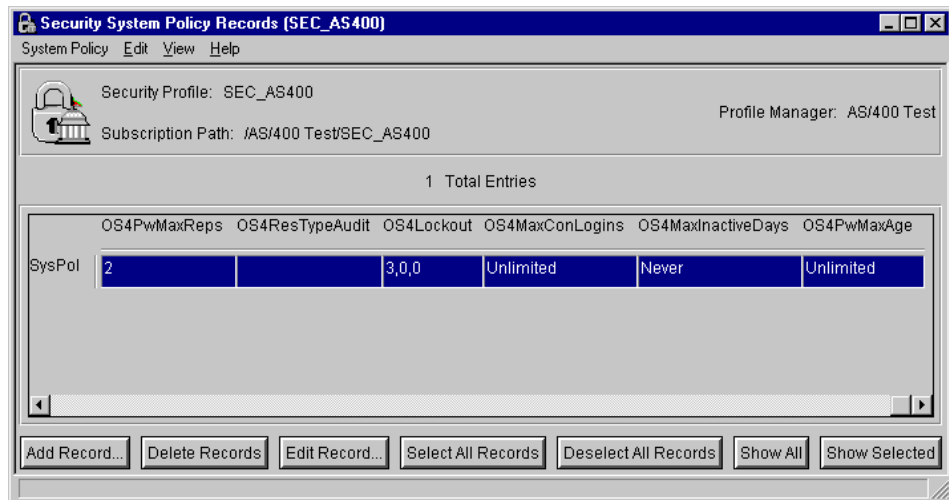


Figure 315. Security System Policy records

To edit the record, highlight the SysPol record, and click the **Edit Record...** button.

- On the System Policy Record Properties screen, select **Edit Login Policy** from the Actions list. A list of policy types will be displayed. Select **OS/400 Login Policy** from the list as shown in Figure 316.

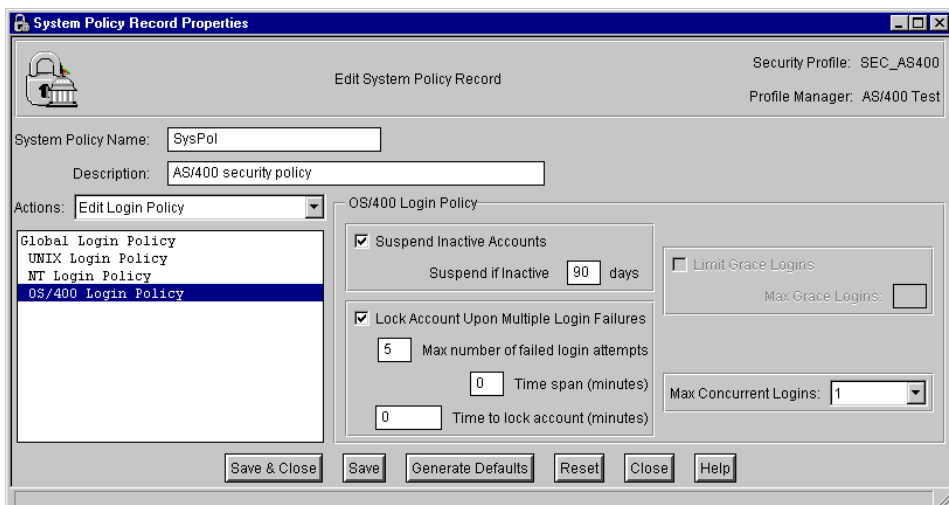


Figure 316. Editing the OS/400 login policy

8. Select the **Suspend Inactive Accounts** checkbox, and enter 90 in the Suspend if Inactive field.
9. Select the **Lock Account Upon Multiple Login Failures** checkbox, and enter 5 in the Max number of failed login attempts field.
10. Set the Max Concurrent Logins field to 1. When finished, click the **Save & Close** button.
11. Close the Security System Policy Records screen and return to the Security Profile screen.
12. Select the **Distribute** option from the Profile.
13. Choose the AS/400s to which you want to distribute the new login policy. In this case, we will choose ishii1 & ishii2 as shown in Figure 317.

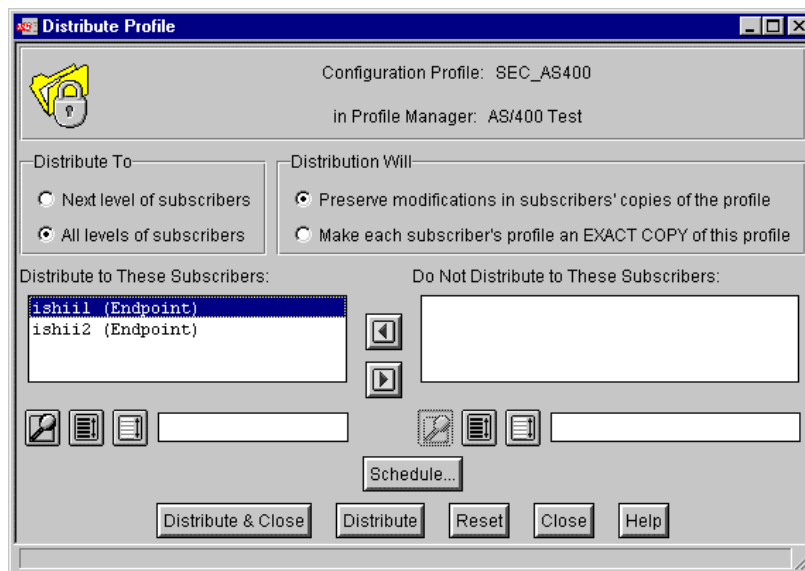


Figure 317. Distribute security policy

Click on the **Distribute & Close** button to distribute the new login policy.

The new login policy will be distributed to the AS/400s in our network. The new login policy values will be changed in the AS/400 System Values. Use the `WRKSYSVAL` command to view the changes.

Tivoli Security Management has many other security functions. It can be used to secure all your resources, such as files, libraries, objects, and user profiles. Through the use of roles, you can organize your resources, based on job

function, to suit any environment. One of the advantages of using Tivoli Security Management is that, once a policy is created, it can be distributed to any of your AS/400s, thus, maintaining a consistent policy across all your machines. Security auditing is achieved through the use of the Tivoli Security Reports task library, which provides predefined tasks to monitor the QAUDJRN security journal on the AS/400.

For more information about using and configuring Tivoli Security Management, consult the *Security Management Users Guide 3.6*, SC31-8475.

9.5 AS/400 inventory and deployment management

In this section, we will use examples to show how Tivoli Inventory and Tivoli Software Distribution for the AS/400 can be used to manage all your hardware and software inventories. Tivoli Inventory will allow you to scan and query your inventories while Tivoli Software Distribution allows mass deployment of new software, databases, and Preventative Temporary Fixes (PTF).

9.5.1 Tivoli Inventory for AS/400

Tivoli Inventory can be used for maintaining all your AS/400 hardware and software inventory in one database. Our first example shows how we perform a system hardware and software scan and how to view the results using the query tools provided by Tivoli. In a more advanced example, we show how we can configure the product to scan for installed PTFs.

9.5.1.1 Example 8: Hardware and software inventory scans

The following steps will detail how to collect and query the hardware and software resources on your AS/400 network. When Tivoli Inventory runs, the collected information is stored in a relational database, thus, creating queries for the inventory data.

1. The first step is to create an Inventory Profile. To do this, click on the **Profile** option in the Create menu. Choose **Inventory Profile** as the type, and click on **Create and Close**.
2. Next, add your AS/400 subscribers using the Subscribers option in the Profile Manager menu as shown in Figure 318 on page 484.

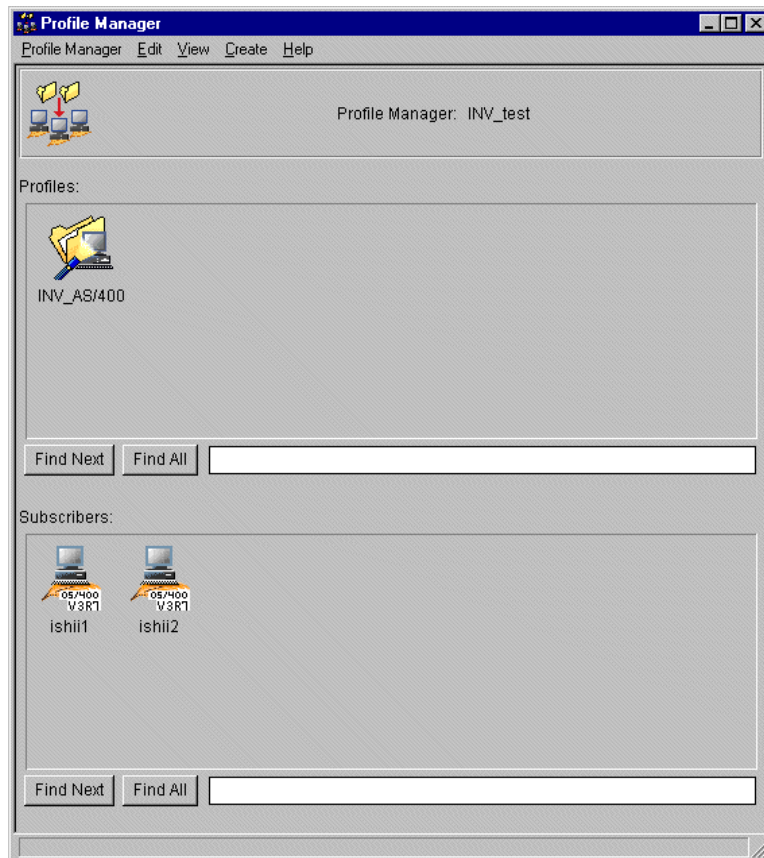


Figure 318. Creating an inventory profile and subscribers

3. Now, we need to modify the profile to perform the hardware and software scans for our selected AS/400s. Open the inventory profile by right-clicking on the new icon. Choose the **Customize...** option on the menu.
4. The Customize Inventory Profile screen will be displayed as shown in Figure 319 on page 485.

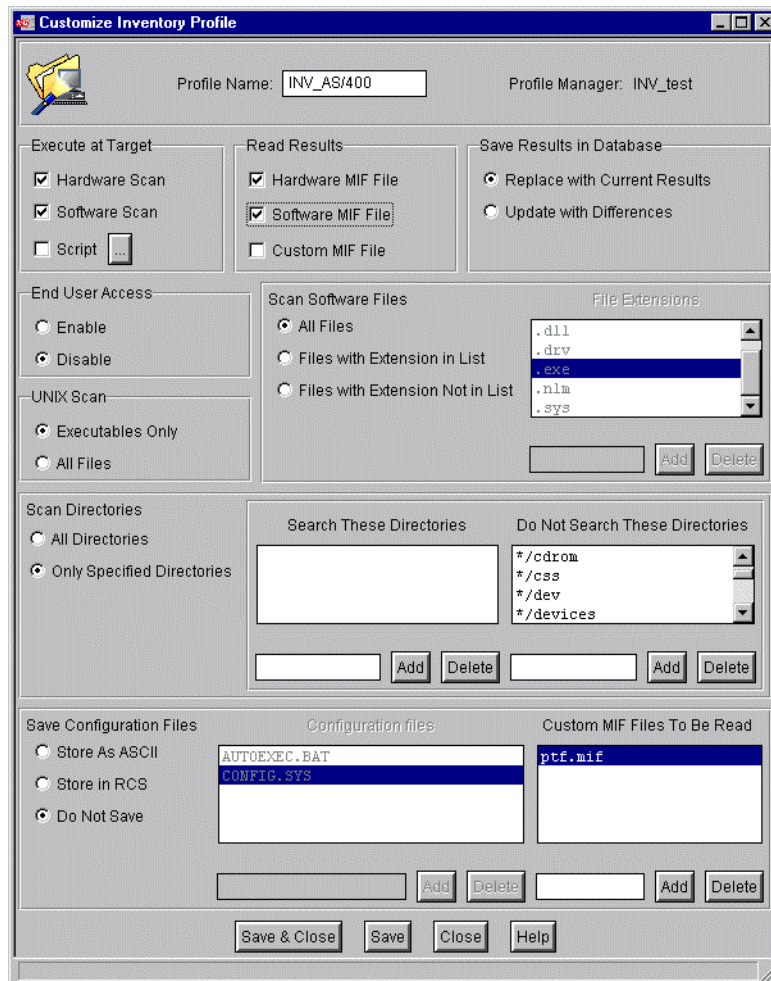


Figure 319. Customizing inventory scans

We want to scan for both hardware and software in this example; so, in the Execute at Target window, select both the **Hardware Scan** and **Software Scan** checkboxes. Also, make sure that the Hardware MIF file and Software MIF file checkboxes are selected in the Read Results window. Leave the defaults selected for the other options on the screen, and then click on **Save & Close**.

5. Now that we have configured the profile to run hardware and software scans, we need to distribute the profile to our AS/400s. To do this, right click on the inventory profile in the Profile Manager again and select **Discover** from the menu. The screen shown in Figure 320 appears.

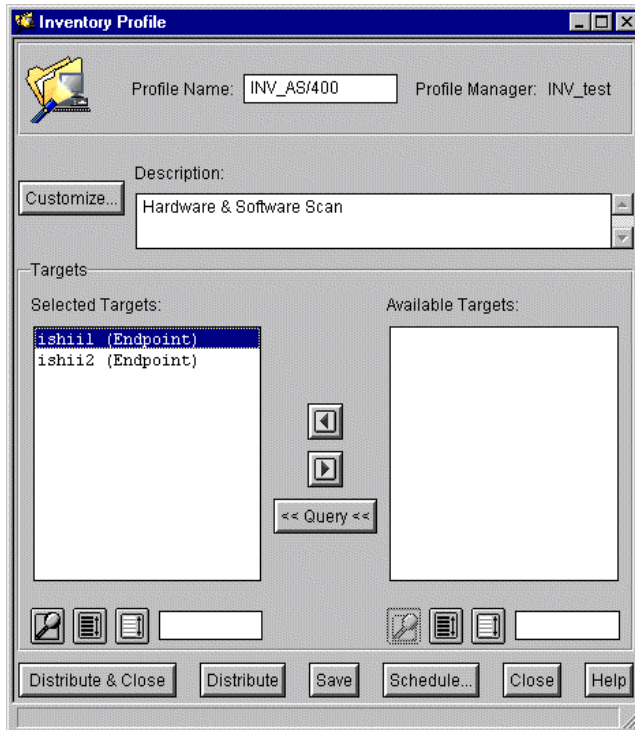


Figure 320. Distributing inventory profiles.

Select the AS/400s to which you want to distribute. In this case, we will select ishii1 and ishii2. To run the inventory scan, click on the **Distribute & Close** button.

Note

Initial hardware and software scans can sometimes take a long time to run and may effect system performance. To avoid performance degradation during business hours, you can use the Schedule option to schedule the scan at a more convenient time.

6. When the distribution has finished, we can query the results using the INVENTORY_QUERIES query library in the inventory policy region as shown in Figure 321 on page 487.

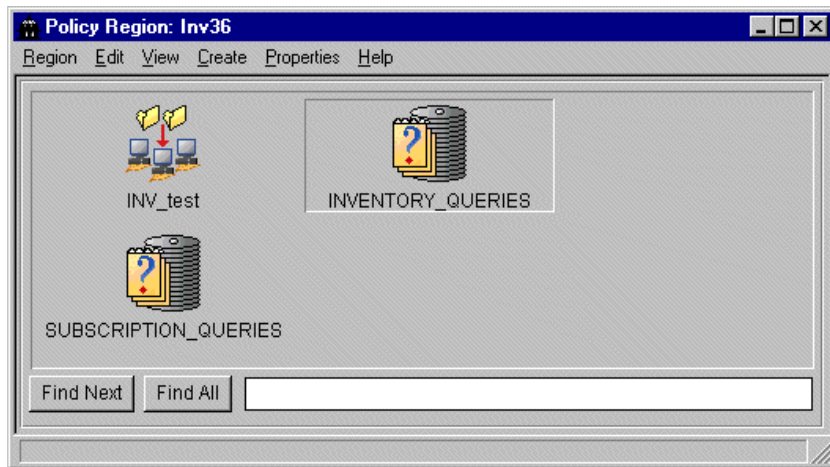


Figure 321. Inventory queries library

Double click in the **INVENTORY_QUERIES** icon to open the library. To view the results of our inventory scans, we will use the **INVENTORY_HARDWARE** and the **INVENTORY_OS400_SOFTWARE** queries in this library as shown in Figure 322.

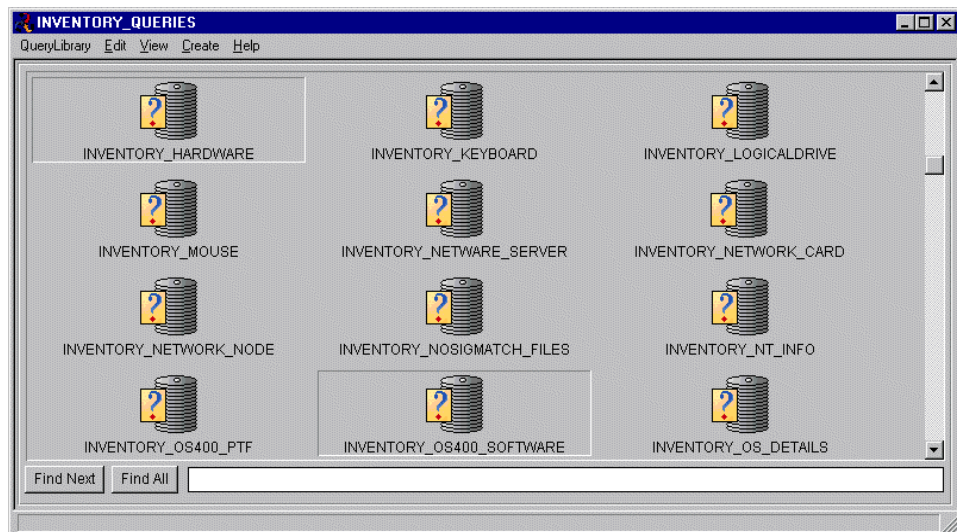


Figure 322. Inventory queries for the AS/400

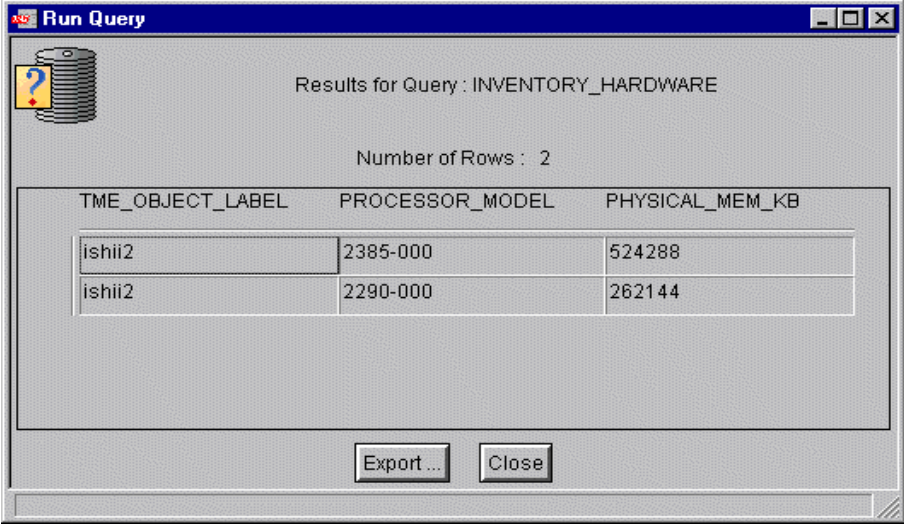
For this example, we will use the INVENTORY_HARDWARE query to view the results of our hardware scan. Double click the query icon to open the Edit Query screen.

7. For this query, we only want to view the hardware installed on the ishii2 AS/400. The TME_OBJECT_LABEL column contains the AS/400 system name information. We will use this column in our Where Clause to restrict the query results to ishii2. To do this, enter `TME_OBJECT_LABEL` in the Column Name field and `ishii2` in the Column Value field. Click the **Add** button to add the query logic to the Where Clause window as shown in Figure 323.

The screenshot shows the 'Edit Query' window for the 'INVENTORY_HARDWARE' query. The window is titled 'Edit Query' and 'INVENTORY_HARDWARE'. It has a 'Query Name' field with 'INVENTORY_HARDWARE' and a 'Description' field with 'Basic Inventory Hardware'. Below these is a 'Repository' dropdown set to 'Inventory' and a 'Table/View Name' field with 'INVENTORYDATA'. There are 'Set' and 'No Duplicates' checkboxes. The 'Chosen Columns' section lists 'TME_OBJECT_LABEL', 'PROCESSOR_MODEL', and 'PHYSICAL_MEM_KB'. The 'Available Columns' section lists 'HWARE_SYS_ID', 'COMPUTER_ARCH', 'BOOTED_OS_VER', 'PROCESSOR_SPEED', 'COMPUTER_ARCHITECTURE', and 'COMPUTER_SCANTIME'. The 'Where Clause' section contains the condition '(TME_OBJECT_LABEL = 'ishii2')'. The 'Additional Clauses' section shows 'order by TME_OBJECT_LABEL'. The bottom of the window has buttons for 'Save & Close', 'Save', 'Run Query ...', 'Close', and 'Help'.

Figure 323. AS/400 hardware query

In this query we only want to view the system name, processor installed, and physical memory details. To do this, we select only these columns in the Columns field. When you have entered the query details, click the **Run Query** button to run the query and view the results as shown in Figure 324.



Results for Query : INVENTORY_HARDWARE

Number of Rows : 2

TME_OBJECT_LABEL	PROCESSOR_MODEL	PHYSICAL_MEM_KB
ishii2	2385-000	524288
ishii2	2290-000	262144

Export... Close

Figure 324. AS/400 hardware query results

Defining a software query is very similar, and you should follow the same steps as for the hardware query example. However, if you want to scan for AS/400 Program Temporary Fixes (PTFs), you will have to configure the system to do so. This is described in the next example.

9.5.1.2 Example 8: Scanning for AS/400 PTFs

To perform an inventory scan for AS/400 PTFs, we must first extend the Inventory Configuration Repository to support the PTF format. We do this by running a supplied script that creates the necessary PTF format using a customized Management Information Format (MIF) file. We will also have to create a new inventory profile that runs a script to read the results collected on the MIF file, ptf.mif, on the AS/400.

In this example, we will configure Tivoli Inventory to scan for OS/400 PTFs with a product ID of 5769SS1.

To do this, perform the following steps:

1. On the TMR server (in this case, itso1) we need to connect to our database and run the following script:

```
tivoli_dbtype_schema_as400_ptf.sql
```

where `dbtype` is one of the following in Table 31.

Table 31. Database type for AS/400 PTF script

Database type	dbtype
DB2	db2
Microsoft SQL	ms_sql
Oracle	ora
Informix	inf
Sybase	syb

These scripts are located in the
\$BINDIR/TME/INVENTORY/SCRIPTS/RDBMS directory.

2. Create a new inventory profile in the Profile Manager screen as shown in Figure 325 on page 491.

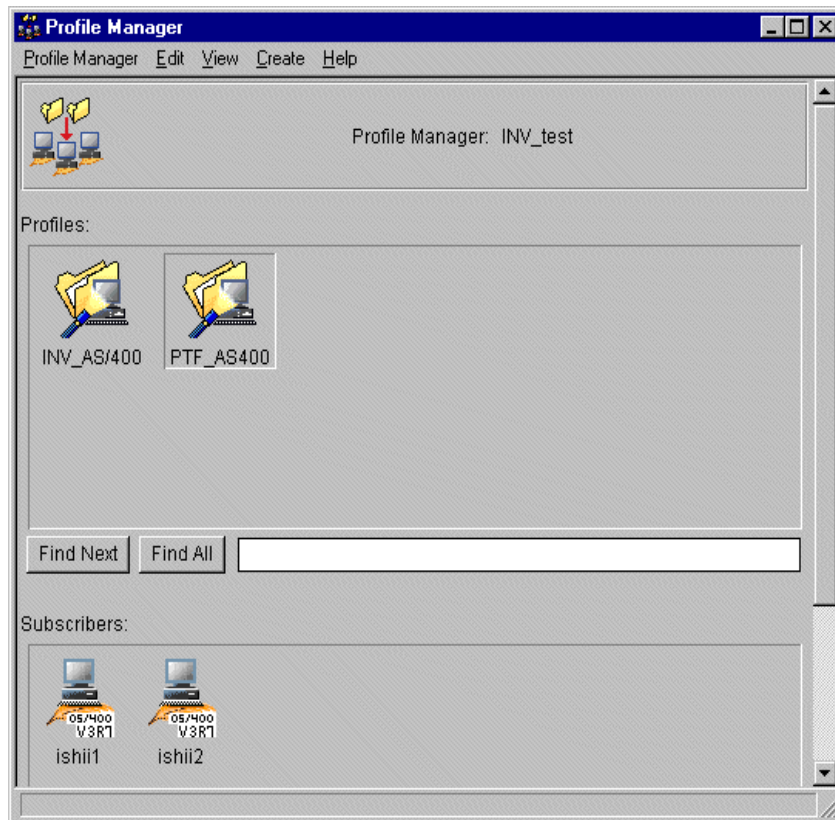


Figure 325. Creating a new inventory profile for AS/400 PTFs

3. Right click on the profile icon (in this case, **PTF_AS400**), and select **Customize** from the menu.
4. On the Customize Inventory Profile screen, click the **Script** checkbox, and then press the ... button beside it. Enter the following command in the field provided:

```
"CALL PGM(QLCF CACHE/OS4_PTFDIS) PARM(5769SS1)"
```

Note

You must place double quotes (" ") around the `CALL` command.

To scan for more than one product's PTFs, include the product ID in the `PARM` section of the command, separated by spaces, for example:

```
CALL PGM(QLCF CACHE/OS4_PTFDIS) PARM(5769SS1 5769999 5769XW1)
```

If the `PARM` section of the command is omitted, the command will scan all PTFs for all products installed on the AS/400.

5. Since we are using a custom MIF file, select the **Custom MIF file** checkbox in the Read Results section. We also need to add `ptf.mif` to the Custom Files to be Read section as shown in Figure 326 on page 493.

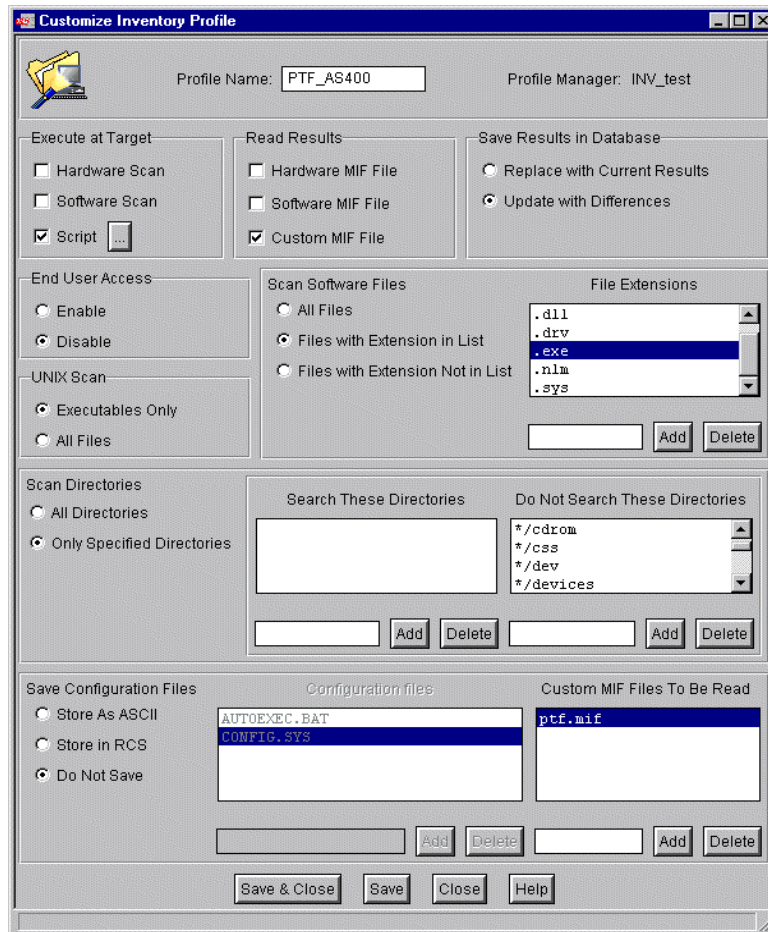


Figure 326. Customizing an inventory profile for an AS/400 PTF scan

When you have finished entering the information, click **Save and Close**.

6. Now, we need to distribute the profile to our AS/400s, ishii1 and ishii2. Right click on the **PTF_AS400** inventory profile icon, and select the **Discover...** option from the menu. Now, select the to which AS/400s to distribute and click on the **Distribute and Close** button.

9.5.1.3 Example 9: Querying PTF scan results

Now that we have collected our AS/400 PTF information, we want to query the results of the scan. To do this, Tivoli provides the INVENTORY_OS400_PTF query in the INVENTORY_QUERIES library.

However, this query does not come pre-installed, and we must complete the following steps before using it:

1. Create a temporary directory on the TMR Server. Copy the `inventory_as400_queries.sh` script into the temporary directory from the `$BINDIR/TME/INVENTORY/SCRIPTS/QUERIES` directory on the TMR server.
2. Run the script to create the query in the INVENTORY_QUERIES library as follows:

```
inventory_as400_queries.sh region INVENTORY_QUERIES
```

where `region` is the name of the policy region in which you want the `INVENTORY_OS400_PTF` query created.

3. When this command has completed, you should have a new query in your `INVENTORY_QUERIES` library called `INVENTORY_OS400_PTF` as shown in Figure 327.

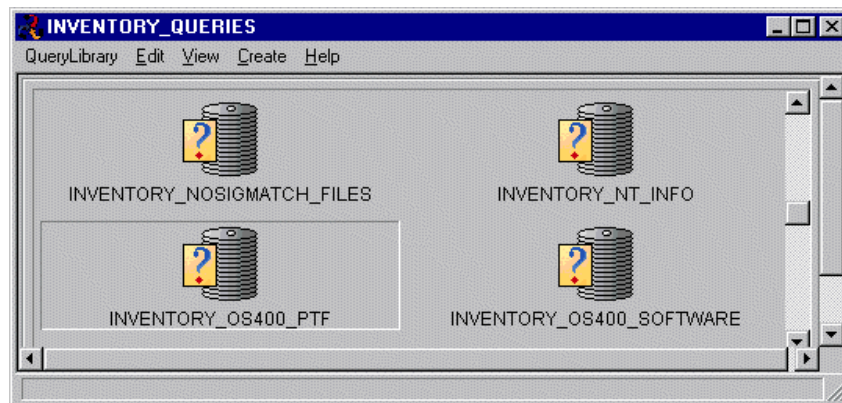


Figure 327. AS/400 PTF query

4. The AS/400 PTF query is now set up and ready to use. Double click on the **INVENTORY_OS400_PTF** icon to configure and run the query. Follow the steps outlined in the example in Section 9.5.1.1, "Example 8: Hardware and software inventory scans" on page 483.

As we have seen, Tivoli Inventory can be used to collect a complete hardware and software inventory for all AS/400s in a network. The information is stored

in a central database that allows us to compare our systems easily through the use of queries. This management tool can be very useful for audit purposes, planning upgrades, and periodic systems maintenance of both hardware and software.

For more information about using Tivoli Inventory and queries, refer to the *TME 10 Inventory 3.6 User's Guide*, GC31-8330.

9.5.2 Tivoli Software Distribution for AS/400

Software deployment management is achieved using Tivoli Software Distribution for AS/400. This product can be used to distribute software and database files to a large number of AS/400 systems simultaneously. Software Distribution can save you a lot of time and effort when a mass installation of software is required. Included are a number of features that can help you automate the installation of new applications, databases, and PTF packages.

9.5.2.1 Example 10: Distributing a new application

The ishii1 AS/400 contains the development and testing environment for new and customized financial environment applications. When an application has been completed and tested, we need to distribute this to the other AS/400s in our network. Traditionally, this would have been a tedious exercise, sending the packages to each AS/400 and restoring them individually. Software Distribution allows you to automate this function. We can send the package that we wish to distribute to the TMR Server, itso1, and, from there, distribute it to all applicable AS/400s using the AS/400 features of Software Distribution.

This example will show how we distribute a new application to ishii2 from our development environment on ishii1. The package is contained in the FIN_NEWAPP library. First, we have to create a save file containing the objects to be distributed and send it using FTP to the TMR server. Then, we configure the Software Distribution profile with the application package details.

Before the file package is distributed, we will run a Before Distribution REXX script to remove the existing library and group user profile. An After Distribution REXX program is also configured to run a REXX script to extract and secure the objects contained in the package.

Note

The Object Packager utility could have been used to create and distribute this application package. However, Object Packager does not support file members with different names to the files themselves. For example, if we distribute the following:

```
QGPL/FIN_APP (MEMBER_1)
```

where MEMBER_1 is a member of the file, it will be restored as

```
QGPL/FIN_APP (FIN_APP)
```

thus, losing the member identifier of the file.

The Before Distribution REXX script will remove the old application library, FIN_OLDAPP, and delete a group profile, FIN_OLDUSR, on the target machine, ishii2, in preparation for our new software. The Before Distribution REXX script, qrexsrc.before, is as follows:

```
"DLTLIB LIB(FIN_OLDAPP)"
"DLTUSRPRF USRPRF(FIN_OLDUSR)"
"SNDDMSG MSG('Old application removed') TOUSR(*SYSOPR)"
```

The After Distribution REXX script will create a new group profile and extract and secure the objects from the FIN_PACK save file. The After Distribution script, qrexsrc.after, is as follows:

```
"CRTUSRPRF USRPRF(FIN_NEWUSR)"
"RSTLIB SAVLIB(FIN_NEWAPP) DEV(*SAVF) SAVF(FIN_PACK)"
"SNDDMSG MSG('Library FIN_NEWAPP restored') TOUSR(*SYSOPR)"
```

The following steps show how we distribute and install the FIN_NEWAPP library and its contents to ishii2.

1. Create the save file on ishii1 containing the libraries and objects needed for the application installation on ishii2. Our save file, QGPL/FIN_PACK, was created using the following commands:

```
CRTSAVF FILE(QGPL/FIN_PACK)

SAVLIB LIB(FIN_NEWAPP) DEV(*SAVF) SAVF(QGPL/FIN_PACK)
```


2. When the save file has been created, we send it to the TMR Server, its01, using ftp. Remember to set the transfer type to binary using the `BINARY` subcommand. In this example, we put the file in the `/tmp` directory of its01 as follows:

```
> bin
> put QGPL/FIN_PACK /tmp/FIN_PACK
```

3. Open the Tivoli desktop, and create a new Software Distribution profile in your policy region. Give the profile a type of **FilePackage** as shown in Figure 328.

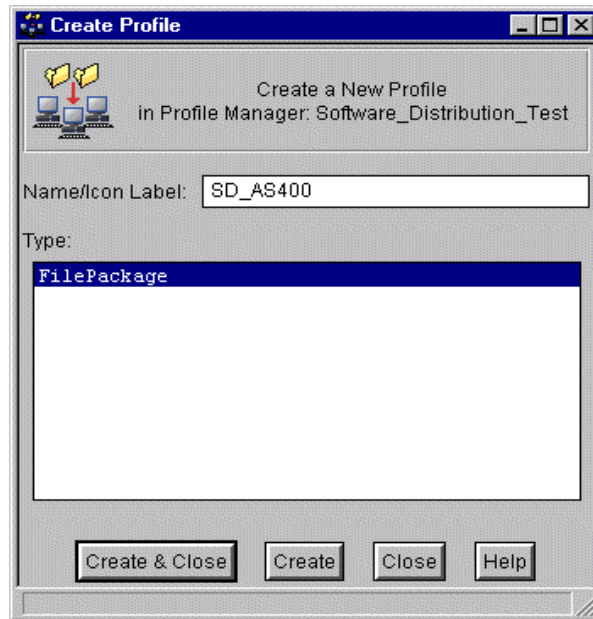


Figure 328. Creating a software distribution profile

4. When the profile icon is created, right click on it and choose the **Properties...** option from the menu.
5. Click the **Source Host** button to select the host on which the file package to distribute resides; in this case, it is its01.
6. In the Source Files and Directories section, click on the **Directories and Files** button to select the file package to distribute. See Figure 329 on page 498.

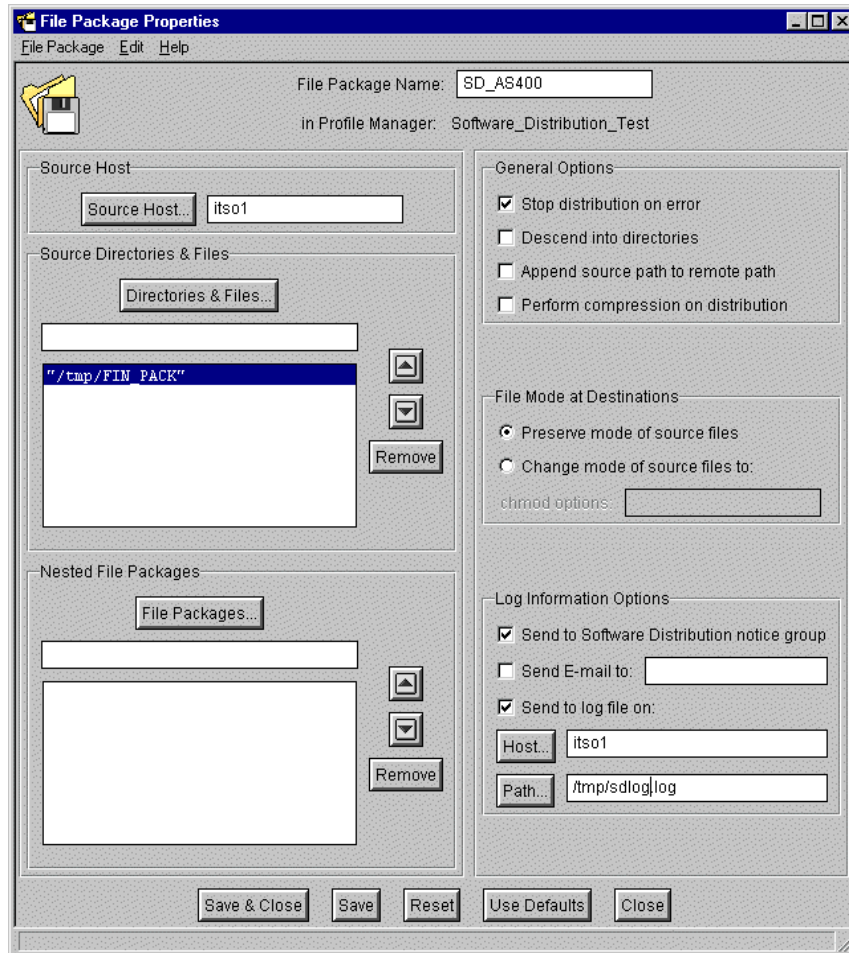


Figure 329. Software distribution properties

If you wish to create a log file for debugging distribution problems, select the **Send to log file on** checkbox and fill in the Host and Path fields.

7. Next, we have to define the procedures that we want to run before and after the program has been distributed. Choose **OS/400 Options...** from the Platform Specific Options in the Edit menu.
8. On the File Package OS/400 Options screen, enter the destination path of the save file containing the objects to be distributed. This must be entered in Integrated File System (IFS) format. We want the save file to be placed in the QGPL library; so, in the Destination Directory Path, we type:

`/QSYS.LIB/QGPL/LIB`

9. To specify our pre-distribution script, click on the **Before Distribution** button. Select **Source Host** in the Get Program section to specify that we want to get the program from the same place as the file package, itso1. Enter the location of the script, /OS400_App/qrexsrc.before, in the Enter program name field as shown in Figure 330.

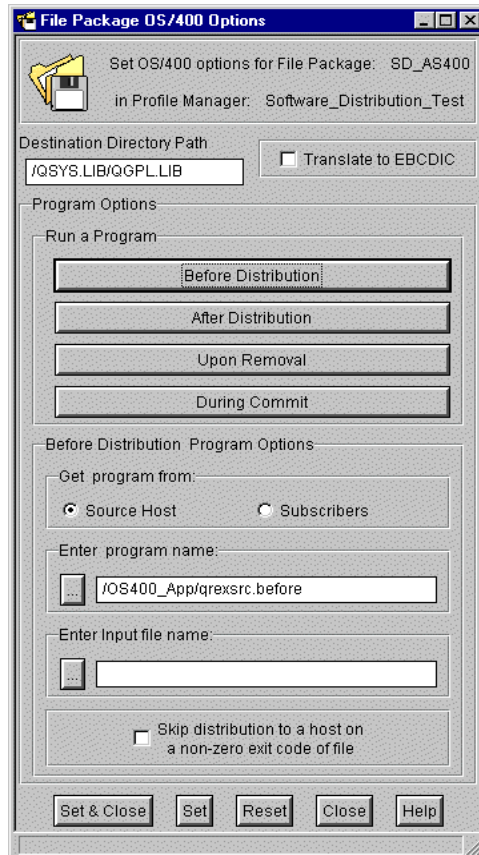


Figure 330. Specifying a Before Distribution program

10. Now, we need to specify our post-distribution program. As before, select the **Source Host** option in the Get Program from section. Next, click on the **After Distribution** button and enter /OS400_App/qrexsrc.after, which is the path to our REXX script on the TMR. See Figure 331 on page 500.

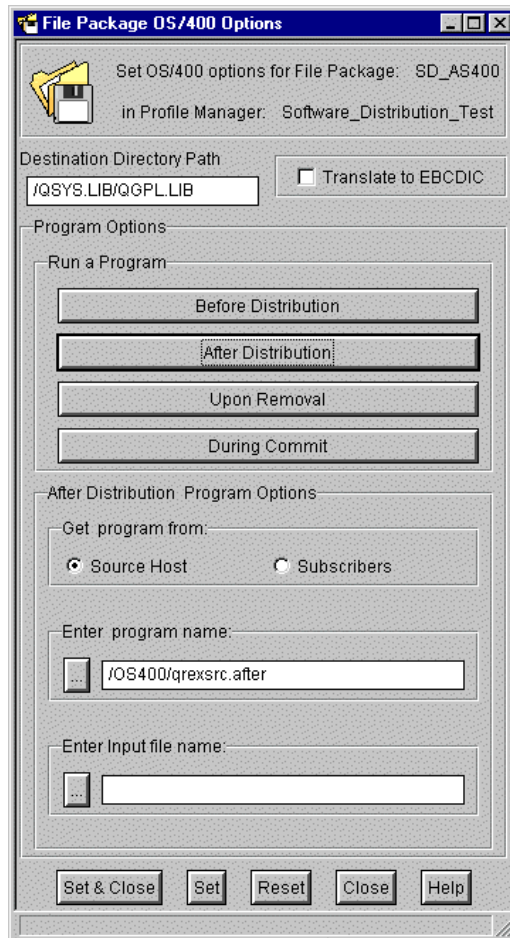


Figure 331. After Distribution program

When you are finished specifying the distribution option, click the **Set & Close** button to return to the File Package Properties screen.

11. To distribute our file package to the AS/400, select the **Distribute** option from the File Package menu. Select the AS/400 to which to distribute from the **Available Subscribers** list, and click **Distribute & Close** to send the package.

9.5.2.2 Example 12: AS/400 PTF distribution REXX script

To distribute PTFs from the TMR, you should follow the same procedure as detailed in Section 9.5.1.2, “Example 8: Scanning for AS/400 PTFs” on page 489. However, we will need to create a new REXX script to load and apply the

PTF package. The following script is an example of how REXX can be used to load and apply a new PTF installed on an AS/400.

We will assume that PTF save file is called QSF55298 and will be distributed to the QGPL library using Tivoli Software Distribution. The PTF is for OS/400, product 5769SS1, and we want to apply it at the next unattended IPL.

```
"LODPTF LICPGM(5769SS1) DEV(*SAVF) SAVF(QGPL/QSF55298)"
"APYPTF LICPGM(5769SS1) DELAYED(*YES)"
"SNDRMSG MSG('PTF has been loaded and applied') TOUSR(*SYSOPR)"
```

Save the REXX script, and add it to the Enter program name field in the After Distribution section of the File Package OS/400 Options screen.

Tivoli Software Distribution allows a fast, efficient, and consistent method of deploying new applications, databases, and PTF packages over large AS/400 environments. Tivoli Software Distribution can save you large amounts of time and help avoid mistakes by automating the whole deployment procedure.

For more information about configuring and using Tivoli Software Distribution, refer to the *TME 10 Software Distribution 3.6 User's Guide*, GC31-8330.

Appendix A. Custom scripts for AS/400 Distributed Monitoring

Disclaimer

The information contained in this document has not been subjected to any formal testing and is distributed *as is*. While each item may have been reviewed by Tivoli for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

A.1 General information

The examples that are shown in this appendix are applied to the following conditions:

- All of the output from the monitor must be returned to the dmepengine on file descriptor 1 (stdout if QIBM_USE_DESCRIPTOR_STDIO is set). Any errors must be returned to the dmepengine on file descriptor 2 (stderr if QIBM_USE_DESCRIPTOR_STDIO is set).
- QIBM_USE_DESCRIPTOR_STDIO is set by the engine whenever it runs monitors, and the Operating System is V4R1 or higher, since this option is not supported before V4R1.
- By default, the monitors are run under the user profile, *nobody*, which is mapped to QTIVUSER on the AS/400. QTIVUSER has a user class of *USER and no special authorities. You may change the user that the dm engine runs monitors as by changing the user in the DM profile.
- All code examples contained here are provided *as is*. No guarantee is made to the functionality or suitability for use in a production environment. Use it at your own risk.

A.2 C and C++ monitors

From C and C++, implementing a monitor is a fairly easy task. The following is an implementation of a program that could be used to monitor a file's size.

A.2.1 Example 1: FileSize.c

For releases prior to V4R1, you can use the function write().

```
#include <stdlib.h>
#include <sys/stat.h>

int main( int argc, char *argv[] )
{
```

```

char *usage = "Usage: FileSize <FileName>"
struct stat sbuf;

/* Check for the right number of parameters */
if( argc < 2 )
{
    write( 2, usage, strlen(usage) );
}

/* Stat the file to get the size */
if( stat( argv[1], &sbuf ) == 0 )
{
    char buf[32];

    sprintf( buf, "%d", sbuf.st_size );
    /* Write the results to stdout (file descriptor #1) */
    write(1, buf, strlen(buf) );
}
else
{
    char buf[1024];
    char dir[512];

    sprintf( buf, "Error: File %s does not exist(%d)\n",
        argv[1], errno );
    /* Write the error to stderr (file descriptor #2) */
    write( 2, buf, strlen(buf) );
    sprintf( buf, "Current directory: %s\n",
        getcwd( dir, sizeof(dir) ) );
    write( 2, buf, strlen(buf) );
    write( 2, usage, strlen(usage) );
}
}

```

To use this, you would create a custom numeric monitor and specify the following for the monitor argument:

```
CRTBNDC PGM(QGPL/FILESIZE) SRCFILE(QGPL/QCSRC) SYSIFCOPT(*IFSIO)
```

A.2.2 Example 2: FileSize.c

For V4R1 and newer, the code could, optionally, be changed to use `printf()` and/or `fprintf()`. It would then read as follows.

```

#include <stdio.h>
#include <sys/stat.h>

int main( int argc, char *argv[] )
{
    char *usage = "Usage: FileSize <FileName>"
    struct stat sbuf;

    /* Check for the right number of parameters */
    if( argc < 2 )
    {
        fprintf( stderr, usage );
        exit( 1 );
    }

    /* Stat the file to get the size */
    if( stat( argv[1], &sbuf ) == 0 )

```



```

{
    /* Write the results to stdout */
    printf( "%d", sbuf.st_size );
}
else
{
    char dir[512];

    /* Write the error to stderr */
    fprintf( stderr, "Error: File %s does not exist(%d)\n",
        argv[1], errno );
    fprintf( stderr, "Current directory: %s\n",
        getcwd(dir,sizeof(dir)) );
    fprintf( stderr, usage );
    exit( 1 );
}
exit( 0 );
}

```

To use this, you would create a custom numeric monitor and specify the following for the monitor argument:

```
CRTBND CPGM(QGPL/FILESIZE) SRCFILE(QGPL/QCSRC) SYSIFCOPT(*IFSIO)
```

Note

In this example,

```
printf( "%d", sbuf.st_size );
```

is equivalent to:

```
fprintf( stdout, "%d", sbuf.st_size );
```

A.2.3 Example 3: FileSize.c

```

#include <stream.h>
#include <sys/stat.h>

int main( int argc, char *argv[] )
{ char *usage = "Usage: FileSize <FileName>"
  struct stat sbuf;

  /* Check for the right number of parameters */
  if( argc < 2 )
  {
      cerr << usage;
      exit( 1 );
  }

  /* Stat the file to get the size */
  if( stat( argv[1], &sbuf ) == 0 )
  {
      /* Write the results to stdout */
      cout << sbuf.st_size;
  }
  else
  {
      char dir[512];

```

```

/* Write the error to stderr */
cerr << "Error: File " << argv[1] << " does not exist("
    << errno << ")\n";
cerr << "Current directory: " << getcwd(dir,sizeof(dir));
cerr << usage;
exit( 1 );
}
exit( 0 );
}

```

At this time, the AS/400 does not provide a native C++ compiler; so, the previous example would need to be cross compiled using Visual Age C++.

To use this, create a custom numeric monitor, and specify the following for the monitor argument:

```
CALL PGM(QGPL/FILESIZE) PARM('<filename>')
```

Where <filename> is any IFS file. For files in QSYS, the call to stat() would need to be replaced with a call to the QUSROBJD API using the OBJD0300 format.

A.3 CL monitors

If you would like to write a monitor in CL, we would suggest creating the following C program and calling it from the CL program to send the results to the engine.

A.3.1 Example 4: CLHelp.c

```

#include <stdlib.h>
#include <string.h>
#include <unistd.h>

int main( int argc, char *argv[] )
{
    /* Write the second argument to the file descriptor
       specified by the first argument */
    write( atoi(argv[1]), argv[2], strlen(argv[2]) );
    exit( 0 );
}

```

To use this, you would create a custom numeric monitor and specify the following for the monitor argument:

```
CRTBND C PGM(QGPL/CLHELP) SRCFILE(QGPL/QCSRC) SYSIFCOPT(*IFSIO)
```

A.3.2 Example 5: CLMONHELP.CMD

Create a command to go with the CLHELP program so that we don't have to do a CALL LIB/PGM to use it.

```
CLMONHELP:  CMD
            PARM  KWD(FD)  TYPE(*CHAR)  RSTD(*YES)  DFT('1')  +
                VALUES('1' '2')  PROMPT('File Descriptor')
            PARM  KWD(MSG)  TYPE(*CHAR)  PROMPT('Message Text')

CRTCMD CMD(QGPL/CLMONHELP)  PGM(QGPL/CLHELP)  +
        SRCFILE(QGPL/QSRC)  SRCLIBR(CLMONHELP)
```

A.3.3 Example 6: SECONDS.CLP

Here is an example using the helper.

```
SECONDS:    PGM
            DCL          VAR(&SEC)  TYPE(*CHAR)  LEN(2)
            RTVSYSVAL  SYSVAL(QSECOND)  RTNVAR(&SEC)
            QGPL/CLMONHELP  FD('1')  MSG(&SEC)

CRTCLPGM PGM(QGPL/SECONDS)  SRCFILE(QGPL/QCSRC)
```

To use this, you would create a custom numeric monitor and specify the following for the monitor argument:

```
CALL PGM(QGPL/SECONDS)
```

A.4 REXX monitors

In this section, we introduce some REXX monitor examples.

A.4.1 Example 7: MSGWjobs.rexx

```
/*****
/*  REXX program to determine the number of jobs in MSGW          */
/*  Parameters:  Subsystem                                         */
/*  Returns:    number of jobs in MSGW                             */
/*  *****/

PARSE ARG subsystem

/* Add a Buffer to the REXX data queue.*/
buffer = 0
'ADDREXBUF  BUFFER(&buffer) '

/* Create active job listing. */
'WRKACTJOB SBS(&subsystem) OUTPUT(*PRINT) '

0/* Create temporary work file to store spooled file output */
/* from (WRKACTJOB) if it doesn't already exist. */
'CHKOBJ QTEMP/ACTJOBLST *FILE'
IF rc \= '0' THEN
    'CRTPF QTEMP/ACTJOBLST RCDLEN(132) '
```

```

/*Copy the last spooled file from WRKACTJOB to the temporary work file*/
'CPYSPLF FILE(QPDSPAJB) TOFILE(QTEMP/ACTJOBLST) SPLNBR(*LAST)'

/*Copy the WRKACTJOB output from QTEMP to the REXX queue.*/
'CPYFTOEXQ FROMFILE(QTEMP/ACTJOBLST) MBR(ACTJOBLST)'

MSGWJobs = 0

DO QUEUED()
/* Pull in a line from the queue that contains the output from */
/* the WRKACTJOB */
PARSE PULL act_Job act_User act_Number act_Type act_Pool act_Pty,
          act_CPU act_Int act_Rsp act_AuxIO act_CPUPercent,
          act_Function act_Status act_Threads

IF act_Status = 'MSGW' THEN DO
/* Increment the number of Jobs in MSGW */
MSGWJobs = MSGWJobs + 1
/* Send a comment with the job that we found */
Say '#' + act_Job + '/' + act_User + '/' + act_Number
END
END

/* Remove Buffer from the REXX data queue.*/
'RMVREXBUF BUFFER(&buffer)'

/* Send the number of jobs we found in MSGW */
say MSGWJobs
EXIT(0)

```

To use this, you would create a custom numeric monitor and specify the following for the monitor argument:

```
STREXPRC SRCMBR(MSGWJOBS) SRCFILE(QGPL/QREXSRC) PARM(QSYSWRK)
```

A.5 QSH monitors

If your AS/400 is V4R2 or later and you have Qshell installed, you can use the following example.

A.5.1 Example 8: FileExists.sh

```

#!/usr/bin/sh

ls $1
if [ $? = 0 ]; then
    echo YES
else
    echo NO
fi

```

To use this, you would create a custom numeric monitor and specify the following for the monitor argument:

```
QSH CMD('/tmp/FileExists.sh /tmp/junk')
```

Note

You cannot currently run a script directly. You need to use the QSH command.

A.6 Getting more information

You may find the following Web pages helpful:

The entire AS/400 online library (including V3R2 - V4R4)

- <http://publib.boulder.ibm.com:80/pubs/html/as400/online/lib.htm>

C Programmers Guide

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC09-2712-00>

C Language Reference

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC09-2711-00>

C and C++ Runtime Reference

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AT800/CONTENTS>

Rexx Programmers guide

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5728-00/CONTENTS>

Rexx Reference

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5729-00/CONTENTS>

CL Programming

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/in-dk/bookmgr/bookmgr/DOCNUM/SC41-5721-01>

CL Reference (Abridged)

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5722-02>

CL Reference: ADDxxx through CHGxxx commands

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5723>

CL Reference: CHKxxx through CVTxxx commands

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5724>

CL Reference: DATA through RPLxxx commands

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5725>

CL Reference: RQSxxx through WRKxxx commands

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/DOCNUM/SC41-5726>

System API reference

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AMA02/BIBLIOGRAPHY.2>

Object APIs

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AMQ02/CCONTENTS>

File APIs

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AMI02/CCONTENTS>

Work Management APIs

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AM702/CCONTENTS>

UNIX Type API reference

- <http://publib.boulder.ibm.com:80/cgi-bin/bookmgr/BOOKS/QB3AM402/CCONTENTS>

Appendix B. AS/400 User Administration task library definition file

This task library definition file is provided *as is* and basic structure only. To create the Task Library, use the `wtll` command. Refer to the *Tivoli Framework Reference Manual*, SC31-8434, for details of the command. In our environment, we issued the command this way:

```
wtll -p itsol-region -P /usr/bin/cat /lpv/UserAdmin/useradmin.tll
```

The following is the `useradmin.tll` file:

```
*****
TaskLibrary "AS400 User Administration Tasks" {
    Context = ("!_", "1");
    Distribute = ("!_", "ALI", 1);
    HelpMessage = (useradmin, "The Display User Profile (DSPUSRPRF) command displays
the contents \n\tof a user profile. The user profile contains the user's operational
\n\tlimits for system resources. This includes the names of the \n\tobjects, commands,
and devices that the user owns, and that the \n\tuser is the primary group for.
\n\n\tContinued help text goes here\n", 2 );
    Requires = ("!_", ">2.5", 1);
    Version = ("!_", "1.0", 1);

    ArgLayout Simple_Text {
        Text;
    };
    ArgLayout Prompted_Simple_Text {
        Text;
        ButtonLabel = (useradmin, "Prompt", 50 );
    };
    ArgLayout User_Profile {
        TextChoice Multi ("", " ", " ")
        {{(useradmin, "All", 137 ) "*All"}};
        ButtonLabel = (useradmin, "Prompt", 50 );
    };
    ArgLayout Info_Type {
        TextChoice Multi ("", " ", " ")
        {{(useradmin, "Basic", 5 ) "*BASIC"}
        {(useradmin, "All", 137 ) "*ALL"}
        {(useradmin, "Authorized CL Commands", 7 ) "*CMDAUT"}
        {(useradmin, "Authorized Devices", 8 ) "*DEVAUT"}
        {(useradmin, "Group Members", 9 ) "*GRPMBR"}
        {(useradmin, "Authorized Objects", 10 ) "*OBJAUT"}
        {(useradmin, "Owned Objects", 11 ) "*OBJOWN"}
        {(useradmin, "Primary Group Objects", 12 ) "*OBJPGP"}};
        ButtonLabel = (useradmin, "Prompt", 50 );
    };
    ArgLayout Output {
        TextChoice Multi ("", " ", " ")
        {{(useradmin, "Screen", 14 ) ""}
        {(useradmin, "Print", 15 ) "*PRINT"}
        {(useradmin, "Outfile", 16 ) "*OUTFILE"}};
        ButtonLabel = (useradmin, "Prompt", 50 );
    };
    ArgLayout Library {
        TextChoice Multi ("", " ", " ")
        {{(useradmin, "Library List", 51 ) "*LIBL"}
        {(useradmin, "Current Library", 52 ) "*CURLIB"}};
        ButtonLabel = (useradmin, "Prompt", 50 );
    };
};
```

```

ArgLayout File_Member {
    TextChoice Multi ("", ",", "")
    {{(useradmin, "First Member ", 53) "*FIRST"}};
    ButtonLabel = (useradmin, "Prompt", 50 );
};
ArgLayout Replace_or_Add_Member {
    RadioButton
    {{(useradmin, "Replace Member", 54) "*REPLACE"}}
    {{(useradmin, "Add to member ", 55) "*ADD"}};
};

ArgLayout Object_Type {
    TextChoice Multi ("", ",", "")
    {{(useradmin, "Alert Table", 57) "*ALRTBL"}}
    {{(useradmin, "Authorization List", 58) "*AUTL"}}
    {{(useradmin, "Binding Directory", 59) "*BNDDIR"}}
    {{(useradmin, "Configuration List", 60) "*CFGL"}}
    {{(useradmin, "Chart Format", 61) "*CHTFMT"}}
    {{(useradmin, "C Locale Description ", 62) "*CLD"}}
    {{(useradmin, "Class", 63) "*CLS"}}
    {{(useradmin, "Command", 64) "*CMD"}}
    {{(useradmin, "Connection List", 65) "*CNNL"}}
    {{(useradmin, "Class of Service Description", 66) "*COSD"}}
    {{(useradmin, "Change Request Description", 67) "*CRQD"}}
    {{(useradmin, "Communication Side Information", 68) "*CSI"}}
    {{(useradmin, "Cross System Product Map", 69) "*CSPMAP"}}
    {{(useradmin, "Cross System Product Table", 70) "*CSPTBL"}}
    {{(useradmin, "Controller Description", 71) "*CTLD"}}
    {{(useradmin, "Device Description", 72) "*DEVD"}}
    {{(useradmin, "Data Area", 73) "*DTAARA"}}
    {{(useradmin, "Data Dictionary", 74) "*DTADCT"}}
    {{(useradmin, "Data Queue", 75) "*DTAQ"}}
    {{(useradmin, "Edit Description ", 76) "*EDTD"}}
    {{(useradmin, "Forms Control Table", 77) "*FCT"}}
    {{(useradmin, "File", 78) "*FILE"}}
    {{(useradmin, "Font Resource", 79) "*FNTRSC"}}
    {{(useradmin, "Font Table", 80) "*FNTTBL"}}
    {{(useradmin, "Form Definition", 81) "*FORMDF"}}
    {{(useradmin, "Filter", 82) "*FTR"}}
    {{(useradmin, "Graphics Symbol", 83) "*GSS"}}
    {{(useradmin, "Internetwork Packet Exchange Description", 84)
    "*IPXD"}}
    {{(useradmin, "Job Description", 85) "*JOB D"}}
    {{(useradmin, "Job Queue", 86) "*JOBQ"}}
    {{(useradmin, "Journal", 87) "*JRN"}}
    {{(useradmin, "Journal Receiver", 88) "JRNRCV*"}}
    {{(useradmin, "Library", 89) "*LIB"}}
    {{(useradmin, "Line Description", 90) "*LIND"}}
    {{(useradmin, "Locale", 91) "*LOCALE"}}
    {{(useradmin, "M36", 92) "*M36"}}
    {{(useradmin, "M36CFG", 93) "*M36CFG"}}
    {{(useradmin, "Menu", 94) "*MENU"}}
    {{(useradmin, "Mode Description", 95) "*MODD"}}
    {{(useradmin, "Module", 96) "*MODULE"}}
    {{(useradmin, "Message File", 97) "*MSGF"}}
    {{(useradmin, "Message Queue", 98) "*MSGQ"}}
    {{(useradmin, "Node Group", 99) "*NODGRP"}}
    {{(useradmin, "Node List", 100) "*NODL"}}
    {{(useradmin, "NetBIOS Description", 101) "*NTBD"}}
    {{(useradmin, "Network Interface Description", 102) "*NWID"}}
    {{(useradmin, "Network Server Description", 103) "*NWSD"}}
    {{(useradmin, "Output Queue", 104) "*OUTQ"}}
    {{(useradmin, "Overlay", 105) "*OVL"}}
}

```



```

        {(useradmin, "Page Definition", 106 ) "**PAGDFN"}
        {(useradmin, "Page Segment", 107 ) "**PAGSEG"}
        {(useradmin, "Print Descriptor Group", 108 ) "**PDG"}
        {(useradmin, "Program", 109 ) "**PGM"}
        {(useradmin, "Panel Group", 110 ) "**PNLGRP"}
        {(useradmin, "PRDAVL", 111 ) "**PRDAVL"}
        {(useradmin, "Product Definition ", 112 ) "**PRDDFN"}
        {(useradmin, "PRDLOD ", 113 ) "**PRDLOD"}
        {(useradmin, "PSF Configuration", 114 ) "**PSFCFG"}
        {(useradmin, "Query Management Form", 115 ) "**QMFORM"}
        {(useradmin, "Query Management Query", 116 ) "**QMORY"}
    {(useradmin, "Query Definition", 117 ) "**QRYDFN"}
        {(useradmin, "Reference Code Translation Table", 118 ) "**RCT"}
        {(useradmin, "System/36", 119 ) "**S36"}
        {(useradmin, "Subsystem Description", 120 ) "**SBSD"}
        {(useradmin, "Search Index", 121 ) "**SCHIDX"}
        {(useradmin, "Spelling Aid Dictionary ", 122 ) "**SPADCT"}
        {(useradmin, "SQL Package", 123 ) "**SQLPKG"}
        {(useradmin, "Service Program", 124 ) "**SRVPGM"}
        {(useradmin, "Session Description", 125 ) "**SSND"}
        {(useradmin, "Server Storage", 126 ) "**SVRSTG"}
        {(useradmin, "Translation Table", 127 ) "**TBL"}
        {(useradmin, "User Index", 128 ) "**USRIDX"}
        {(useradmin, "User Queue", 130 ) "**USRQ"}
        {(useradmin, "User Space", 131 ) "**USRSPC"}
        {(useradmin, "Validation List", 132 ) "**VLDL"}
        {(useradmin, "Work Station User Customization", 133 ) "**WSCST"};
    ButtonLabel = (useradmin, "Prompt", 50 );
};

    ArgLayout CurOwn_Authority {
        RadioButton
            {{(useradmin, "Revoke", 27 ) "**REVOKE"}}
        {(useradmin, "Same", 28 ) "**SAME"}};
    };
ArgLayout Yes_No {
    RadioButton
        {{(useradmin, "Yes", 134 ) "**YES"}}
        {(useradmin, "No", 135 ) "**NO"}};
    };
ArgLayout User {
    TextChoice Multi ("", "", "")
    {{(useradmin, "None", 136 ) "**NONE"}};
    ButtonLabel = (useradmin, "Prompt", 50 );
};
ArgLayout System_Name {
    TextChoice Multi ("", "", "")
    {{(useradmin, "Local", 40 ) "**LCL"}}
    {(useradmin, "PC", 41 ) "**PC"}}
    {(useradmin, "Error", 42 ) "**ERROR"}};
    ButtonLabel = (useradmin, "Prompt", 50 );
};
Task Display_User_Profile {
    Description = (useradmin, "Display User Profile (DSPUSRPRF)", 1 );
    HelpMessage = (useradmin, "The Display User Profile (DSPUSRPRF) command displays the
    contents \n\tof a user profile. The user profile contains the user's operational
    \n\tlimits for system resources. This includes the names of the \n\tobjects, commands,
    and devices that the user owns, and that the \n\tuser is the primary group for.
    \n\n\tContinued help text goes here\n", 2 );
    Comments = ("_!", "admin", 1);
    Roles = ("_!", "admin", 1);
    Argument (useradmin, "User Profile . . . . .", 3 ) {
    Layout = ("", "User_Profile", 1);
    };
};

```

```

Argument (useradmin, "Type of information . . .", 4 ) {
Layout = ("","Info_Type",1);
DefaultValue = "*BASIC";
};
Argument (useradmin, "Output . . . . .", 13 ) {
Layout = ("","Output",1);
DefaultValue = "*";
};
Argument (useradmin, "File to receive output . .", 17 ) {
Layout = ("","Simple_Text",1);
};
Argument (useradmin, "Library . . . . .", 56 ) {
Layout = ("","Library",1);
DefaultValue = "*LIBL";
};
Argument (useradmin, "Output member options:\n      Member to receive output . . .", 19 ) {
Layout = ("","File_Member",1);
DefaultValue = "*FIRST";
};
Argument (useradmin, "      Replace or add records . . .", 20 ) {
Layout = ("","Replace_or_Add_Member",1);
DefaultValue = "*REPLACE";
};
Implementation ( "os400-v3r2", "os400-v3r7" )
.
.Trace On
.Parse Arg usrprf type output outfile library member repladd
.
.CALL ON ERROR NAME error_handler
.CALL ON FAILURE NAME error_handler
.
. if ( output = '*' | output = '*PRINT' ) then
.   Do
.     "DSPUSRPRF USRPRF("usrprf") TYPE("type,"
.       ") OUTPUT("output")"
.   End
. if ( output = '*OUTFILE' ) then
.   Do
.     "DSPUSRPRF USRPRF("usrprf") TYPE("type,"
.       ") OUTPUT("output") OUTFILE("library"/"outfile,"
.       ") OUTMBR("member" "repladd")"
.   End
.
.Exit 0
.
.
.error_handler:
.   Exit 1
;
};
Task Change_Object_Ownership {
Description = (useradmin, "Change Object Owner (CHGOBJOWN)", 21 );
HelpMessage = (useradmin, "Help text for CHGOBJOWN can go here..", 22 );
Comments = ("!_", "admin",1);
Roles = ("!_", "admin",1);
Argument (useradmin, "Object . . . . .", 23 ) {
Layout = ("","Simple_Text",1);
};
Argument (useradmin, "Library . . . . .", 56 ) {
Layout = ("","Library",1);
DefaultValue = "*LIBL";
};
Argument (useradmin, "Object type . .", 24 ) {

```

```

Layout = ("","Object_Type",1);
};
Argument (useradmin, "New owner . .", 25 ) {
Layout = ("","Simple_Text",1);
};
Argument (useradmin, "Current owner authority . . ", 26 ) {
Layout = ("","CurOwn_Authority",1);
DefaultValue = "*REVOKE";
};

Implementation ( "os400-v3r2", "os400-v3r7" )
.
.Trace On
.Parse Arg object library objtype newown curownaut
.
.CALL ON ERROR NAME error_handler

.CALL ON FAILURE NAME error_handler

.
.      "CHGOBJOWN OBJ("library"/"object") OBJTYPE("objtype,
.      ") NEWOWN("newown") CUROWNNAUT("curownaut")"
.
.Exit 0
.
.
.error_handler:
.  Exit 1
;
};
Task Add_Directory_Entry {
Description = (useradmin, "Add Directory Entry (ADDDIRE)", 33 );
HelpMessage = (useradmin, "Help text for ADDDIRE can be placed here ...",
43 );
Comments = ("!_", "admin",1);
Roles = ("!_", "admin",1);
Argument (useradmin, "User identifier:\n      User ID . . . . .",
34 ) {
Layout = ("","Simple_Text",1);
};
Argument (useradmin, "      Address . . . . .", 35 ) {
Layout = ("","Simple_Text",1);
};
Argument (useradmin, "User Description . . . .", 36 ) {
Layout = ("","Simple_Text",1);
};
Argument (useradmin, "User Profile . .", 37 ) {
Layout = ("","User",1);
};
Argument (useradmin, "System name:\n      System name . . . .", 38 ) {
Layout = ("","System_Name",1);
DefaultValue = "*LCL";
};

Implementation ( "os400-v3r2", "os400-v3r7" )
.
.Trace On
.Parse Arg usrid address usrd user sysname
.
.CALL ON ERROR NAME error_handler

.CALL ON FAILURE NAME error_handler

.
      "ADDDIRE USRID("usrid" "address") USRD("usrd,
.      ") USER("user") SYSNAME("sysname")"

```

```

        .
        .Exit 0
        .
        .
        .error_handler:
        . Exit 1
    ;
};
Task Remove_Directory_Entry {
    Description = (useradmin, "Remove Directory Entry (RMVDIRE)", 33 );
    HelpMessage = (useradmin, "Help text for ADDDIRE can be placed h
ere ...", 43 );
    Comments = ("!_", "admin",1);
    Roles = ("!_", "admin",1);
    Argument (useradmin, "User identifier:\n      User ID . . . . .
. . . .", 34 ) {
        Layout = ("","Simple_Text",1);
    };
    Argument (useradmin, "      Address . . . . . . .", 35 ) {
        Layout = ("","Simple_Text",1);
    };
    Argument (useradmin, "User Description . . . . .", 36 ) {
        Layout = ("","Simple_Text",1);
    };
Implementation ( "os400-v3r2", "os400-v3r7" )
    .
    .Trace On
    .Parse Arg usrid address usrd
    .
    .CALL ON ERROR NAME error_handler
    .
    .CALL ON FAILURE NAME error_handler
    .
    .      "RMVDIRE USRID("usrid" "address") USRD("usrd)"
    .
    .
    .Exit 0
    .
    .
    .error_handler:
    . Exit 1
    ;
};
}

```

Appendix C. w4rmusrprop

The `w4rmusrprop` script allows you to selectively remove an OS/400 AEF added property from a Tivoli UserProfile. This, in effect, will cause the property to not be managed by Tivoli on an AS/400 Endpoint.

Note

This script is an unreleased, unsupported, and unofficial product. Use it at your own risk.

```
#!/bin/sh
#
# Primary Author: Nick Lloyd
#
# (C) COPYRIGHT Tivoli Systems/IBM. 1999
# Unpublished Work
# All Rights Reserved
# Licensed Material - Property of Tivoli Systems/IBM.
#
# This script is an unreleased, unsupported, and unofficial product.
# Use at your own risk. Tivoli Systems will not be responsible for
# the loss of data or other damage resulting from the use of this
# script.
#

APP_NAME=`basename $0`
IRONAME=`wtmrname`
AFLAG=""
UFLAG=""
UP_NAME=""

usage()
{
    echo "Usage,"
    echo "$APP_NAME [ -h ] [ -a | -u @UserProfile:<profile_name> ] -p <OS400_Property>"
    echo "where:"
    echo "  -h:          Display this help message."
    echo "  -a:          Remove <OS400_Property> from ALL Tivoli UserProfiles."
    echo "              This is the default action."
    echo "  -u:          Remove <OS400_Property> from specific Tivoli UserProfile."
    echo "              The -a and -u flags are mutually exclusive."
    echo "  -p:          AEF property to remove."
    echo ""
    echo "The w4rmusrprop script allows customers to selectively remove an OS400 AEF added"
    echo "property from a Tivoli UserProfile. This in effect will cause the property to"
    echo "NOT be managed by Tivoli on an AS/400 Endpoint."
    echo ""
    echo "-----"
    echo "Special Considerations:"
    echo ""
    echo "Do not remove the OS400_Login property. The OS/400 Endpoint code requires"
    echo "this property to exist."
    echo ""
    echo "Records created through population will need to have the desired properties"
    echo "removed again. This is because the Endpoint does not know the property has"
```

```

echo "been removed. It returns all supported properties during a population."
echo ""
echo "After running this script, it maybe required to kill the following processes:"
echo "1. uto_skel1"
echo "2. UtoUiCbs"
echo ""
echo "These processes may contain cached copies of the records from which"
echo "<OS400_Property> has been removed. These copies will still have"
<OS400_Property>."
echo "Thus, the effect of the removal will not be noticed until the process is
restarted."
echo ""
echo "Examples:"
echo ""
echo "w4rmusrprop -p OS400_Initial_Menu"
echo "This will remove OS400_Initial_Menu from every Tivoli UserProfile, including the"
echo "TivoliDefaultUserProfile."
echo ""
echo "w4rmusrprop -u @UserProfile:Development -p OS400_UID"
echo "This will remove OS400_UID only from Tivoli UserProfile Development."
echo ""
echo "w4rmusrprop -u \"@UserProfile:HAS SPACE\" -p OS400_Login_Text"
echo "A UserProfile with spaces in its name must be quoted."

exit 1
}

while getopts ahu:p: flag
do
    case $flag in
        a) AFLAG=1
            ;;

        h) usage
            ;;

        u) UFLAG=1
            UP_NAME="$OPTARG"
            ;;

        p) PROP_TO_REMOVE="$OPTARG"
            ;;

        *) usage
            ;;
    esac
done

if [ "$UFLAG" != "1" ]
then
    AFLAG=1
fi

if [ "$AFLAG" = "$UFLAG" ]
then
    echo "${APP_NAME}: The -a and the -u flags are mutually exclusive. They cannot be
specified at the same time."
    usage
fi

if [ "$PROP_TO_REMOVE" = "" ]
then
    echo "${APP_NAME}: You must specify a property to remove with the -p flag."

```

```

        usage
    fi

    if [ "$AFLAG" = "1" ]
    then
        #
        # Find the default UserProfile and register it.
        #
        REGISTER=0
        wlookup -r UserProfile "TivoliDefaultUserProfile#$IRONAME" >/dev/null 2>&1
        if [ $? -ne 0 ]
        then
            DEF_UP_OID=`wls -l /Library/UserProfile | awk '$2 == "TivoliDefaultUserProfile"'`
            {print $1;}'\`
            if [ -n "$DEF_UP_OID" ]
            then
                wregister -r UserProfile TivoliDefaultUserProfile $DEF_UP_OID
                REGISTER=1
            fi
        fi

        DEFAULT_IFS="$IFS"
        IFS="
"
        for USER_PROFILE_ENTRY in `wls -l /Library/UserProfile`
        do
            IFS="$DEFAULT_IFS"
            set $USER_PROFILE_ENTRY
            PROFILE_OID="$1"
            shift
            USER_PROFILE="$*"

            UP="@UserProfile:$USER_PROFILE#$IRONAME"

            echo "Removing $PROP_TO_REMOVE from UserProfile $UP"
            wrmpop "$UP" $PROP_TO_REMOVE
        done

        IFS="$DEFAULT_IFS"

        #
        # Unregister the DefaultUserProfile if we registered it.
        #
        if [ $REGISTER -eq 1 ]
        then
            wregister -u -r UserProfile "TivoliDefaultUserProfile#$IRONAME"
        fi
    else
        wrmpop "$UP_NAME#$IRONAME" $PROP_TO_REMOVE
    fi

    exit

```

Appendix D. Special notices

This publication is intended to help system administrators, technical users and customers of Tivoli Enterprise products to understand more about the AS/400 Endpoint. The information in this publication is not intended as the specification of any programming interfaces that are provided by Tivoli Enterprise application suite. See the PUBLICATIONS section of the IBM Programming Announcement for Tivoli for AS/400 Endpoints and Tivoli core applications for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

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, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (vendor) products in this manual has been supplied by the vendor, and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate

them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

This document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain 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.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	AS/400
DB2	IBM
OS/2	OS/390
RISC System/6000	RMF
RS/6000	S/390
SP	System/36
System/390	SystemView

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli

A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Lotus Notes is a registered trademark of Lotus Development Corporation.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 529.

- *All About Tivoli Management Agents*, SG24-5134
- *An Introduction to Tivoli Enterprise*, SG24-5494
- *Tivoli Enterprise Internals and Problem Determination*, SG24-2034
- *Tivoli Enterprise Performance Tuning Guide*, SG24-5392
- *Tivoli Security Management Design Guide*, SG24-5101
- *Tivoli User Administration Design Guide*, SG24-5108

E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at <http://www.redbooks.ibm.com/> for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

E.3 Other resources

These publications are also relevant as further information sources:

- *AS/400 Alerts Support V4R1*, SC41-5413
- *AS/400 Backup and Recovery Guide*, SC41-8079
- *AS/400 Programming: Control Language Ref V2R3MO*, SC41-0030
- *AS/400 Programming Work Management Guide*, SC21-8078
- *AS/400 Security - Basic V4R1*, SC41-5301
- *AS/400 System API Reference V3R2*, SC41-3801
- *AS/400 System Operation V3.6.0*, SC41-4203
- *Basic System Operation, Administration, and Problem Management*, SC41-5206
- *Communications Configuration V4R1*, SC41-5401
- *Distributed Monitoring User's Guide 3.6*, GC31-8382
- *IBM AS/400 Screen Design Aid User's Guide and Reference*, SC09-1171
- *OS/400 Alerts Support V3.2.0*, SC41-3413
- *OS/400 CL Reference V3R2*, SC41-3722
- *OS/400 Security - Reference V4R4*, SC41-5302
- *Security Management User's Guide 3.6*, GC31-8475
- *Software Installation Services 3.6 User's Guide*, GC31-5121
- *TCP/IP Configuration and Reference V4R4*, SC41-5420
- *Tivoli for AS/400 Endpoints CD-ROM*, LK3T-3639
- *Tivoli Framework AS/400 3.6.2 User's Guide*, GC32-0279
- *TME 10 Task Library Developer's Guide*, SC31-8436
- *TME 10 AEF 3.6 User's Guide*, GC31-8345
- *TME 10 AS/400 3.6 Release Notes*, GI10-4775
- *TME 10 Enterprise Console Adapter's Guide Version 3.6*, SC31-8507
- *TME 10 Enterprise Console Rule Builder's Guide*, SC31-8508
- *TME 10 Enterprise Console User's Guide Version 3.6*, GC31-8506
- *TME 10 Framework 3.6 Planning & Installation Guide*, SC31-8432
- *TME 10 Framework 3.6 Reference Manual*, SC31-8434
- *TME 10 Inventory 3.6 User's Guide*, GC31-8381
- *TME 10 Software Distribution 3.6 User's Guide*, GC31-8330
- *TME 10 Software Distribution 3.6 Reference Manual*, SC31-8331

- *TME 10 Task Library 3.6 Developer's Guide*, SC31-8436
- *TME 10 TEC Reference Guide 3.6*, SC31-8505
- *User Admin 3.6.2 Users Guide*, GC32-0291

The following materials are product documentation and can only be purchased with the software product:

- *Tivoli AS/400 3.6.1 Release Notes*, GI10-8016
- *Tivoli Enterprise Console 3.6.1 Release Notes*, GI10-8020

E.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- <http://www.tivoli.com/products/documents/whitepapers/>
- <http://www.redbooks.ibm.com/solutions/tivoli>
- <http://publib.boulder.ibm.com:80/>
- <http://www.tivoli.com/support/>
- <http://www.redbooks.ibm.com>
- <http://www.as400.ibm.com/>

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** <http://www.redbooks.ibm.com/>

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States or Canada	pubscan@us.ibm.com
Outside North America	Contact information is in the "How to Order" section at this site: http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibm.link.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: http://www.elink.ibm.link.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name
------------	-----------

Company

Address

City	Postal code	Country
------	-------------	---------

Telephone number	Telefax number	VAT number
------------------	----------------	------------

<input type="checkbox"/> Invoice to customer number	
---	--

<input type="checkbox"/> Credit card number	
---	--

Credit card expiration date	Card issued to	Signature
-----------------------------	----------------	-----------

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

AIX. Advanced Interactive Executive.

BARC. Before After Remove Commit Script.

CLI. Command Line Interface.

DB2. Database 2.

DLL. Dynamic Link Library.

DNS. Domain Name Service.

FTP. File Transfer Protocol.

GUI. Graphical User Interface.

IBM. International Business Machines.

IFS. Integrated File System.

IPCS. Integrated Personal Computer Server.

ITSO. International Technical Support Organization.

LCF. Lightweight Client Framework.

MDist. Multiplexed Distribution.

MIF. Management Information File.

RDBMS. Relational Database Management System.

TAP. Tivoli Authentication Package.

TCP/IP. Transmission Control Protocol/Internet Protocol.

TEC. Tivoli Enterprise Console.

TMA. Tivoli Management Agent.

TME. Tivoli Management Environment.

TMF. Tivoli Management Framework.

TMR. Tivoli Management Region.

TNR. Tivoli Name Registry.

Index

Symbols

.netrc file 185

A

account information 140
AIX 8
alarm 218
alert 440
alert filter 298
alert table 294
alternate Gateway 74
AnyNet/400 464
API 174, 298
application life-cycle 401
Application Response Management (ARM) 41, 225
AS/400 2, 38, 71, 416
AS/400 Alert Adapter 7, 271, 300, 452, 464
AS/400 Application Development Tools 250
AS/400 Client Access 414
AS/400 database file 327
AS/400 Event Adapter 217
AS/400 LPO 314
AS/400 Message Adapter 7, 270, 332, 452
AS/400 platform 1
ASCII text file 403
assigned Gateway 64, 68, 74, 87
automated task 313
autopack 404, 408
AutoStart job 125
availability management 217

B

backup 4
backup job 447
BARC script 409
baroc file 222
batch job 432, 440
Before Distribution 499
binary 39
binary file 74
boot_engine 234
boot_method 70
Brazilian Portuguese 12

C

C program 403
CCSID parameter 368
CD-ROM 26, 39, 49, 273, 414, 421
CDS file 283, 300, 307, 312, 325, 361
central/US time 222
CGI script 130
Chinese 12
CISC 6, 25, 38
CL program 243, 250, 308, 403, 454, 467
CL Source 125
class definition 289
CLI 10, 14, 72, 82
Cobol program 402
Common Object Request Broker Architecture (CORBA) 1
communication 2
communication line 467
configuration file parameter 454
configuration program 412
configuration repository 421
configuration repository schema 421
Control Language command 76, 173
CPU 5, 217
CPU usage 467
custom MIF file 432
custom monitor 219, 226

D

data file 401
database profile manager 121, 408, 426
dataless client 120
dataless mode 407
dataless profile manager 120, 182, 219, 408, 426
DB2/400 402
Dedicated Server for Domino (DSD) 4
default Endpoint policy script 72
default policy 174
delete_endpoint 88
dependency set 124
deployment management 401
Destination Directory Path 498
DFU 273
disk space 39
disk storage 467
diskette drive 421
Distributed Monitoring 6, 23, 40, 217, 440

- Distributed Monitoring ARM 41
- Distributed Monitoring configuration 233
- Distributed Monitoring engine 41, 57, 217, 219
- Distributed Monitoring monitor 235
- Distributed Monitoring profile 217
- dm_ep_engine 223
- DM36.log 227
- DMEPEENGINE 219, 227, 258
- dogEndpoint 223
- DOGENPT 220
- downcall 15, 89, 93

E

- EBCDIC 410
- EIF 273, 314
- e-mail 218, 222, 445, 467
- Endpoint 6, 12, 14, 21, 37, 39, 47, 61, 93, 120, 426, 438
- Endpoint configuration file 74
- Endpoint configuration parameter 77
- Endpoint daemon 124, 126, 133
- Endpoint Gateway 13, 15, 20, 31, 35, 38, 39, 46, 61, 70, 91, 123, 271, 402, 423, 438
- Endpoint Gateway database 123
- Endpoint Gateway object reference 68
- Endpoint job 57
- Endpoint list 16
- Endpoint login 61, 89
- Endpoint machine 71
- Endpoint Manager 13, 16, 37, 43, 63, 123
- Endpoint Manager database 50
- Endpoint method 122, 128, 133
- Endpoint method cache 122, 128
- Endpoint Policies
 - after_install_policy 19, 67, 70, 84
 - allow_install_policy 67, 83
 - login_policy 70, 86, 134, 136
 - select_gateway_policy 67, 84, 95
- Endpoint policy 17, 19, 134
- Endpoint policy argument 70
- Endpoint policy script 68
- Endpoint software 86
- Endpoint web interface 127
- English 12
- environment variable 264
- error 412
- error message 316

- Ethernet 38
- event 218
- Event Adapter 6
- event class 218
- event correlation 307
- executable program 401

F

- fan-out 16
- FETCH statement 285
- field data 274
- file member 413
- file package 401, 413
- file package block 404, 413
- Filter statement 284
- FilterCache statement 284
- FilterMode statement 284
- financial organization 437
- flat file 58
- French 12
- FTP 48, 52, 54, 106, 116, 273, 307, 331, 407, 420, 495

G

- Gateway assignment 65, 68, 87
- German 12
- global users 181
- group ID 243
- group profile 5, 141
- GUI 18, 140, 414, 420
- gwdb.bdb file 123

H

- hardware 38, 402
- hardware inventory 421
- hardware MIF file 427
- hardware requirement 142, 210, 224, 405, 422
- hardware scanning 427
- help dialog 75
- high availability 4, 20
- hostname 83, 126
- HTTP daemon 126
- HTTP interface 56
- HTTP server 130

I

- IBM 4

- icon 221
- IFS directory format 116
- IFS folder 419
- IFS format 283, 412, 498
- imdb.bdb file 123
- indicator collection 221
- initial login 15, 39, 65, 70, 81, 87
- installation 42, 58
- installation repository (IR) 58
- Integrated File System (IFS) 106, 174, 404, 419
- Integrated PC File Server 402
- intercepting Gateway 65
- Internationalization 10
- interpreter 17, 25, 39
- Inventory 6, 23, 41, 402, 421, 483
- Inventory database 421, 424
- Inventory profile 424, 426, 483
- IP 329
- IP address 68, 71, 126
- IPL 15, 124, 309, 440, 501
- isolated login 65
- Italian 12

J

- Japanese 11
- Java applet 130
- job 467
- job name 454
- joblog 171, 227, 233, 258

K

- Korean 12

L

- language 51
- language code set 11
- last.cfg file 74, 107, 128
- LCF architecture 9
- lcf.dat file 61, 74, 128
- lcf.d 14, 19, 51, 220
- lcf.d.log file 112, 128
- license program 273
- licensed program 56
- Lightweight Client Framework (LCF) 2, 12, 38
- local installation 45
- log threshold 81
- login information 74

- login interfaces 95
- login interfaces list 68
- login policy 478
- login request 64, 68
- login_interval 80, 130

M

- Managed Node 12, 15, 21, 38, 45, 46, 91, 120, 218
- managed resource 122
- management topology 20
- MAP statement 285
- MAP_DEFAULT statement 285, 304
- mass installation 59
- MDist 402
- MDist repeater 16, 20
- memory 421
- message 440
- message ID 274, 279, 461
- message queue 170, 202, 283, 353, 452
- message text 274
- method 91, 122
- method invocation 16
- microcode 4
- Microsoft Internet Explorer 126
- mid-level manager 20
- MIF file 428, 489
- migration 88
- migratory downcall 93
- migratory login 89
- migratory upcall 91
- monitor 219, 449
- monitoring action 219
- monitoring collection 218, 237
- MS Network Neighborhood 403
- msg_key attribute 357
- multiple TMRs 95

N

- Netscape Navigator 126
- NetView DM 404
- NetWare 10, 140
- NetWare Managed Site 12, 120
- network 5, 209, 414
- network alias 74
- network collision 217
- network computing 402
- network interface card 421

- networking 38
- new_endpoint 88
- NIS Domain 120
- NLS 173
- nobody 225
- non-TME adapter 218, 271, 282
- non-TME event 218, 271
- normal login 61, 70
- numeric script 267

O

- object 202, 467
- object database 121
- Object Management Group (OMG) 1
- object ownership 172
- Object Packager utility 413
- Object Packaging Utility for AS/400 404
- object reference 71
- object request broker 218
- object type 13
- OBJECT_NIL 71
- odnum 17
- operating system 2, 17, 39, 127
- OS/2 10
- OS/400 6, 33, 39
- OS/400 Alert 217
- OS/400 attribute 145
- OS/400 auditing information 146
- OS/400 Commands

- ADDALRACNE 299
- ADDALRSLTE 299
- ADDIRE 106, 172
- ADDRTGE 377
- ANZPRB 332
- CALL 55, 225, 433, 492
- CHGCMDDFT 174
- CHGJOB 315
- CHGNETA 292
- CHGOBJOWN 106, 172
- CHGTCPDMN 312
- CHGUSRAUD 162, 169
- CHGUSRPRF 162, 164
- CHKPRDOPT 315
- CPYF 340
- CPYFRMSTRMF 110
- CPYSPLF 115
- CRTDTA 298

- CRTLIB 113
- CRTMSGQ 376
- CRTMSGQA 280
- CRTPF 113
- CRTSRCPF 244, 384
- CRTUSRPRF 162, 163
- DLTLICPGM 56
- DLTUSRPRF 57, 162, 170
- DSPLIB 249, 406
- DSPLOG 311
- DSPMSG 253, 281
- DSPNETA 292
- DSPOBJAUT 100
- DSPPFM 116, 328
- DSPPTF 315
- DSPSFWRSC 314
- DSPUSRPRF 106, 185
- EDTOBJAUT 105
- ENDJOB 57
- ENDTECADP 251, 308, 465
- ENDTMEEPT 57, 76, 110
- GO 391
- QLCFSTAR 125
- QSH 225
- RMVDIRE 106, 172
- RSTLICPGM 55, 273
- SBMJOB 225
- SNDMSG 226, 253, 278
- SNDRPY 353
- SNDUSRMSG 279, 332
- STRPRTWTR 226
- STRSDA 384
- STRSEU 245, 342, 465
- STRTECADP 284, 307, 454
- STRTMEEPT 74, 107, 125
- WRKACTJOB 57, 179, 250, 323
- WRKALR 330, 333
- WRKALRD 294
- WRKFTR 330
- WRKJOB 179, 263
- WRKJOBQ 177
- WRKLNK 330
- WRKMBRPDM 244
- WRKMSGD 275
- WRKOBJOWN 57
- WRKPRB 294
- WRKSPLF 176, 233, 320
- WRKSYSVAL 222, 482
- WRKUSRPRF 208

OS/400 directory information 148
OS/400 display session information 150
OS/400 group membership information 152
OS/400 international information 153
OS/400 login information 154
OS/400 output information 156
OS/400 password information 157
OS/400 privileges information 158
OS/400 session startup information 160
oserv 218

P

password 48, 129, 157, 185, 476
patch 30, 39, 58, 134, 143, 210, 224, 405, 422
PC Managed Node 12, 120
PDM 250, 342
performance 5, 427
Perl 47
physical file 106
platform 10, 17, 124, 209, 217
policy region 18, 45, 70, 84
polling loop 220
pop-up 218, 444
port 51, 74, 126, 220, 283
prerequisite check 47
printer 421
privilege class 159
processor 38, 421
product ID 492
profile distribution 162, 427
profile manager 84, 182, 219, 235, 407
profile manager hierarchy 120
profile population 144
protocol 39
PTF 4, 33, 39, 272, 382, 421, 483
PTF information 428

Q

QBASE subsystem 441
QINTER subsystem 441
QLCFD 234
Qshell 225, 228
Qshell Interpreter 225, 228
QSYS 412
QSYS library 281, 443
QTIVOLI 223
QTIVROOT 223
QTIVUSER 223

query 402, 423, 493
query library 428, 494

R

RDBMS 402, 421, 428
recovery 4
region redirect 65
relational database 2, 402
response 218, 221
REXX 369, 403, 432, 495
RIM object 421
RISC 6, 25, 38
role-based security administration 209
RPG 402
rule-based event management 218
rules engine 313
Run Query 489

S

S/390 8
save file 54
scalability 4
scanning 423
scanning information 428
scanning target 424
Screen Design Aid (SDA) 384
security 2, 4
Security group 213
Security Management 6, 23, 40, 139, 209
security policy 5, 476
Security profile 211
Security system policy 214
SELECT statement 285
SEU 250, 272, 342, 361, 453
severity 274, 464
sign-on information 160
SIS 40, 42, 46, 58
SNA 172
SNA alert 464
SNA event 307
SNA format 307
SNA link 440, 464
SNA_Equipment_Malfunction class 312
SNMP trap 300
socket connection 54
software 39, 57, 402
Software Distribution 6, 23, 41, 401, 483
software inventory 421

- software MIF file 427
- software requirement 143, 210, 224, 405, 422
- software scanning 427
- source host 412, 497
- Spanish 12
- special character 182
- spooled file 320
- SQL data 402
- string script 267
- sub_source attribute 357
- subcommand 497
- subnet 83
- subscriber 120, 145, 182, 411, 426
- subsystem 309
- subsystem management 440
- system administrator 172, 421
- system configuration 421
- system operator 445
- system requirement 38
- system resource 440
- system security policy 141
- systems management 2

T

- tape drive 421
- target 408
- task 172, 222, 313, 462
- task library 99, 172, 456
- Tasks
 - Command 99
 - Power_Down_System 99
 - Run_Backup 99
 - Send_Reply 99, 356
 - Start_Cleanup 99
 - Vary_Configuration 99
- TCP 39
- TCP/IP 39, 310, 312, 329, 464
- TCP/IP domain 312
- TCP/IP host table 329
- TEC 6, 42, 106, 171, 218, 222, 244, 441, 452
- TEC Adapters for AS/400 23, 27, 440
- TEC console 311, 331
- TEC event 312
- TEC event attribute 286
- TEC event server 218, 271, 282, 329, 445
- TEC reception log 331
- TEC rule base 222, 313

- tec_reception 218
- tec_recv_agent_port 283
- three-tiered structure 20, 61
- threshold 217
- timezone 222
- Tivoli 4
- Tivoli administrator 17, 47, 59, 244
- Tivoli Application Extension Facility (AEF) 140, 175
- Tivoli Commands
 - lcf.sh 19
 - postmsg 332
 - w4getusr.pl 184
 - w4inslcf.pl 19, 42, 52, 58
 - w4instlcf.pl 47
 - wadminep 20, 33, 82, 134
 - wclient 45
 - wcrtgate 19, 45, 46
 - wdelgate 20
 - wdistfblock 420
 - wdistrib 397
 - wep 20, 87, 129
 - wgateway 20, 124
 - wgeteppol 20, 72
 - wgetusr 146
 - winstlcf 19
 - wln 19
 - wls 44
 - wlseng 232
 - wpasswd 164
 - wpopusr 144, 184
 - wpostmsg 332
 - wputeppol 20, 72
 - wrmprop 162
 - wrpt 16
 - wsetdefpol 145
 - wsetpm 19
 - wsetusr 146, 164
 - wstopeng 234
 - wtdumprl 331
 - wtemp 55
- Tivoli Desktop 10, 14, 17, 28, 45, 46, 144, 164, 235, 256, 424, 428, 455
- Tivoli Enterprise 1, 9, 402
- Tivoli Enterprise application 122
- Tivoli for AS/400 Endpoints 23, 39, 49
- Tivoli Management Agent (TMA) 2, 9, 12, 62, 121
- Tivoli Management Application 2
- Tivoli Management Framework 1, 9, 23, 42, 43,

- 72, 122, 134, 401
- Tivoli Manager for Domino 35
- Tivoli Manager for MQSeries 35
- Tivoli Name Registry (TNR) 17
- Tivoli notice 221
- Tivoli Remote Execution Service (TRIP) 45
- Tivoli security authority 40
- Tivoli Security Management 472
- Tivoli Service Desk 402
- Tivoli user account 57
- Tivoli website 32
- TME adapter 218
- TME event 218, 271
- TMR 2, 7, 13, 20, 38, 39, 45, 61, 83, 213, 499
- TMR redirection 95
- TMR server 15, 20, 30, 37, 40, 43, 61, 123, 219, 420, 432, 438
- TMR structure 21
- Token-Ring 38
- trace log 128
- two-tiered structure 20

U

- UDP 39, 329
- Universal collection 219, 226, 267
- UNIX 8, 118, 140, 229, 421
- upcall 16, 89, 91
- URL 32, 126
- User Administration 6, 23, 40, 139, 472
- user class 159
- user ID 48, 129, 243
- user profile 4, 99, 141, 144, 167, 182, 472
- user record 191
- user's initial program 160
- user's special authority 159

V

- validation policy 145, 174, 175
- vi editor 128

W

- web browser 126
- web server function 126
- Windows Explorer 416
- Windows interface 414
- Windows NT 10, 45, 140, 407
- work flow 440

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular, we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook addressing value, subject matter, structure, depth, and quality as appropriate.**

- Use the online **Contact us** review redbook form found at <http://www.redbooks.ibm.com/>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number	SG24-5517-00						
Redbook Title	Managing AS/400 with Tivoli Enterprise						
Review	<table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>						
What other subjects would you like to see IBM Redbooks address?	<table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>						
Please rate your overall satisfaction:	<input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor						
Please identify yourself as belonging to one of the following groups:	<input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer <input type="radio"/> IBM, Lotus or Tivoli Employee <input type="radio"/> None of the above						
Your e-mail address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	<input type="radio"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.						
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. http://www.ibm.com/privacy/yourprivacy/						

SG24-5517-00
Printed in the U.S.A.

Managing AS/400 with Tivoli Enterprise

SG24-5517-00

